

**НАЦІОНАЛЬНИЙ УНІВЕРСИТЕТ КОРАБЛЕБУДУВАННЯ
ІМЕНІ АДМІРАЛА МАКАРОВА**

Навчально-науковий інститут комп'ютерних наук та управління проектами
(повне найменування інституту, назва факультету)

Інформаційних управляючих систем та технологій
(повна назва кафедри)

Пояснювальна записка

до дипломного проекту (роботи)

магістр

(освітньо-кваліфікаційний рівень)

на тему **Дослідження web-технологій та розробка інформаційної
системи хостингової компанії**

Виконав: студент 6 курсу, групи 6144м
напряму підготовки (спеціальності)

126 – "Інформаційні системи та технології".
(шифр і назва напряму підготовки, спеціальності)

Крамаренко О.С.

(прізвище та ініціали)

Керівник Ушкац М.В.

(прізвище та ініціали)

Рецензент Книрік Н.Р.

(прізвище та ініціали)

м. Миколаїв – 2021 р.

НАЦІОНАЛЬНИЙ УНІВЕРСИТЕТ КОРАБЛЕБУДУВАННЯ ІМЕНІ АДМІРАЛА МАКАРОВА

Інститут, факультет Навчально-науковий інститут комп'ютерних наук та управління проектами
Кафедра Інформаційних управляючих систем та технологій
Рівень вищої освіти другий (магістерський)
Спеціальність 126 – "Інформаційні системи та технології"

ЗАТВЕРДЖУЮ

Завідувач кафедри к.т.н., доцент
Михелєв І.Л.

" " 2021 року.

ЗАВДАННЯ НА ДИПЛОМНИЙ ПРОЕКТ (РОБОТУ) СТУДЕНТУ

Крамаренко Олексій Сергійович

1. Тема проекту (роботи) Дослідження web-технологій та розробка інформаційної системи хостингової компанії.

керівник проекту (роботи) Ушкац Михайло Вікторович,
затверджені наказом вищого навчального закладу від " 12 " 10 2021 року № 1232-уч.

2. Строк подання студентом проекту (роботи) 20.12.2021 року.

3. Вихідні дані до проекту (роботи) ДСТУ щодо обробки інформації, літературні джерела, технічна документація на існуючі аналоги систем, матеріали практики.

4. Зміст розрахунково-пояснювальної записки (перелік питань, які потрібно розробити):

- Титульний аркуш, завдання на дипломну роботу, анотація (українською, російською, англійською), зміст, перелік умовних означень, символів, одиниць та термінів (при необхідності).
- Вступ.
- Формування вимог до проекту інформаційної системи хостинг-компанії "MyHost", розробка її концепції та постановка задачі.
- Розробка проекту інформаційної системи.
- Розробка робочої документації.
- Розділ з економіки.
- Розділ з охорони праці.
- Розділ з охорони навколишнього середовища.

- Висновки.
- Список використаної літератури.
- Додатки.

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

- 5.1 Концепція системи з обмеженим вибором здійснення платежів.
- 5.2 Концепція системи без взаємодії з системою управління сервером.
- 5.3 Вдосконалена концепція інформаційної системи.
- 5.4 Технічне забезпечення, схема автоматизації системи.
- 5.5 Контекстна діаграма потоків даних.
- 5.6 Діаграма "Сутність-зв'язок".
- 5.7. Логічна структура бази даних системи
- 5.8. Алгоритм виконання замовлення.
- 5.9. Екранні форми (графічний інтерфейс).

6. Консультанти розділів проекту (роботи)

Розділ	Прізвище, ініціали та посада консультанта	Підпис, дата	
		завдання видав	завдання прийняв
1.	Ушкац М.В.	07.09.2021	08.10.2021
2.	Ушкац М.В.	11.10.2021	22.10.2021
3.	Ушкац М.В.	25.10.2021	26.11.2021
4.	Ушкац М.В.	07.09.2021	29.11.2021
5.	Гурець Н.В.	07.09.2021	17.12.2021
6.	Гурець Н.В.	07.09.2021	17.12.2021

7. Дата видачі завдання 07.09.2021

КАЛЕНДАРНИЙ ПЛАН

№ з/п	Назва етапів дипломного проекту (роботи)	Строк виконання етапів проекту (роботи)	Примітка
1.	Вивчення і опис предметної області	08.10.2021	
2.	Розробка концепції ІС	22.10.2021	
3.	Розробка проектних рішень по ІС	26.11.2021	
4.	Реалізація проекту, розробка робочої документації	26.11.2021	
5.	Вирішення питань з економіки	29.11.2021	
6.	Вирішення питань з охорони праці та навколишнього середовища	17.12.2021	
7.	Виконання графічної частини роботи	17.12.2021	
8.	Оформлення пояснювальної записки	17.12.2021	
9.	Подання роботи рецензенту	17.12.2021	
10.	Подання роботи на державний захист	20.12.2021	

Студент _____ Крамаренко О.С.

Керівник проекту (роботи) _____ д.ф.-м.н., професор Ушкац М.В.

АНОТАЦІЯ

Дипломна робота присвячена розробці проекту інформаційної системи хостингової компанії.

В роботі сформовані вимоги до системи, розроблені її концепції і проведено вибір найкращої системи. Дипломна робота містить основні проектні рішення за технічним, інформаційним та програмним забезпеченням, сформовано робоча документація розробленого програмного забезпечення.

Дипломна робота викладена на 120 друкованих сторінках й містить: 20 рисунків, 26 таблиць, 5 додатків та список літератури з 10 найменувань.

АННОТАЦИЯ

Данная дипломная работа посвящена разработке проекта информационной системы хостинговой компании.

В работе сформулированы требования к системе, разработаны ее концепции и выбрана наилучшая система. Дипломная работа содержит основные проектные решения по техническому, информационному и программному обеспечению, сформирована рабочая документация по разработанному программному обеспечению.

Дипломная работа изложена на 120 страницах машинописного текста и содержит: 20 рисунков, 26 таблиц, 5 приложений и список литературы из 10 наименований.

ABSTRACT

Degree work is devoted to the drafting of an information system hosting company.

In this paper we formulated the requirements for the system, it developed the concept and selected the best system. Diploma thesis contains the basic design decisions on technical, information and software, a working documentation developed software.

Diploma work is expounded on 120 the pages of the typewritten text and contains: 20 pictures, 26 tables, 5 appendixes and list of literature from the 10 names.

ЗМІСТ

Вступ.....	9
1. Формування вимог до проекту інформаційної системи хостинг-компанії "MyHost", розробка її концепції та постановка задачі.....	12
1.1 Формування вимог до інформаційної системи	12
1.1.1 Характеристика системи і результати її діяльності.....	12
1.1.2 Опис аналогів інформаційної системи та її недоліків	14
1.1.3 Обґрунтування необхідності вдосконалення аналогів інформаційної системи	19
1.1.4 Цілі, критерії та обмеження створення інформаційної системи.....	20
1.1.5 Функції і задачі створюваної інформаційної системи.....	20
1.1.6 Очікувані техніко-економічні результати створення підсистеми	21
1.1.7 Висновки і пропозиції	21
1.2 Розробка концепції інформаційної системи.....	21
1.2.1 Опис результатів вивчення об'єкта автоматизації.....	21
1.2.2 Опис і оцінка варіантів концепції програмних засобів.....	22
1.2.2.1 Вибір технологій для розробки інформаційної системи	22
1.2.2.2 Розробка варіантів концепції програмних засобів	27
1.2.3 Опис і оцінка недоліків розроблених альтернативних варіантів концепції створення системи.....	30
1.2.4 Функціональний аналіз.....	32
1.2.5 Вартісний аналіз	34
1.3 Постановка задачі	37
2. Розробка проектних рішень інформаційної системи хостингової компанії....	42
2.1 Загальносистемні рішення.....	42
2.1.1 Схема організаційної структури.....	42
2.1.2 Схема функціональної структури	42
2.1.3 Опис функцій, що автоматизуються.....	49
2.1.3.1 Вихідні дані	49
2.1.3.2 Цілі автоматизованої системи і автоматизовані функції	50
2.1.3.3 Характеристика функціональної структури	50
2.1.4 Опис постановки задачі	50
2.1.4.1 Характеристика комплексу задач.....	50
2.1.4.2 Вихідна інформація	51
2.1.4.3 Первісна інформація.....	52
2.1.5 Проектна оцінка надійності системи	52
2.1.6 Загальний опис системи	54
2.2 Рішення з організаційного забезпечення.....	55
2.2.1 Опис організаційної структури.....	55
2.2.2 Інструкція користувача	56
2.2.3 Заходи щодо організаційного забезпечення	56
2.2.4 Організаційно-кадрові заходи.....	56
2.3 Рішення з інформаційного забезпечення	56
2.3.1 Опис інформаційного забезпечення системи.....	56
2.3.2 Перелік вхідних сигналів і даних	60

2.3.3 Перелік вихідних сигналів і даних	61
2.3.4 Відомість машинних носіїв інформації.....	61
2.3.5 Опис організації інформаційної бази	61
2.3.6 Каталог бази даних	65
2.3.7 Склад вихідних даних.....	66
2.3.8 Інструкція з формування і ведення бази даних.....	69
2.4 Рішення з технічного забезпечення	71
2.4.1 Схема автоматизації	71
2.4.2 Опис комплексу технічних засобів	71
2.5 Рішення з математичного забезпечення. Опис алгоритму.....	73
2.5.1 Призначення і характеристика	73
2.5.2 Використана інформація.....	73
2.5.3 Результати розв'язання	73
2.5.4 Алгоритм розв'язання	73
3. Розробка робочої документації інформаційної системи хостингової компанії	76
4. Розрахунок економічного ефекту від впровадження ІС	78
4.1 Загальні положення.....	78
4.2 Розрахунок витрат на створення і впровадження системи	79
4.3 Економічна ефективність розробки і впровадження ІС.....	81
5. Охорона праці	84
5.1 Вступ.....	84
5.2 Аналіз небезпечних і шкідливих виробничих факторів, характерних для приміщення відділу.....	85
5.3 Розрахунок системи освітлення для приміщення відділу.....	87
5.4 Розробка заходів щодо забезпечення сприятливих умов праці при роботі з персональним комп'ютером	89
6. Охорона навколишнього середовища	93
6.1 Вступ.....	93
6.2 Моніторинг як система спостереження і контролю навколишнього середовища	93
6.3 Методи контролю забруднюючих речовин в об'єктах навколишнього середовища	95
Висновки	99
Список використаних джерел	100
Додаток А – Технічне завдання	102
Додаток Б – SQL-дамп бази даних.....	107
Додаток В – Текст програми	110
Додаток Г – Програма і методика випробувань	114
Додаток Д – Керівництво користувача.....	117

Вступ

У теперішній час інформаційні технології здобули великого значення у промисловості, комерції та звичайному побуті. Мережі передачі даних відіграють все більшу роль у повсякденному житті.

У сфері інформаційної діяльності людей виник інформаційний сектор економіки, основну частину якого становить виключно інформаційна індустрія. В інформаційній індустрії виокремлюється одна з найперспективніших структур – Internet-економіка.

Глобальна комп'ютерна мережа Internet вважається "четвертим каналом" зв'язку людей між собою (після особистого спілкування, телефону і пошти).

Нині активно відбувається зміщення акцентів з комунікаційної та інформаційно-пошукової функцій Internet на реалізацію з її допомогою сучасного бізнесу. Це відбувається завдяки здатності мережових технологій докорінно змінювати спосіб взаємодії між людьми і компаніями, методи дослідницької діяльності, купівлі-продажу тощо. Internet не тільки забезпечує швидке просування мережевого бізнесу, а й змінює та підсилює конкуренцію в більшості традиційних галузей економіки.

Насамперед Internet охоплює найдешевші та найкращі на сьогодні технічні комунікації, що відкриває бізнесменам і споживачам можливості встановлювати і підтримувати в режимі реального часу постійний зв'язок з будь-яким респондентом у світі. Глобальна мережа стала невід'ємним засобом для проведення маркетингу і здійснення прямих on-line продажів, підвищення рівня обслуговування клієнтів, потужним інструментом управління фірмою. Тому кількість web-сайтів в мережі Internet з кожним днем зростає. Відповідно, зростає кількість хостингових компаній, що займаються наданням послуг розміщення даних і web-сайтів на своїх технічних платформах.

Метою даної дипломної роботи є розширення способів здійснення електронних платежів та забезпечення режиму реального часу роботи клієнта з замовленнями у інформаційній системі хостинг-компанії "MyHost".

Для досягнення поставленої мети необхідно вирішити наступні задачі:

- ✓ провести аналіз сучасних інформаційних технологій, які використовуються для розробки інформаційних систем;

- ✓ сформулювати вимоги до системи;
- ✓ розробити концепцію системи;
- ✓ провести техніко-економічне обґрунтування розробки системи;
- ✓ розробити загальносистемні рішення, рішення з інформаційного, організаційного, технічного і програмного забезпечення;
- ✓ розробити необхідне програмне забезпечення системи;
- ✓ розробити відповідну документацію.

1. Формування вимог до проекту інформаційної системи хостинг-компанії "MyHost", розробка її концепції та постановка задачі.

1.1 Формування вимог до інформаційної системи.

1.1.1 Характеристика системи і результати її діяльності.

Управління замовленнями послуг хостингу в Internet відбувається на Web-сайті хостинг-компанії.

Схема інформаційних потоків системи зображена на рисунку 1.1.

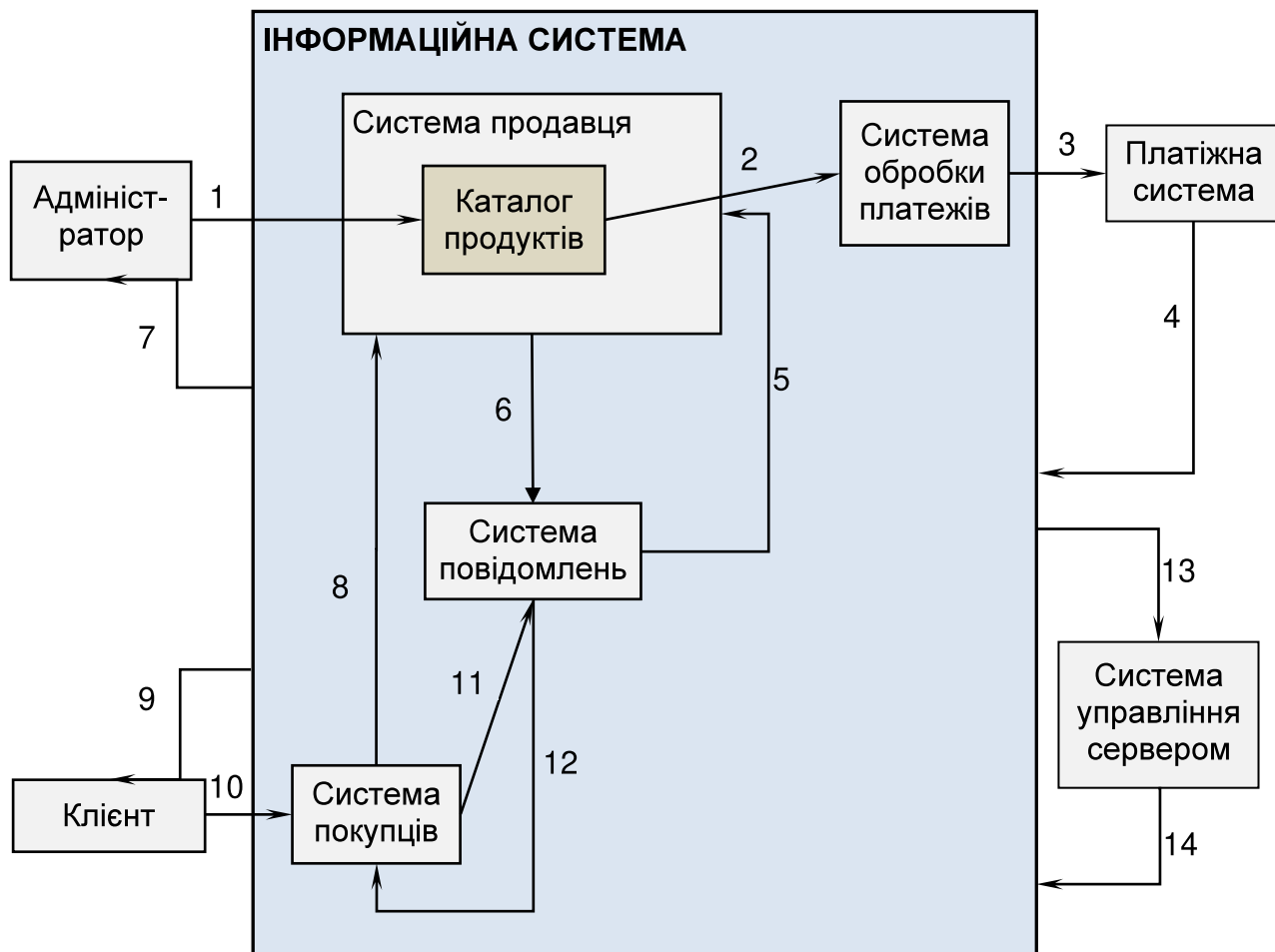


Рисунок 1.1 – Схема інформаційних потоків системи

Умовні позначення, що використані даній схемі розшифровано у вигляді таблиці 1.1.

Таблиця 1.1 – Умовні позначення

Умовне позначення	Пояснення
1	Дані про продукти
2	Дані про вибрані продукти

Умовне позначення	Пояснення
3	Дані для оплати

Продовження таблиці 1.1

4	Дані про результат платежу
5	Дані про повідомлення для адміністратора
6	Дані про повідомлення адміністратора
7	Дані про статистику замовлень
8	Дані про замовлення
9	Дані про статистику замовлень
10	Дані про замовлення
11	Дані про повідомлення покупця
12	Дані про повідомлення для покупця
13	Дані для системи управління сервером
14	Дані від системи управління сервером

Internet-торгівля – форма здійснення торгівлі товарами чи послугами з використанням технічних і програмних можливостей: глобальної комп'ютерної мережі Internet.

Спершу клієнт переглядає перелік можливостей та функцій, розцінки на тарифні плани та умови співпраці що надає хостингова компанія та обирає більш відповідний для його вимог пакет. Потім реєструється на сайті, підтверджує обране замовлення, обирає спосіб оплати та здійснює платіжні операції. Сплата за послуги компанії тарифікується помісячно.

Усі операції, що до створення аккаунту (облікової записі) клієнта відповідно до його замовлення відбуваються системою автоматично. Додаткові операції, такі як, реєстрація доменного імені, встановлення SSL-сертифікату (англ. **Secure Sockets Layer** – рівень захищених сокетів) тощо, здійснює адміністратор. Окрім цього, адміністратор постійно стежить за своєчасною сплатою клієнтами свого тарифного плану і, у разі несплати, нагадує клієнту або деактивує аккаунт.

Клієнти, які вперше роблять замовлення, спочатку реєструються, вводячи свої особисті данні, а потім вже здійснюють замовлення. Наступного разу, коли клієнт буде міняти тарифний план, продовжувати існуючий або замовляти новий, йому потрібно буде лише пройти аутентифікацію і не треба буде вводити всі інші данні.

Для практичної реалізації схем електронної комерції необхідні підтримуючі механізми, насамперед ефективні системи платежів в Internet.

Саме платіжна система дає змогу перетворити Internet-службу з оброблення замовлень або електронну вітрину на повноцінну крамницю з усіма стандартними атрибутами вибравши товар або послугу на сайті продавця, клієнт може здійснити платіж, не відходячи від комп'ютера.

Усі платіжні Internet-системи, залежно від способу розрахунків, поділяють на:

- ✓ кредитні (які працюють з кредитними картками);
- ✓ дебетові (які працюють з електронними чеками і цифровою готівкою).

Кредитні електронні платіжні системи є аналогами звичайних систем, які працюють з кредитними картками. Різниця полягає лише в тому, що всі транзакції здійснюються через Internet, що потребує додаткових засобів безпеки та аутентифікації.

1.1.2 Опис аналогів інформаційної системи та її недоліків.

На даний час, аналог інформаційної системи не є повністю автоматизованою та універсальною.

Велика кількість хостингових компаній або немає автоматизованого управління клієнтським акаунтом на сервері без участі адміністратора системи або використовує обмежену кількість платіжних систем, що, в свою чергу, не дає змогу охопити увесь ринок цілком.

Управління клієнтськими акаунтами на сервері відбувається за допомогою панелі управління хостингом, яка реалізує доступ до всього серверу та всіх акаунтів. Інформаційна система є зовнішній програмний продукт з яким працює клієнт. Тому, вона взаємодіє з панеллю за допомогою API (**Application Programming Interface** – інтерфейс програмування приложень), які має панель. Цей набір готових класів, функцій, структур і констант, що надаються панеллю, дає можливість автоматизувати замовлення клієнта шляхом автоматичного створення (редагування) акаунту на сервері без участі адміністратора.

У здійсненні платежів через Internet з допомогою кредитних карток беруть участь наступні учасники.

- ✓ Покупець: клієнт, що має комп'ютер з Web-браузером і доступом до Internet.

✓ Банк-емітент: тут знаходиться розрахунковий рахунок покупця – картрахунок. Банк-емітент випускає картки і є гарантом виконання фінансових зобов'язань клієнта.

✓ Продавці: це сервери електронної комерції, на яких створені й підтримуються каталоги товарів і послуг та приймаються замовлення клієнтів на покупку.

✓ Банки-еквайєри: банки, які обслуговують продавців. Кожен продавець має єдиний банк, у якому відкрито його розрахунковий рахунок.

✓ Платіжна система в Internet: електронні компоненти, що є посередниками між іншими учасниками платіжної системи.

✓ Традиційна платіжна система: комплекс фінансових і технологічних засобів для обслуговування банківських карток певного типу. Її завдання забезпечення використання карток як засобів платежу за товари і послуги, користування банківськими послугами, здійснення взаємозаліків тощо, її учасниками є фізичні та юридичні особи, об'єднані відносинами з використання кредитних карток.

✓ Процесінговий центр платіжної системи: установа що забезпечує інформаційну та технологічну взаємодію між учасниками традиційної платіжної системи.

✓ Розрахунковий банк платіжної системи: кредитна установа, яка здійснює взаємозаліки між учасниками платіжної системи за дорученням процесінгового центру.

Процес здійснення платежів відбувається за наступною схемою.

1. Покупець в електронній крамниці формує кошик товарів і обирає засіб оплати кредитною картою.

2. Параметри кредитної картки (номер, ім'я власника, дата закінчення дії картки) повинні бути передані платіжній системі в Internet для подальшої авторизації. Це може бути зроблено двома шляхами:

а) параметри картки вводяться на сайт, після чого передаються платіжній системі в Internet;

б) параметри картки вводяться на сервері платіжної системи.

Другий шлях має перевагу: відомості про картки не залишаються на сайті компанії і, відповідно, знижується ризик отримання їх третіми особами або

шахрайства продавців. Однак і в тому, і в іншому разі під час передавання реквізитів кредитної картки все ж існує можливість їх перехоплення. Щоб уникнути цього, дані під час передавання кодуються.

3. Платіжна система в Internet передає запит на авторизацію традиційній платіжній системі.

4. Наступний крок залежить від того, чи підтримує банк-емітент базу даних (БД) рахунків в режимі on-line. За наявності он-лайнової бази даних рахунків у банку-емітенті, процесінговий центр передає йому запит на авторизацію картки і після цього отримує його результат. Якщо такої бази немає, то процесінговий центр сам зберігає відомості про стан рахунка власників карток, стоп-аркуші й виконує запити на авторизацію. Ці відомості регулярно оновлюються банками-емітентами.

5. Результат авторизації передається платіжній системі в Internet.

6. Сайт компанії одержує результат авторизації.

7. Клієнт одержує результат авторизації через сайт компанії або безпосередньо від платіжної системи в Internet.

8. У разі позитивного результату авторизації:

а) компанія надає послугу;

б) процесінговий центр передає в розрахунковий банк відомості про здійснену транзакцію; гроші з рахунка покупця в банку-емітенті перераховуються через розрахунковий банк на рахунок крамниці в банку-еквайєрі.

Отже, основою кредитних схем є використання кредитних карток. При разових покупках на електронному ринку кредитна пластикова картка діє так само, як під час звичайної покупки у крамниці: клієнт купує товар або послугу і передає продавцю для оплати номер своєї кредитної картки.

Основні переваги кредитних схем платежів в Internet:

- ✓ Звичність для клієнтів і правова визначеність.
- ✓ Висока захищеність конфіденційної інформації через використання протоколу SSL.

Недоліки кредитних систем:

- ✓ Необхідність перевірки кредитоспроможності клієнта та авторизації картки, що підвищує витрати на здійснення транзакції і робить системи

непристосованими для мікроплатежів (це є цільовим ринком платіжних систем в Internet).

✓ Відсутність анонімності, що зумовлює агресивність сервісної поведінки торговців.

Дебетові Internet-системи поділяють на два різновиди – електронні чеки та електронні гроші.

Електронні чеки – аналог звичайних паперових чеків, тобто доручення платника своєму банку перерахувати гроші зі свого рахунка на рахунок одержувача платежу.

На паперовому чеку платник ставить свій реальний підпис, а в он-лайновому варіанті – електронний підпис. Чеки видаються в електронному вигляді.

Здійснення платежів відбувається в декілька наступних етапів:

1. Платник виписує електронний чек, підписує електронним підписом і пересилає його одержувачу. Для забезпечення надійності та безпеки, номер чекового рахунка можна закодувати відкритим ключем банку.

2. Чек подається до оплати платіжній системі. Далі (або тут, або в банку, який обслуговує одержувача) відбувається перевірка електронного підпису.

3. У разі підтвердження його справжності, постачається товар або надається послуга. З рахунка платника гроші перераховують на рахунок одержувача.

Простота схеми здійснення платежів, на жаль, компенсується труднощами її впровадження в пострадянських країнах. Тут чекові схеми поки що не поширені.

Електронні гроші (електронна готівка) – система оплати товарів і послуг шляхом передавання числових даних від одного комп'ютера до іншого.

Електронні гроші цілком копіюють реальні гроші. При цьому емітент випускає електронні аналоги реальних грошей, які називаються в різних платіжних системах по-різному (наприклад, купони). Далі вони купуються користувачами, які з їх допомогою оплачують придбані товари і послуги, після чого продавець погашає їх у емітента. При емісії кожна грошова одиниця засвідчується електронною печаткою, яка перевіряється структурою, що їх випустила перед погашенням.

Усі номери цифрових грошей унікальні. Вони випускаються емітентом і представляють собою певну суму реальних грошей.

Слід зазначити, що при використанні електронних грошей немає необхідності в аутентифікації, оскільки система заснована на випуску грошей в обіг перед їх використанням.

1. Покупець заздалегідь обмінює реальні гроші на електронні. Зберігати електронну готівку клієнт може двома способами: на жорсткому диску комп'ютера; на смарт-картках. Смарт-картка – картка з інтегральною схемою і мікропроцесором, здатним виконувати розрахунки і зберігати електронні гроші. Зазначимо, що псування диска або смарт-картки обертається втратою електронних грошей.

Різні платіжні системи пропонують різні схеми обміну реальних грошей на електронні. Деякі відкривають спеціальні рахунки на які перераховуються кошти з рахунка покупця в обмін на електронні купюри. Деякі банки самі можуть бути емітентами електронної готівки. При цьому готівка емітується тільки на запит клієнта з наступним перерахуванням на комп'ютер або смарт-картку клієнта і зняттям грошового еквіваленту з його рахунку.

2. Покупець перераховує на сервер продавця електронні гроші за придбаний товар або послугу.

3. Гроші надаються емітенту, який перевіряє їх справжність.

4. У разі справжності електронних купюр рахунок продавця збільшується на суму купівлі, а покупцю відвантажується товар або надається послуга.

Однією з найважливіших характеристик електронних грошей є можливість здійснювати мікроплатежі. Це пов'язано з тим, що номінал купюр може не відповідати реальним монетам (наприклад, 37 копійок).

Головним чинником, що стримує розвиток проектів електронної комерції на основі смарт-карток, окрім відносно високої вартості самих карток, є необхідність придбання користувачем додаткового обладнання – карт-рідера, вартість якого кілька десятків доларів. Привабливим чинником є повне вирішення з допомогою смарт-карток проблеми захисту комерційної інформації під час платежів у глобальній мережі.

Перевага дебетових схем платежів в Internet у тому, що вони звільняють клієнта від оплати відсотків за кредит. У країнах, де домінують дебетові картки,

розрахунки на електронному ринку за дебетовими схемами вписуються до звичної моделі поведінки клієнтів, що дуже важливо.

Переваги дебетових платіжних систем з використанням електронних грошей:

- ✓ Зручні для здійснення мікроплатежів.
- ✓ Забезпечується необхідна анонімність платежів.

До недоліків цих схем можна віднести:

- ✓ Необхідність попередньої покупки купюр.
- ✓ Неможливість надання кредиту.

Недоліком дебетових схем платежів в Internet є те, що вони не підтримуються солідними фірмами (на відміну від систем з кредитними картками, за якими стоять такі гіганти, як VISA та MasterCard).

Дебетові платіжні системи, однак, успішно конкурують з кредитними. Вважається, що смарт-картки і електронні гроші в майбутньому якщо не домінуватимуть, то будуть вагомими в системі електронних платежів в Internet.

1.1.3 Обґрунтування необхідності вдосконалення аналогів інформаційної системи.

Аналоги інформаційної системи обмежать можливості компанії щодо виходу на міжнародний ринок внаслідок відсутності розширеного списку способів оплати. Розширення цього списку дасть змогу привернути більшу кількість клієнтів.

Відсутність автоматизованого виконання замовлення клієнта знижує ефективність роботи та потребує додаткових робочих місць для ручного адміністрування. Використання сучасних інформаційних технологій для автоматизації роботи інформаційної системи може вирішити цей недолік, підвищити ефективність і якість роботи системи, а також збільшити її функціональність.

Створення інформаційної системи яка дозволить реєструвати клієнтів, проводити перевірку всіх платежів, автоматично адмініструвати клієнтські аккаунти на сервері та своєчасно повідомляти клієнтів про необхідність подальшої оплати за послуги, тощо, не тільки позбавить компанію від необхідності розширювати штат робітників але й дасть можливість клієнту

працювати в режимі реального часу.

Для звучності та підвищення надійності спілкування необхідно також створити систему повідомлень, яка дозволить передавати їх між користувачами системи.

Такі можливості дадуть змогу підвищити рівень обслуговування клієнта, що в свою чергу позитивно відобразиться на іміджі компанії.

1.1.4 Цілі, критерії та обмеження створення інформаційної системи.

Метою створення інформаційної системи є автоматизація замовлень та розширення систем оплати.

Основні критерії створення інформаційної системи:

Соціальний критерій. Інформаційна система розрахована на користувачів з різною комп'ютерною підготовкою. Необхідний зручний та інтуїтивно зрозумілий інтерфейс для роботи з нею.

Економічний критерій. Автоматизована інформаційна система створюється для надання послуг хостингу. Перевагою користування системою буде те, що клієнту буде надана можливість вибору тієї системи оплати, з якої йому буде найбільш зручно працювати.

Критерій мобільності. Користувач повинний мати можливість працювати із системою з будь-якого комп'ютера, що має доступ в Інтернет.

Критерій безпеки. Система повинна бути безпечною з погляду збереження і обробки даних. Ушкодження або втрата даних може призвести до збоїв у роботі системи та втрати або розповсюдження комерційної інформації, тому необхідний захист системи від несанкціонованого втручання та збоїв.

1.1.5 Функції і задачі створюваної інформаційної системи.

Головною задачею створюваної системи є автоматизація роботи компанії, але система також повинна вирішувати наступні задачі та функції:

- ✓ реєстрації клієнтів в системі;
- ✓ створення клієнтського аккаунту на сервері;
- ✓ формування замовлень послуг, які неможна автоматизувати;
- ✓ облік клієнтів;
- ✓ автоматизація повідомлень клієнтам;

- ✓ взаємодію з платіжними системами;
- ✓ передача повідомлень всередині системи;
- ✓ відображення та формування динаміки замовлень;
- ✓ відображення та формування фінансових звітів.

1.1.6 Очікувані техніко-економічні результати створення підсистеми.

У результаті впровадження створюваної підсистеми передбачається усунути недоліки альтернативних інформаційних систем, що були описані в пункті 1.1.2.

При усуненні розглянутих недоліків очікуються наступні результати:

- ✓ підвищення ефективності контролюючих та управлінських дій;
- ✓ можливість виходу на міжнародний ринок;
- ✓ зниження вартості послуг компанії, що розширить коло потенційних клієнтів;
- ✓ висока швидкість роботи системи за рахунок потужного технічного забезпечення та якісного кодування системи;

1.1.7 Висновки і пропозиції

У результаті вивчення аналогів системи були виявлені недоліки. Для усунення цих недоліків необхідно модифікувати дану систему.

Для цього в підрозділі 1.1 були сформульовані мета, критерії й обмеження створення системи; функції і задачі створюваної системи; описані очікувані техніко-економічні результати створення системи.

1.2 Розробка концепції інформаційної системи.

1.2.1 Опис результатів вивчення об'єкта автоматизації.

Проект інформаційної системи хостинг-компанії розробляється для автоматизації процесу оплати та управління клієнтськими аккаунтами. Для створення системи потрібно розробити єдину структуру збереження даних о клієнтах, а також програмне забезпечення, необхідне для реалізації замовлень та контролю компанією здійснених продажів.

Розробка проекту інформаційної системи дозволить оптимізувати і підвищити швидкість виконання замовлення та розширить кількість способів

оплати, що, в свою чергу, збільшить кількість потенційних клієнтів.

Крім функціональних вимог до інформаційної системи необхідно врахувати вимоги до системи в цілому:

- ✓ простий і зручний інтерфейс користувача;
- ✓ доступ до системи через http-протокол з будь-якого комп'ютера мережі;
- ✓ багатокористувацька робота;
- ✓ масштабованість системи;
- ✓ висока продуктивність;
- ✓ надійність;
- ✓ дуже висока конфіденційність інформації та доступу до неї.

Аналізуючи зазначені вище вимоги, можна зробити висновок, що інформаційна система повинна мати або підтримувати:

- ✓ web-інтерфейс;
- ✓ архітектуру клієнт-сервер;
- ✓ сучасні стандарти в області інформаційних технологій (протоколи передачі даних, стандарти збереження даних і т.п.).
- ✓ сучасні серверні платформи і програмне забезпечення;
- ✓ сучасні рішення в області інформаційних технологій по розробці розподілених систем;
- ✓ ефективні методи доступу до даних;
- ✓ простоту модернізації, можливість повторного використання програмного забезпечення (коду).

1.2.2 Опис і оцінка варіантів концепції програмних засобів.

1.2.2.1 Вибір технологій для розробки інформаційної системи.

Зараз існують системи зі схожими функціями. Основні відмінності цих систем полягають у наборі функцій які надають ці системи.

З огляду на те, що система буде зберігати багато різноманітної інформації, а також, що система планується для роботи при великій кількості запитів, єдиним прийнятним засобом зберігання інформації буде СУБД.

У цілому технології створення систем поділяються на два класи: технології на стороні клієнта та технології на стороні сервера. У першу групу входять:

- ✓ елементи ActiveX, створені за допомогою Visual C++ або Visual Basic;

- ✓ аплети Java;
- ✓ сценарії, що виконуються на стороні клієнта, та динамічний HTML.

До технологій, що використовуються на стороні сервера, входять:

- ✓ CGI;
- ✓ фірмові API для веб-серверів, такі як ISAPI та NSAPI;
- ✓ Active Server Pages;
- ✓ Java Server Pages Java Servlets;
- ✓ Server-Side JavaScript;
- ✓ PHP.

У випадку обробки головної інформації на комп'ютерах клієнтів ("товстий" клієнт) до клієнтського технічного та програмного забезпечення висуваються дуже високі вимоги, що є непридатним у системі, яка призначення для взаємодії з великою кількістю клієнтів. Тому для даної системи більш придатним є випадок обробки основної інформації на стороні сервера, а на клієнтських місцях обробляти лише дані, що відносяться до інтерфейсу ("тонкий" клієнт), та створити цю обробку за допомогою найбільш розповсюдженого засобу.

Технології на стороні сервера.

Не так давно, єдиним практичним рішенням для розміщення в мережі динамічного змісту була технологія за назвою Common Gateway Interface (CGI – інтерфейс загального шлюзу). CGI-програми надавали порівняно простий спосіб створення веб-додатків. Вони одержували введені користувачем дані, виконували запит до бази даних і повертали результатів назад браузеру. Як Microsoft, так і Netscape розробили власні API, за допомогою яких для обслуговування веб-запитів можна було створювати код, що виконується в рамках того ж процесу (in-process). У число пропонованих технологій на стороні сервера входять Active Server Pages (ASP), Java Servlets і JavaServer Pages (JSP), хоча існує багато інших. Деякі з цих технологій більш докладно описуються і порівнюються нижче.

Інтерфейс загального шлюзу – Common Gateway Interface (CGI).

CGI є мережною технологією, що найбільш часто використовується. Ледве не кожен існуючий сьогодні веб-сервер має підтримку CGI-програм. CGI-програма може бути написана майже на будь-якій мові, хоча найчастіше для

програмування CGI використовується Perl. Веб-сервери, що реалізують CGI, діють як шлюз між запитом користувача і необхідних даних. Для цього спочатку створюється новий процес, у якому буде виконуватися програма. Потім у нього завантажується необхідне середовище часу виконання і сама програма. Нарешті, передається об'єкт, що представляє запит, і викликається програма. По закінченні роботи програми веб-сервер зчитує з стандартного виходу дані, що вона повернула.

Найбільш великим недоліком CGI-програмування є те, що воно погано масштабується. Для кожного одержаного веб-сервером запиту створюється зовсім новий процес. Кожен процес складається з власного набору змінних оточення, окремого екземпляра необхідного середовища етапу виконання, екземпляра програми і пам'яті, виділеної для використання програмою. Тому, коли одночасно приходить велика кількість запитів ресурси сервера піддаються великим навантаженням, що може викликати його відмовлення.

Тут приходять на допомогу такі технології, як FastCGI і mod_perl для Apache. Обидві технології намагаються вирішити проблеми продуктивності: FastCGI шляхом спільного використання екземплярів кожної CGI-програми, а mod_perl – шляхом інтерпретації і виконання сценаріїв Perl усередині веб-сервера Apache (зовсім обходячи, таким чином, CGI).

Фірмові API веб-серверів (ISAPI, NSAPI).

У відповідь на неефективність CGI, Microsoft і Netscape розробили власні API, що дозволяють розроблювачам створювати серверні додатки у виді бібліотек спільного доступу. Ці бібліотеки завантажуються в процес веб-сервера і можуть обслуговувати кілька запитів, не створюючи нових процесів. Вони можуть завантажуватися при запуску веб-сервера чи в міру необхідності в них. Якщо протягом визначеного часу вони не використовуються, веб-сервер вивантажує їх з пам'яті.

Хоча виконувані в рамках процесу бібліотеки є ефективним розширенням веб-сервера, з ними теж виникає ряд наступних проблем.

- ✓ Оскільки ці API специфічні для кожної платформи, написані з їхнім використанням програми можуть працювати тільки на відповідній платформі. Перенесення цих програм в інше середовище є дуже складною задачею.

- ✓ Оскільки доступ до цих бібліотек здійснюється одночасно декількома

користувачами, вони повинні бути написані із застосуванням методів забезпечення безпеки потокових перемінних (thread-safe). Це означає, що доступ до глобальних і статичних змінних повинен здійснюватися з великою обережністю.

✓ Якщо у виконуваний на сервері програмі відбувається порушення прав доступу, то, оскільки вона знаходиться в тім же процесі, що і сам веб-сервер, може відбутися аварія усього веб-сервера.

Active Server Pages (ASP).

ASP має подібність із PHP, сполучаючи HTML, сценарії і компоненти на стороні сервера в одному файлі, що має назву активна серверна сторінка (Active Server Page). Якщо сервер одержує запит файлу ASP, то спочатку шукає скомпільовану сторінку, а потім виконує її. Якщо сторінка ще не скомпільована, сервер компілює її, а потім виконує. Результатом файлу ASP є закінчена веб-сторінка, що повертається браузеру.

Active Server Page може бути написана з використанням HTML, Jscript (варіант JavaScript, створений Microsoft) і VBScript. Завдяки використанню сценаріїв, Active Server Page може мати доступ до компонентів на стороні сервера. Компоненти можна створювати на будь-якій мові, що надає інтерфейс до COM (специфікація компонентів, розроблена Microsoft). Безсумнівним недоліком активних серверних сторінок є те, що їх можна використовувати тільки з веб-сервером Microsoft (IIS, PWS) під операційними системами Microsoft (Win9x, WinNT). Існують переноси на інші платформи і веб-сервери, але відсутність широкої підтримки COM знижує їхню ефективність.

JavaScript на стороні сервера (SSJS).

JavaScript на стороні сервера став відповіддю Netscape на ASP. Подібно Active Server Pages, сторінки, що використовують SSJS, складаються з коду HTML у який вбудовані секції зі сценаріями, виконуваними на стороні сервера. Цей код виконується на сервері і генерує веб-сторінку з чистого HTML, що відправляється браузеру.

SSJS має ту перевагу, що використовує JavaScript – стандартну мережну мову. Однак є невеликий недолік у порівнянні з ASP і PHP, що полягає у тому, що додатки, що використовують SSJS, повинні бути скомпільовані перед запуском на виконання. А це ускладнює процес модифікації сторінок SSJS.

Однак більш серйозний недолік полягає, у тім, що SSJS тепер підтримується тільки Netscape Enterprise Server, який значно уступає Apache і Microsoft IIS по поширеності використання в Інтернеті. Це різко обмежило інтерес до SSJS.

Java Servlets і JSP.

Java-сервлети (Java Servlets) і JavaServer Pages (JSP) є технологіями, що застосовують на стороні сервера і використовують мову Java. Корпорація Sun представила ряд нових API, що дозволяють підключати програми до корпоративних служб і даних. Java Servlet API є одним з наріжних каменів цих розширень платформи Java, надаючи могутній засіб створення динамічного змісту і розширення функціональності веб-серверів.

Сервлет Java є програмою, що виконується на стороні сервера, що обслуговує запити HTTP і повертає результати у виді HTML сторінки. У цьому відношенні вона дуже схожа на CGI, але на цьому подібність і закінчується. Гарною аналогією сервлета є невидимий аплет, виконуваний на веб-сервері. Його життєвий цикл аналогічний життєвому циклу аплета, а виконується він усередині віртуальної машини Java (JVM),

JavaServer Pages (JSP) аналогічні активним серверним сторінкам Microsoft (ASP). Сторінка JavaServer містить HTML, код Java і компоненти JavaBean. JSP дозволяє вбудовувати в сторінки компонента і забезпечивши виконання ними роботи з генерації сторінки, що посилається в остаточному підсумку клієнту. При запиті користувачем файлу JSP веб-сервер спочатку генерує відповідний сервлет, якщо його ще не існує. Потім у сервер запускає сервлет і повертає створене їм зміст браузеру

JavaServer Pages і Java Servlets страждають тим же недоліком, що і мова Java, використовуваний на стороні клієнта: Java відносно складний для вивчення програмістами-новачками.

PHP.

PHP працюють аналогічно JSP і ASP: розділи сценарію полягають у теги `<?php..?>` і вбудовуються в сторінку HTML. Ці сценарії виконуються на сервері перед тим, як послати сторінку браузеру, тому проблеми підтримки браузером PHP не існує. Однак, на відміну від ASP, PHP є незалежним від платформи, і існують різні версії для різних варіантів Windows, Unix і Linux, а також ряду веб-серверов, у тому числі Apache і IIS. Визначальне значення має те, що він є

безкоштовним продуктом з відкритим кодом (open source).

Сьогодні PHP використовується на багатьох комерційних сайтах.

Переваги обробки на стороні сервера.

Обробка і генерація веб-сторінок на стороні сервера має ряд переваг перед технологіями, використовуваними тільки на стороні клієнта:

- ✓ зменшується потік даних у мережі в результаті зменшення необхідності багаторазового спілкування браузера і сервера один з одним;
- ✓ сприяє зменшенню часу завантаження, оскільки в підсумку фактично завантажується тільки сторінка HTML;
- ✓ усуваються проблеми сумісності браузерів;
- ✓ клієнту можуть поставлятися дані, що відсутні на його машині;
- ✓ забезпечується кращий захист даних, оскільки код неможливо переглянути браузером.

Оскільки розроблювальна система представляє собою "тонкого" клієнта, то в цьому випадку доцільно буде використовувати технологію, що забезпечує доступ до даних на стороні сервера, а для створення зручного інтерфейсу користувача використовувати найбільш поширені технології, що виконуються на стороні клієнта.

1.2.2.2 Розробка варіантів концепції програмних засобів.

Виходячи з аналізу вимог до інформаційної системи, її функціональних характеристик, фізичної архітектури і технологій для розробки, можна розробити концепцію програмних засобів інформаційної системи. Основними компонентами системи є:

- ✓ серверна платформа і серверне програмне забезпечення – один з основних факторів ідеології організації інформаційної системи;
- ✓ система управління базами даних;
- ✓ клієнтська платформа і клієнтське програмне забезпечення;
- ✓ мережне середовище;
- ✓ засоби розробки.

Від вибору перерахованих компонентів багато в чому залежить ефективність інформаційної системи. Очевидно, що для вирішення подібної задачі може існувати кілька варіантів.

Серверна платформа і серверне програмне забезпечення.

Для реалізації подібних систем найчастіше використовують платформи UNIX чи серверні варіанти Windows. Інформаційна система має дуже високі вимоги до функціональності операційної системи, тому для її реалізації підійдуть стабільні версії UNIX систем та серверні платформи Windows. У зв'язку з тим що система потребує високої надійності та стійкості, доцільним буде з UNIX систем виділити FreeBSD та Linux які зараз є найбільш популярними та стабільними у роботі та безкоштовні у використанні.

Всі названі системи мають різноманітні конфігурації, але так як системи FreeBSD і Linux безкоштовні, їхня конфігурація практично не буде впливати на вартість системи.

У якості http-сервера може бути використаний один з найбільш популярних серверів Apache.

Сервер Apache розроблений Open Source Foundation більш пристосований до систем Unix і як більшість продуктів Open Source – безкоштовний.

Система управління базами даних.

Вибір сервера баз даних дуже великий, але для даної інформаційної системи необхідна висока продуктивність і краща масштабованість. З популярних на сьогоднішній час СУБД можна вибрати наступні:

- ✓ MySQL компанії MySQL AB;
- ✓ MS SQL Server компанії Microsoft;
- ✓ PostgreSQL 8.3.

При цьому варто врахувати деякі особливості: розроблювальна система повинна бути багато платформною (тобто працювати як під управлінням Unix-подібних систем, так і під Windows), а також те, що не існує реалізації СУБД PostgreSQL для Windows також, як і реалізації MS SQL для Unix-систем, тому ми оберемо MySQL.

Засоби розробки.

Для створення системи, у якості засобу розробки можна взяти наступні мови програмування:

- ✓ Visual C++
- ✓ Visual Basic
- ✓ Active Server Pages

- ✓ C#
- ✓ JavaServer Pages Java Servlets
- ✓ Server-Side JavaScript
- ✓ Borland Delphi
- ✓ PHP
- ✓ Perl
- ✓ Active Pearl
- ✓ Gcc

Більшість цих мов розглянута у пункті 1.2.2.1.

Клієнтська платформа і клієнтське програмне забезпечення.

Для клієнтських робочих місць можна використовувати операційні системи MS Windows XP/Vista/7/7 Professional, клони UNIX, OS/2 і інші. Основна вимога – наявність веб-браузера, що підтримує HTML і JavaScript (наприклад, Інтернет Explorer, Netscape Navigator, Mozilla/FireFox/SeaMonkey,Opera).

Мережне середовище

Веб-сервер повинен бути розміщений на вузлі провайдера послуг Internet і мати швидкий доступ до нього. Сервер бази даних доцільно розмістити у віртуальній локальній мережі з веб-сервером та встановленим зв'язком між ними. Для покращення безпеки системи на сервері бази даних доцільно дозволити лише шифровані з'єднання для адміністрування сервера.

Технічні засоби

Технічні засоби сервера повинні задовольняти вимогам операційної системи і СУБД до устаткування. Конфігурація сервера може бути наступною:

Технічне забезпечення веб-сервера:

- ✓ процесор: Intel® Core™ i7-920 Quad-Core 4 фізичних ядра, 4 віртуальних, підтримка технології hyper-threading;
- ✓ дискова система: 1500 GB HDD SATA2;
- ✓ оперативна пам'ять: 12 GB DDR3 5;
- ✓ мережна карта Ethernet, що підтримує швидкості передачі 1000 Мбіт/сек.
- ✓ наявність бекап-серверу.

Технічні засоби клієнтських комп'ютерів повинні мати будь-яку операційну систему з веб-браузером. Конфігурація комп'ютера користувача може бути

наступною:

- ✓ мікропроцесор: Intel® Core™ i3;
- ✓ оперативна пам'ять RAM 2048 Mb;
- ✓ жорсткий диск: HDD 300 Gb;
- ✓ мережна карта Ethernet.

Ґрунтуючись на ступені поширеності і необхідних функціональних можливостей, визначаємо варіанти концепції програмних засобів у вигляді таблиці 1.2.

Таблиця 1.2 – Варіанти концепції програмних засобів.

№	ОС веб-сервера	СУБД	Веб-сервер	Мова програмування
1.	MS Windows Server 2008	MS SQL 2003	MS IIS 7.0	C#
2.	Free BSD 8.0	MySQL 5	Apache 2.0	PHP
3.	Ubuntu Linux	PostgreSQL 9	Apache 2.0	Perl

1.2.3 Опис і оцінка недоліків розроблених альтернативних варіантів концепції створення системи.

На даний час, концепції альтернативних інформаційних систем не є повністю автоматизовані та універсальні.

Більшість концепцій альтернативних інформаційних систем або використовують обмежену кількість платіжних систем, що, в свою чергу, не дає змоги охопити увесь ринок цілком або не мають автоматизованого управління клієнтським аккаунтом на сервері без участі адміністратора системи.

У першому випадку, зображеному на рисунку 1.2, інформаційна система має обмежену кількість варіантів оплати замовлення, тому клієнту доводиться відкривати додатковий рахунок в банку та замовляти додаткову карту або навіть кожен раз йти до банку щоб зробити банківський переказ на рахунок компанії. Такий підхід спричиняє відчутну затримку виконання замовлення та деякі незручності.

У другому випадку, зображеному на рисунку 1.3, інформаційна система не має автоматизованої взаємодії з системою управління сервером. Після оплати клієнтом замовлення, воно потрапляє в режим очікування обробки адміністратором. Коли адміністратор бачить надходження нового замовлення – він власноруч створює аккаунт в системі управління сервером, а вже після

цього пропоставляє замовленню клієнта статус виконання. В цій ситуації також клієнт втрачає можливість роботи з системою в режимі реального часу.



Рисунок 1.2 – Концепція системи з обмеженим вибором здійснення платежів

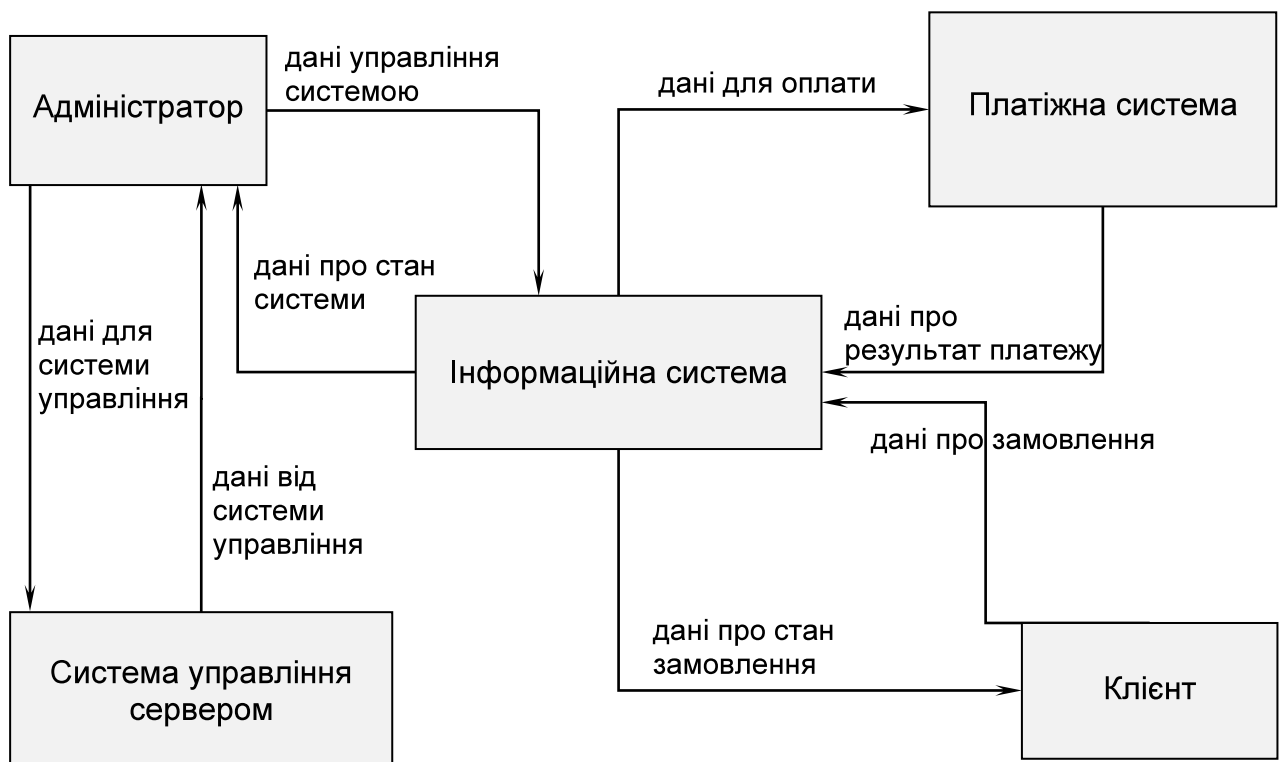


Рисунок 1.3 – Концепція системи без взаємодії з системою управління сервером

Таким чином необхідно вдосконалити наведені вище альтернативні концепції системи шляхом реалізації розвиненої системи оплати та впровадження автоматизованої взаємодії системи з системою управління сервером. Така концепція (див. рисунок 1.1) дозволить клієнтові отримати виконання замовлення в режимі реального часу.

1.2.4 Функціональний аналіз.

Для вибору найбільш ефективного варіанта концепції інформаційної системи можна використати метод функціонально-вартісного аналізу.

В основі методу лежить функціональний підхід, згідно з яким об'єктом аналізу є не сам виріб (система, програмне забезпечення), а функції, що він виконує.

Функціонально-вартісний аналіз складається з двох основних етапів: функціональний аналіз і вартісний аналіз.

На першому етапі функціонального аналізу необхідно визначити ступінь відповідності кожного варіанта концепції підсистеми основним характеристикам даної підсистеми.

Для того, щоб оцінити різні варіанти необхідно визначити основні параметри системи. Основними параметрами системи є:

- ✓ продуктивність технічних та програмних засобів (швидкодія і час відгуку, споживання ресурсів);
- ✓ надійність технічних та програмних засобів (стійкість до відмов, стійкість до користувальницьких помилок, здатність відновлюватися після збоїв);
- ✓ масштабованість (здатність до збільшення функціональних можливостей та обсягів даних, що обробляються);
- ✓ придатність до вивчення та використання (зрозумілість, зручність і простота в роботі);
- ✓ придатність для подальшого супроводження (можливості аналізувати та діагностувати причини помилок, придатність до змін та до тестування, стабільність).

Жоден з цих параметрів не має одиниці виміру. Не існує також стандартної методики визначення їхніх чисельних значень. Таким чином, необхідно

одержати експертні оцінки по кожному варіанту системи і відповідним параметрам. Осереднені експертні оцінки (10 бальна шкала) по кожному варіанту системи і відповідним параметрам представлені в таблиці 1.3.

Таблиця 1.3 – Основні параметри підсистеми

Параметр, F_i	Оцінка в балах по номеру варіанта, B_i		
	1	2	3
Надійність технічних та програмних засобів	10	10	8
Продуктивність технічних та програмних засобів	9	9	8
Масштабованість	8	8	7
Придатність до вивчення та використання	8	9	8
Придатність для подальшого супроводження	7	8	8

Визначимо вагові коефіцієнти для кожного з параметрів, які також вибираються експертним шляхом. Значення вагових коефіцієнтів для кожного з параметрів приведені в таблиці 1.4.

Таблиця 1.4 – Вагові коефіцієнти для параметрів системи

Параметр, F_i	Ваговий коефіцієнт, φ_i
Надійність технічних та програмних засобів	0,35
Продуктивність технічних та програмних засобів	0,25
Масштабованість	0,20
Придатність до вивчення та використання	0,10
Придатність для подальшого супроводження	0,10

Визначимо коефіцієнти технічного рівня для кожного з параметрів по кожному варіанту системи за формулою $K_{тр}(F_i) = B_i \cdot \varphi_i$ у вигляді таблиці 1.5.

Таблиця 1.5 – Розрахунок показників технічного рівня системи

Параметр, F_i	Варіант	Оцінка параметра в балах, B_i	Ваговий коефіцієнт, φ_i	Показник технічного рівня, $K_{тр}(F_i)$
Надійність технічних та програмних засобів	1	10	0,35	3,50
	2	10		3,50
	3	8		2,80
Продуктивність технічних та програмних засобів	1	9	0,25	2,25
	2	9		2,25
	3	8		2,00
Масштабованість	1	8	0,20	1,60

	2	8		1,60
	3	7		1,40

Продовження таблиці 1.5

Придатність до вивчання та використання	1	8	0,10	0,80
	2	9		0,90
	3	8		0,80
Придатність для подальшого супроводження	1	7	0,10	0,70
	2	8		0,80
	3	8		0,80

За даними таблиці 1.5 розраховується коефіцієнт технічного рівня для кожного з варіантів системи за формулою $K_{\text{тр}} = \sum_{i=1}^5 K_{\text{тр}}(F_i)$ у вигляді таблиці 1.6.

Таблиця 1.6 – Коефіцієнти технічного рівня

Варіант	Коефіцієнт технічного рівня
1	8,85
2	9,05
3	7,80

Найкращим на етапі функціонального аналізу є варіант, що має максимальне значення коефіцієнта технічного рівня, тобто варіант №2.

1.2.5 Вартісний аналіз.

Аналіз функцій, що реалізуються системою, повинен бути доповнений вартісним аналізом. Для цього розглядаються зведені витрати Z_i по кожному варіанту системи за формулою:

$$Z_i = C_i + E_n \cdot K_i,$$

де C_i – поточні витрати по i -му варіанту;

K_i – капітальні вкладення по i -му варіанту;

E_n – нормативний коефіцієнт економічної ефективності капітальних вкладень, що дорівнює 0,15.

Зведені витрати Z_i розраховують у сфері виробництва (у розроблювача) або в сфері експлуатації (у користувача).

У нашому випадку доцільно розраховувати витрати в сфері виробництва, при цьому поточні витрати при розробці C_i – це собівартість створення системи C_c , яку можна обчислити за формулою:

$$C_c = C_z + C_{\text{вiд}} + C_m + C_n$$

де C_z – заробітна плата розроблювачів;

$C_{\text{вiд}}$ – відрахування на соціальні заходи (47,5% від фонду оплати праці);

C_m – вартість машинної години, необхідної для розробки та налагодження системи;

C_n – накладні витрати в розмірі 50-100% від витрат на оплату праці (містять оплату праці управлінського персоналу з нарахуванням оплат службових відряджень; оплату консультативно-інформаційних послуг; оплату ремонту і техобслуговування основних фондів, крім ПК; оплату оренди приміщення; тощо).

Витрати на оплату праці розробника системи визначається за формулою

$$C_z = C_z \cdot T,$$

де C_z – погодинна оплата праці програміста;

T – трудомісткість розробки системи.

Вартість машинного часу визначається за формулою

$$C_{m1} = t_m \cdot C_{m-2},$$

де t_m – кількість машинних години, необхідних для розробки (рік.);

C_{m-2} – собівартість 1 машино-години роботи ПК (грн.):

$$C_{m-2} = \frac{\sum_i U_i}{T_{\text{ef}}},$$

де U_i – річні експлуатаційні витрати при обслуговуванні ПК, що включають зарплату фахівця, що обслуговує машину (з поправкою на коефіцієнт зайнятості працівника обслуговуванням ПК); нарахування на зарплату (37,5%); витрати на електроенергію, що визначаються як добуток вартості 1 кВт·год.

електроенергії на потужність, яку споживає ПК, і на ефективний фонд години ПК (за рік); накладні витрати – 50-150% від витрат на оплату праці;

T_{ef} – ефективний фонд години роботи ПК за рік, що визначається виходячи з календарного річного фонду години, крім вихідних і святкових днів.

Таким чином, маємо наступні результати розрахунків собівартості створення системи по варіантах:

Варіант 1	Варіант 2	Варіант 3
$C_3 = 3520$ грн.;	$C_3 = 3203,2$ грн.;	$C_3 = 3203,2$ грн.;
$C_{vid} = 1672$ грн.;	$C_{vid} = 1521,5$ грн.;	$C_{vid} = 1521,5$ грн.;
$C_M = 6065,5$ грн.;	$C_M = 6065,5$ грн.;	$C_M = 6065,5$ грн.;
$C_H = 2640$ грн.;	$C_H = 2402,4$ грн.;	$C_H = 2402,4$ грн.;
$C_{c2} = 13897,5$ грн.	$C_{c2} = 13192,6$ грн.	$C_{c2} = 13192,6$ грн.

Капітальні вкладення K_i – це витрати, що носять одноразовий характер. В нашому випадку вони складаються з вартості ліцензійного програмного забезпечення та витрати на придбання серверу.

Вартість необхідного програмного забезпечення наведена в таблиці 1.7.

Таблиця 1.7 – Вартість програмного забезпечення

Варіант	Найменування ПО	Вартість, грн.	Кількість	Разом, грн.
1	MS Windows Server 2008 (Enterprise)	21875	1	21875
	MS SQL 2003	2183	1	2183
	MS IIS 7.0	0	1	
	Microsoft Visual Studio (C#)	1581	1	1581
2	Free BSD 8.0	0	2	0
	MySQL 5	0	1	0
	Apache 2.0	0	1	0
	PHP	0	1	0
3	Ubuntu Linux	0	2	0
	PostgreSQL 9	0	1	0
	Apache 2.0	0	1	0
	Perl	0	1	0

Вартість придбання сервера – 6150 грн.

Капітальні вкладення по кожному варіантові програмного забезпечення наведено в таблиці 1.8.

Таким чином, можна розрахувати зведені витрати по кожному варіанту. Результати розрахунку наведені в таблиці 1.9.

Таблиця 1.8 – Капітальні вкладення по варіантах

Варіант	Вартість, грн.
1	36434
2	8479
3	8479

Таблиця 1.9 – Зведені витрати по варіантах

Варіант	Вартість, грн.
1	50332
2	21672
3	21672

За результатами функціонального й вартісного аналізу можна розрахувати коефіцієнт техніко-економічного рівня по кожному з варіантів по формулі:

$$K_{\text{тері}} = K_{\text{трі}} / 3_i$$

Результати розрахунку наведено у таблиці 1.8.

Таблиця 1.10 – Коефіцієнти техніко-економічного рівня

Варіант	$K_{\text{тері}}$
1	$1,758 \cdot 10^{-4}$
2	$4,175 \cdot 10^{-4}$
3	$3,599 \cdot 10^{-4}$

Найкращий варіант визначається за максимальним значенням коефіцієнта $K_{\text{тері}}$. Таким чином, визначаємо, що найкращий варіант – це варіант 2.

1.3 Постановка задачі.

Назва розробки: "Управління замовленнями хостинг-компанії MyHost".

Підставою для розробки інформаційної підсистеми є завдання на дипломну роботу "Розробка інформаційної системи хостингової-компанії MyHost".

Призначенням системи є автоматизація функцій, що виконують адміністратори компанії при надходженні замовлень та розширення переліку

способів оплати.

Метою створення системи є підвищення ефективності і якості роботи компанії по наданню послуг.

Інформаційна система повинна володіти рядом наступних функцій.

1. Користування клієнтами:

а) реєстрація клієнтів

- ✓ вхідні дані: особисті дані клієнта, поштова адреса;
- ✓ вихідні дані: результат реєстрації;

б) створення замовлень

- ✓ вхідні дані: обраний пакет, код клієнта;
- ✓ вихідні дані: результати замовлення;

в) аутентифікація клієнтів

- ✓ вхідні дані: логін, пароль;
- ✓ вихідні дані: результат авторизації;

г) вибір клієнтом систем оплати

- ✓ вхідні дані: код системи оплати;

д) перегляд клієнтом переліку замовлень та інформування про оплату замовлень

- ✓ вхідні дані: номер замовлення;
- ✓ вихідні дані: переліз замовлень;

е) зміна клієнтом стану замовлення

- ✓ вхідні дані: номер замовлення;
- ✓ вихідні дані: результат зміни стану.

2. Надання системи передачі повідомлень:

а) можливість створювати повідомлення для користувачів, що зареєстровані у системі

- ✓ вхідні дані: код клієнта, тема, текст повідомлення;

б) перегляд повідомлень

- ✓ вхідні дані: номер повідомлення;
- ✓ вихідні дані: код клієнта, тема, текст повідомлення.

3. Взаємодія з системою управління хостингом:

а) автоматичне створення аккаунта клієнта в відповідності до замовленого пакету

- ✓ вхідні дані: номер клієнта, номер пакету;
- ✓ вихідні дані: результат створення;
- б) редагування аккаунту клієнта
 - ✓ вхідні дані: номер клієнта, дані клієнта.
- 4. Адміністрування:
 - а) надання можливості адміністратору створювати нові пакети та редагувати або видаляти існуючі
 - ✓ вхідні дані: номер пакету, назва та параметри пакету;
 - ✓ вихідні дані: результат створення пакету;
 - б) надання можливості адміністратору переглядати замовлення клієнтів та здійснювати контроль оплати
 - ✓ вхідні дані: номер замовлення;
 - ✓ вихідні дані: статус та параметри замовлення;
 - в) адміністрування клієнтських аккаунтів
 - ✓ вхідні дані: номер клієнта, дані клієнта.

Система повинна бути відкритою, тобто мати властивості мобільності.

Система повинна підтримувати розподілену дворівневу архітектуру клієнт-сервер і мати зручний веб-інтерфейс, який забезпечує доступ до інформації через мережу Інтернет.

Серверні додатки повинні генерувати документи і форми для відображення необхідної інформації мовою HTML. Генерація документів повинна здійснюватися на основі інформації, що зберігається у базі даних.

Система повинна забезпечувати захист даних шляхом аутентифікації й ідентифікації користувачів, захисту каналів зв'язку (шифрування даних), фізичного захисту устаткування.

В якості сервера бази даних необхідно використовувати серверну СУБД із наявністю засобів веб-доступу до баз даних у сполученні з відносно невисокою ціною рішення. Необхідно використовувати стандартні можливості СУБД із метою підтримки переносимості.

Веб-інтерфейс повинен бути реалізований мовою HTML.

Функціональність на стороні клієнта ("тонкий» клієнт") повинна розроблятися мовою JavaScript.

Серверні додатки необхідно розробляти мовою PHP.

Інформаційна система включає веб-сервер, сервер баз даних та клієнтські комп'ютери.

Необхідна наявність наступних документів:

- ✓ загальний опис системи;
- ✓ посібник користувача-адміністратора;
- ✓ посібник користувача-клієнта;
- ✓ опис структури програмного забезпечення і бази даних;
- ✓ інструкція з адміністрування бази даних;
- ✓ програма і методика випробувань;
- ✓ текст програми.

Перевірка документації інформаційної системи робиться замовником з метою зафіксувати відповідність чи невідповідність системи технічному завданню.

Тестування системи виконує розроблювач протягом створення системи без взаємодії з замовником. Процес тестування представляє собою експлуатацію системи в контрольованих умовах і вивчення отриманих результатів. При цьому перевіряється робота системи з нормальними і помилковими даними і подіями. Тестування розподілених систем ускладнене більшою кількістю взаємозв'язків. Кожна залежність, веб-сторінка, компонент, база даних, а також код графічного інтерфейсу, програмне забезпечення середнього рівня і мережна інфраструктура повинні бути протестовані не тільки на коректність роботи, але і на сумісність один з одним у різних конфігураціях.

Найкраще для тестування розподілених систем застосовувати метод "знизу-нагору". Усі ці види тестування виконуються розроблювачем без участі замовника.

Приймання системи виконується замовником. Система вважається придатною, якщо вона задовольняє всім пунктам даного технічного завдання. Замовник і (або) його представники самостійно проводять тестування усіх функцій системи методом "чорної шухляди" у присутності розроблювача. При виявленні помилок у роботі системи складається їхній перелік і обговорюється термін їхнього виправлення розроблювачем. Після цього замовник проводить повторне тестування.

Після виправлення помилок або у випадку, якщо помилки не були

знайдені, а також при відповідності системи встановленим у технічному завданні вимогам, підписується акт приймання системи.

2. Розробка проектних рішень інформаційної системи хостингової-компанії.

2.1 Загальносистемні рішення.

2.1.1 Схема організаційної структури.

Схема організаційної структури містить: склад підрозділів (посадових осіб), що забезпечують функціонування системи хостингової-компанії.

Орієнтовна схема організаційної структури компанії представлена на рисунку 2.1.

Користувачами системи з боку компанії буде адміністратор. Інформаційно-технічний відділ виконує роботи з підтримки працездатності системи.

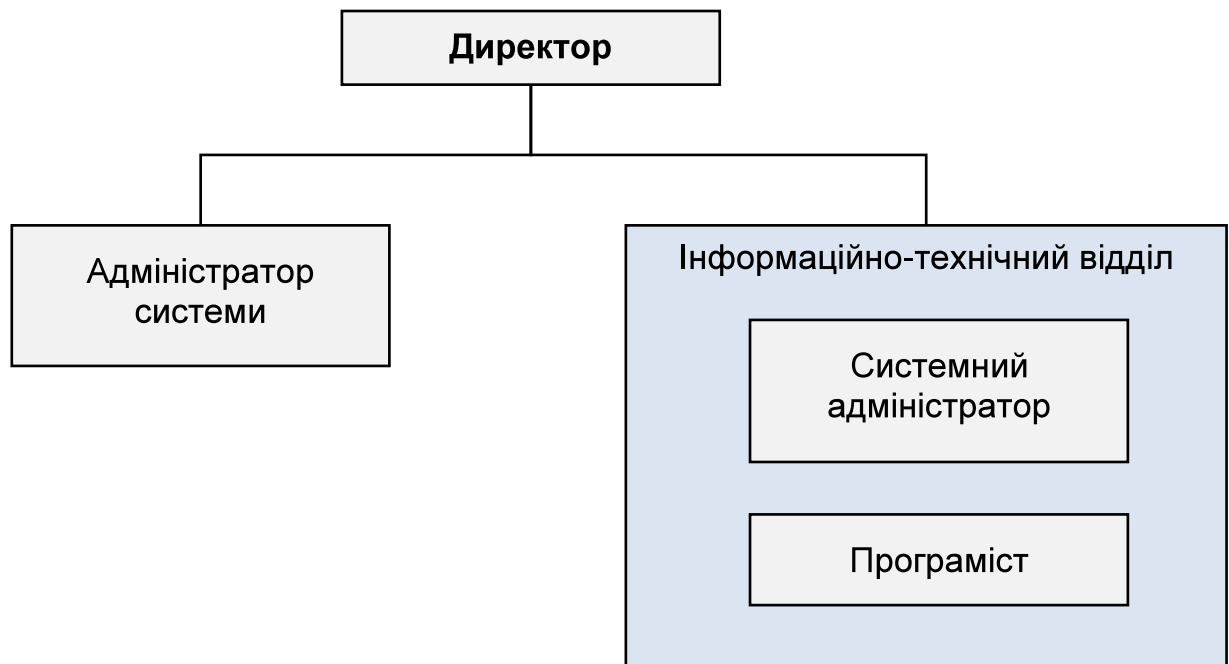


Рисунок 2.1 – Схема організаційної структури

2.1.2 Схема функціональної структури.

Для уточнення функціональності інформаційної системи, виявлення схеми функціональної структури, визначення вхідних у її склад підсистем (модулів) і вивчення динаміки роботи системи в цілому необхідно побудувати функціональну модель.

Функціональна модель описує обчислення в системі. Вона показує, яким

чином вихідні дані обчислюються по вхідним даним, не розглядаючи порядок чи спосіб реалізації обчислень. Функціональна модель складається з набору діаграм потоків даних, які показують потоки значень від зовнішніх входів через операції і внутрішні сховища даних до зовнішніх виходів.

Діаграми потоків даних (Data Flow Diagrams – DFD) є основним засобом моделювання функціональних вимог до проектованої системи. З їхньою допомогою ці вимоги розбиваються на функціональні компоненти (процеси) і представляються у вигляді мережі, зв'язаної потоками даних. Головна мета таких засобів – продемонструвати, як кожен процес перетворить свої вхідні дані у вихідні, а також виявити відносини між цими процесами.

Джерела інформації (зовнішні сутності) породжують інформаційні потоки (потоки даних), які переносять інформацію до підсистем або процесів. Ті у свою чергу перетворюють інформацію і породжують нові потоки, що переносять інформацію до інших процесів або підсистем, накопичувачам даних або зовнішнім сутностям – споживачам інформації. Таким чином, основними компонентами діаграм потоків даних є:

- ✓ зовнішні сутності;
- ✓ процеси;
- ✓ накопичувачі даних;
- ✓ потоки даних.

Зовнішня сутність це матеріальний предмет чи фізична особа, яка представляє собою джерело або приймач інформації. Для розроблювальної системи зовнішніми сутностями є:

- ✓ адміністратори;
- ✓ клієнти;
- ✓ платіжні системи;
- ✓ система управління сервером.

Адміністратори системи формують пакети послуг, що надає компанія, редагують клієнтські дані та замовлення. Від системи адміністратори отримують статистичні дані про клієнтів та їх замовлення.

Клієнти, при реєстрації, надають системі особисті дані та дані про послуги, які вони бажають замовити. Від системи клієнти отримують дані про стан виконання їхніх замовлень.

Платіжна система є зовнішньою інформаційною системою, яка здійснює переміщення грошей від клієнта до компанії. До платіжної системи надходять дані про товари або послуги, які замовляє клієнт, та їхня загальна вартість. Після оплати визначеної суми, платіжна система передає дані компанії про здійснення платежу.

Система управління сервером є зовнішньою інформаційною системою, яка створює та управляє клієнтськими аккаунтами на сервері. Отримує дані про клієнта та його пакет послуг. На виході отримуються дані про стан клієнтських аккаунтів.

Процес обробки даних системою, представляє собою перетворення вхідних потоків даних у вихідні відповідно до визначеного алгоритму. Фізично процес може бути реалізований різними способами.

Для розроблювальної системи основні процеси на DFD пов'язані з аутентифікацією адміністраторів та клієнтів, управлінням пакетами послуг, функціонуванням системи контролю замовлень, системою внутрішніх повідомлень між користувачами системи, вибором та оплатою послуг клієнтом, взаємодією системи з системою платежів та з системою управління сервером.

Сховище даних представляє собою абстрактний пристрій для збереження інформації, яку можна в будь-який момент помістити в сховище і через якийсь час витягти, причому способи поміщення і витягання можуть бути будь-якими. Сховищем даних, у системі що розроблюється, є база даних.

Потік даних визначає інформацію, передану через деяке з'єднання від джерела до приймача. Реальний потік даних може бути інформацією, яка передається по кабелю між двома пристроями, які листуються електронною поштою, а також які можуть бути перенесені з одного комп'ютера на інший.

Важливу специфічну роль у моделі грає спеціальний вид діаграми потоків даних – контекстна діаграма, яка моделює систему найбільш загальним чином. Контекстна діаграма відбиває інтерфейс системи з зовнішнім світом, а саме, інформаційні потоки між системою і зовнішніми сутностями, з якими вона повинна бути зв'язана. Контекстна діаграма системи представлена на рисунку 2.2.

Позначення потоків даних на даній контекстній діаграмі наведено у таблиці 2.1.



Рисунок 2.2 – Контекстна діаграма потоків даних

Таблиця 2.1 – Позначення потоків даних на контекстній діаграмі

Умовне позначення	Потік даних
1	Данні для системи управління сервером
2	Данні від системи управління сервером
3	Данні від платіжної системи
4	Данні для платіжної системи
5	Данні для клієнта
6	Данні від клієнта
7	Данні від адміністратора
8	Данні для адміністратора

Після декомпозиції контекстної діаграми потоків даних отримана діаграма потоків даних першого рівня, яка представлена на рисунку 2.3.

Позначення потоків даних на діаграмі потоків даних першого рівня наведено у таблиці 2.2.

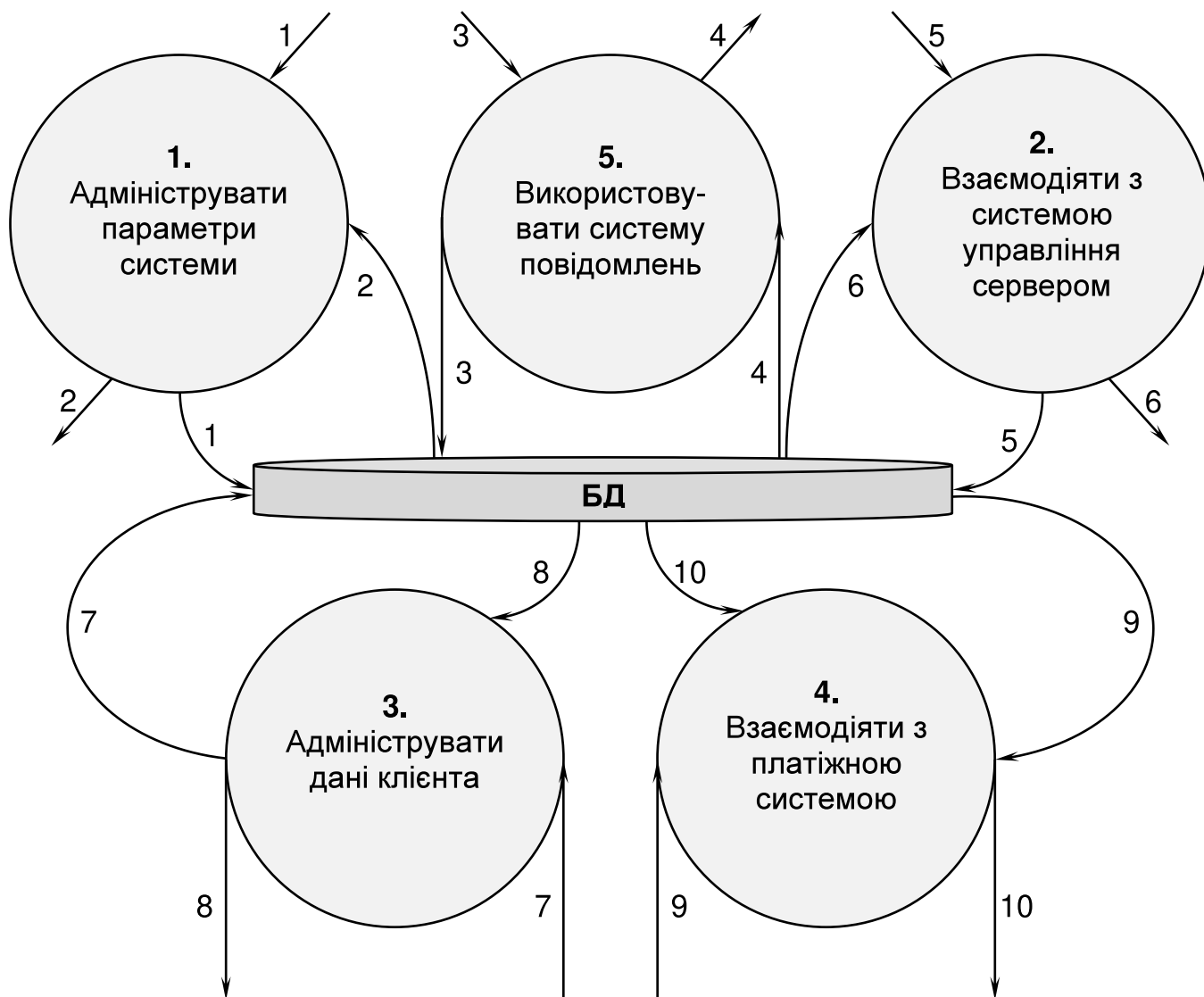


Рисунок 2.3 – Діаграма потоків даних першого рівня

Таблиця 2.2 – Позначення на діаграмі потоків даних першого рівня.

Умове позначення	Потік даних
1	Дані від адміністратора
2	Дані для адміністратора
3	Дані від користувачів
4	Дані для користувача
5	Дані від системи управління сервером
6	Дані для системи управління сервером
7	Дані від клієнта
8	Дані для клієнта
9	Дані від платіжної системи
10	Дані для платіжної системи

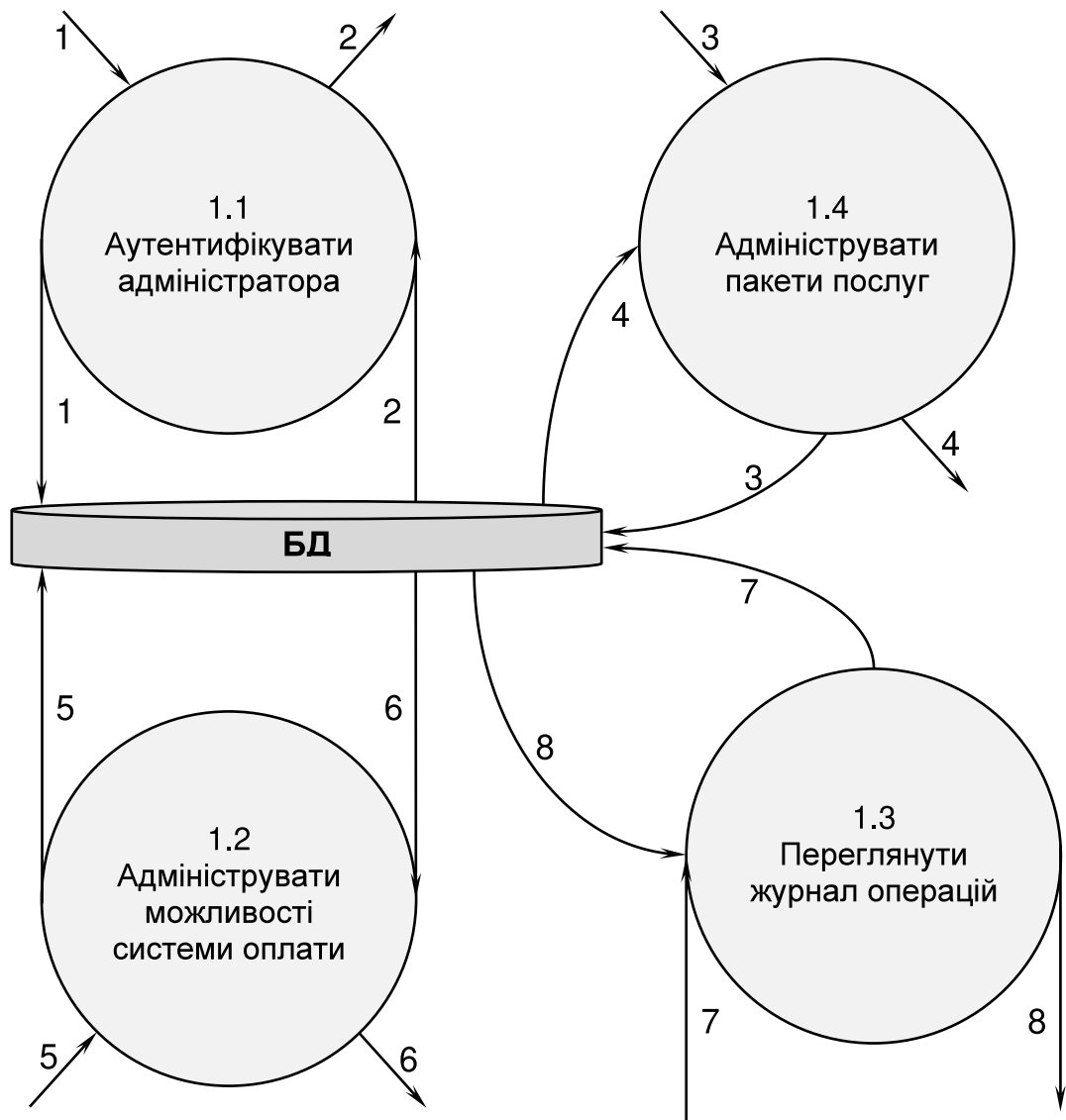


Рисунок 2.4 – Діаграма потоків даних декомпозиції процесу "Адмініструвати параметри системи"

Таблиця 2.3 – Позначення на діаграмі потоків даних декомпозиції процесу "Адмініструвати дані параметри системи"

Умовне позначення	Потік даних
1	Параметри для входу адміністратора
2	Результат авторизації
3	Параметри пакетів послуг
4	Результат зміни параметрів
5	Параметри систем оплати
6	Результат зміни параметрів системи оплати
7	Запит на перегляд журналу операцій
8	Дані з журналу операцій

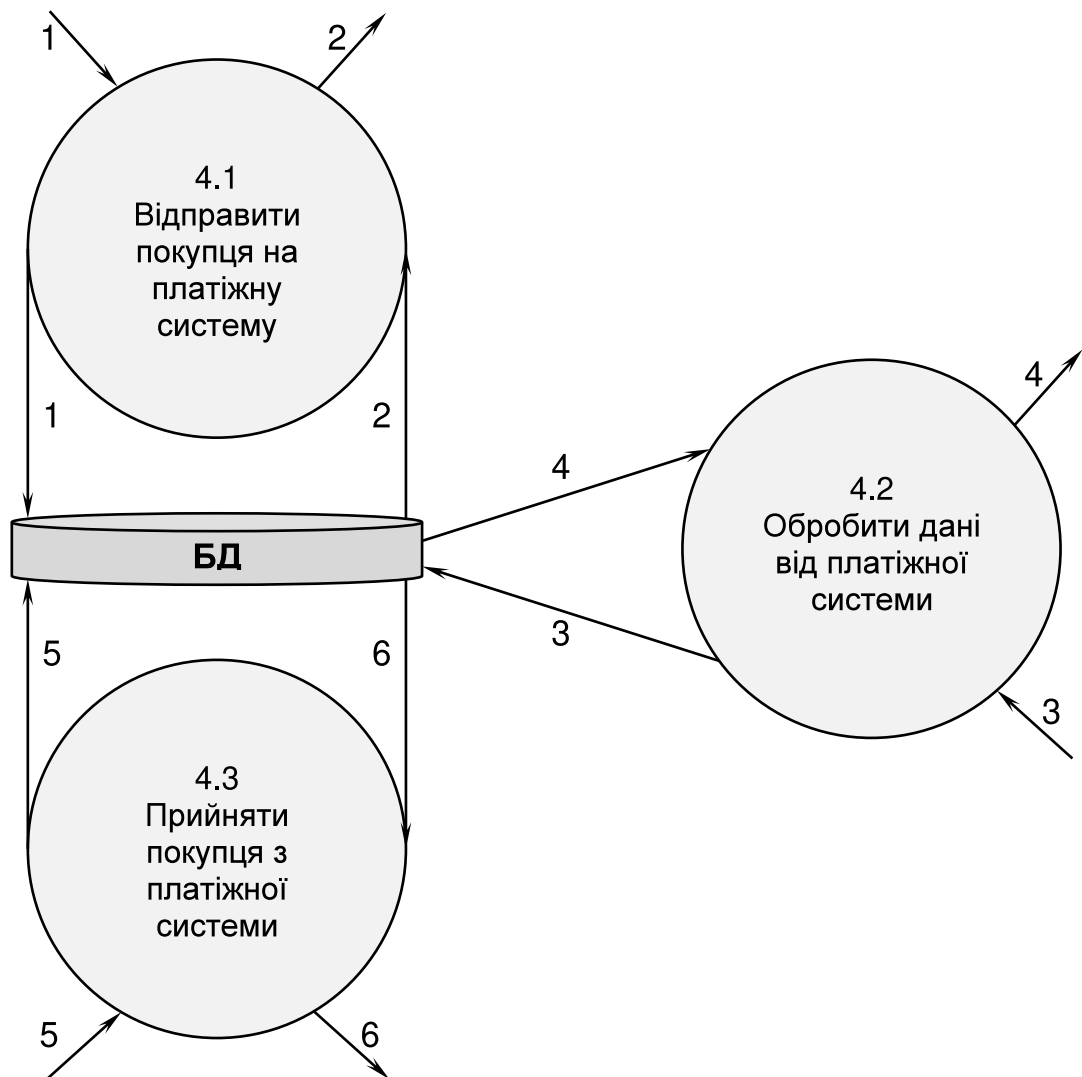


Рисунок 2.5 – Діаграма потоків даних декомпозиції процесу "Взаємодіяти з платіжною системою"

Таблиця 2.4 – Позначення на діаграмі потоків даних декомпозиції процесу "Взаємодіяти з платіжною системою"

Умове позначення	Потік даних
1	Дані про замовлення
2	Дані про перехід до платіжної системи
3	Дані про результат операції
4	Відповідь на результат
5	Дані повернення покупця з платіжної системи
6	Результат обробки замовлення

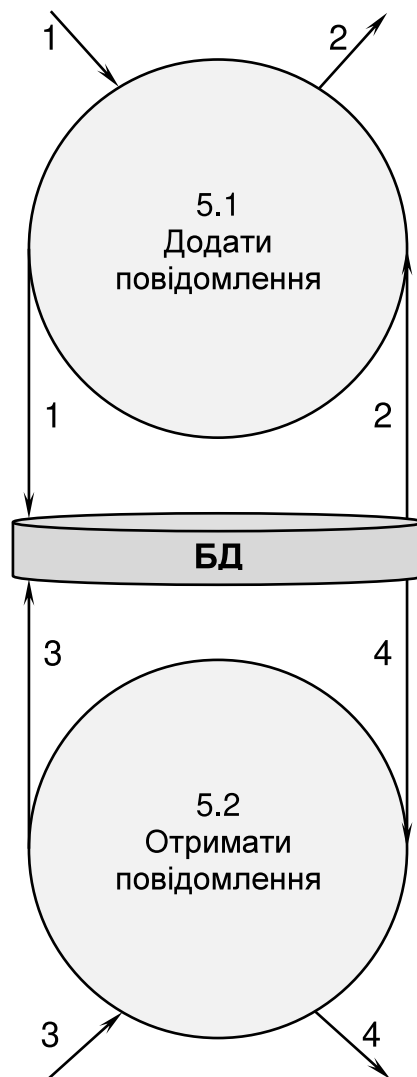


Рисунок 2.6 – Діаграма потоків даних декомпозиції процесу "Використовувати систему повідомлень"

Таблиця 2.5 – Позначення на діаграмі потоків даних декомпозиції процесу "Використовувати систему повідомлень"

Умовне позначення	Потік даних
1	Дані про повідомлення
2	Дані про результат відправлення повідомлення
3	Запит на отримання повідомлень
4	Текст повідомлень

2.1.3 Опис функцій, що автоматизуються.

2.1.3.1 Вихідні дані.

Головною особливістю системи, що впливає на її функціональні характеристики, є поєднання системи прийняття замовлень з системою виконання та реалізації замовлення.

При оформленні замовлення, система взаємодіє з платіжними системами з якими йде обмін даними про замовлення, про власника та про клієнта.

2.1.3.2 Цілі автоматизованої системи і автоматизовані функції.

Ціль системи – забезпечити виконання наступних функцій:

- ✓ створення та редагування пакетів послуг, що надає компанія;
- ✓ реєстрація та облік клієнтів;
- ✓ формування замовлення послуг клієнтом;
- ✓ взаємодія з системою управління сервером для автоматизованого виконання замовлення;
- ✓ взаємодія з платіжними системами;
- ✓ передача повідомлень всередині системи.

2.1.3.3 Характеристика функціональної структури.

Інформаційна система повинна володіти рядом функціональних характеристик, які можна згрупувати в такий спосіб:

1. Користування клієнтами:

- ✓ реєстрація клієнтів;
- ✓ створення замовлень;
- ✓ аутентифікація клієнтів;
- ✓ підтримка різних систем оплати;
- ✓ вибір клієнтом систем оплати та встановлення параметрів кожної системи оплати;
- ✓ перехід клієнта на сайт платіжної системи для оплати свого замовлення;
- ✓ перегляд клієнтом переліку замовлень та інформування про оплату замовлень;
- ✓ зміна клієнтом стану замовлення;
- ✓ конфіденційність передачі даних по комп'ютерній мережі.

2. Надання системи передачі повідомлень:

- ✓ можливість створювати повідомлення для користувачів, що зареєстровані у системі (у відповідності до бізнес-логіки системи);
- ✓ перегляд адміністратором та покупцями повідомлень, що були написані для них та можливість відповідати на ці повідомлення;

3. Взаємодія з системою управління хостингом:

- ✓ автоматичне створення аккаунта клієнта у відповідності до замовленого пакету;

- ✓ редагування аканту клієнта.

4. Адміністрування:

- ✓ надання можливості адміністратору створювати нові пакети та редагувати або видаляти існуючі;
- ✓ прийняття замовлень які не можуть бути автоматизовані (послуги, що діють за партнерськими програмами, реєстрація доменного імені, встановлення SSL-сертифікату, тощо);
- ✓ надання можливості адміністратору переглядати замовлення клієнтів та здійснювати контроль оплати;
- ✓ адміністрування клієнтських аккаунтів.

2.1.4 Опис постановки задачі.

2.1.4.1 Характеристика комплексу задач.

Призначенням системи є автоматизація функцій, що виконують адміністратори компанії при надходженні замовлення (вибір пакету послуг клієнтом, розміщення замовлення та контроль за його оплатою, виконання замовлення).

Метою створення системи є підвищення ефективності та якості роботи компанії по наданню послуг.

Це вирішує більшу частину задач, що виконує відділ продаж у будь-якій компанії та дозволяє здійснювати продаж пакетів послуг в режимі реального часу без обмежень по часу доби та місця положення клієнта.

Система пов'язана з платіжними системами, через які здійснюється оплата послуг та з системою управління сервером, що забезпечує автоматизоване виконання замовлень.

2.1.4.2 Вихідна інформація.

У результаті роботи системи будуть формуватися поточні звіти замовлень, статистика відвідувань клієнтами веб-сайту, загальна кількість користувачів, що використовують систему.

Видача звітів буде здійснюватись за запитом, при встановленні користувачем року та місяця або діапазону дат, за які треба підготувати звіт.

Видача звітів відбуватиметься на екран монітора у форматі HTML, що

дасть змогу друкувати цей документ.

Час видачі звітів не повинен перевищувати 60 секунд. Звіти стосовно виконання замовлень повинні видаватися лише адміністратору системи. Дані у звітах будуть представлені кількістю клієнтів та сумою виручки, яка буде вимірятися в умовних одиницях з двома знаками після коми.

2.1.4.3 Первісна інформація.

Первісною інформацією для системи будуть

1. Дані пакети послуг, що надає компанія: назва, ціна за місяць, параметри. Ці дані адміністратор вводить у систему за допомогою електронної форми, яка надається після авторизації у системі.

2. Дані про клієнтів, що містять: особисті дані клієнта, його контакти та дані для авторизації у системі. Ці дані клієнт вводить під час заповнення електронної форми реєстрації у системі, або при оформленні першого замовлення товарів.

2.1.5 Проектна оцінка надійності системи.

Для оцінки надійності підсистеми використовується принцип визначення показників надійності підсистеми за характеристиками надійності комплектуючих елементів, що дозволяє вести розрахунок в процесі проектування підсистеми виходячи з надійності елементів та вузлів.

Виходячи з важливості серверної станції для функціонування підсистеми загалом, доцільно визначити надійність саме серверної станції. Можна виділити наступні компоненти серверних станцій, вихід з ладу яких є найбільш ймовірним:

- ✓ блок живлення веб-сервера;
- ✓ блок безперебійного живлення;
- ✓ мережний комутатор.

Потрібно розрахувати ймовірність безвідмовної роботи кожного з приведених вище компонентів, а також надійність підсистеми в цілому.

У розрахунку використовуються дані, наведені в технічній документації кожного з компонентів групи елементів.

Середній час роботи даних пристроїв візьмемо з їх технічної документації.

Середній час роботи блоку живлення: $T_{\text{бж.ср.}} = 10000$ годин.

Середній час роботи джерела безперебійного живлення: $T_{\text{дбж.ср.}} = 12000$ годин.

Середній час роботи комутатора: $T_{\text{к.ср.}} = 15000$ годин.

Розрахунок проводимо для визначення ймовірності безвідмовної роботи кожного з перерахованих елементів протягом 1000 годин за формулою:

$$P(t) = \exp\left(-\int_0^t \lambda(t) dt\right) = \exp(-\lambda_0 t),$$

де t – час, протягом якого розраховується надійність;

λ_0 – інтенсивність відмов елемента інформаційної системи протягом середнього часу безвідмовної роботи системи.

Інтенсивність відмов елемента підсистеми за середній час безвідмовної роботи $T_{\text{ср.}}$ розраховується за формулою:

$$\lambda_0 = \frac{1}{T_{\text{ср.}}}.$$

Таким чином обчислюємо частоту відмов:

для блоку живлення ПК:

$$\lambda_0 = 1/10000 = 0,0001; P_{\text{бж}}(t) = \exp(-0,0001 \cdot 1000) = 0,920$$

для джерела безперебійного живлення:

$$\lambda_0 = 1/12000 = 0,000083; P_{\text{дбж}}(t) = \exp(-0,000083 \cdot 1000) = 0,943$$

для комутатора:

$$\lambda_0 = 1/15000 = 0,000066; P_{\text{к}}(t) = \exp(-0,000066 \cdot 1000) = 0,961.$$

Під час визначення надійності підсистеми в цілому через відомі показники надійності її елементів вводяться два припущення: відмови елементів статично незалежні, відмова будь-якого елемента приводить до відмови системи. Прийняті припущення дозволяють застосувати для підрахунку ймовірності безвідмовної роботи інформаційної підсистеми теорему добутку ймовірностей, що представлена формулою:

$$P_{IC}(t) = \prod_{i=1}^n P_i(t),$$

де $P_i(t)$ – ймовірність безвідмовної роботи i -го елемента;

n – кількість елементів системи.

Надійність технічного забезпечення інформаційної системи протягом 1000 годин безперервної роботи визначається добутком ймовірностей безвідмовної роботи блоку живлення, джерела безперебійного живлення та комутатора.

$$P_{IC}(t) = P_{дж}(t) \cdot P_{дбж}(t) \cdot P_{к}(t) = 0,920 \cdot 0,943 \cdot 0,961 = 0,833.$$

Проведений розрахунок показав, що ймовірність безвідмовної роботи системи протягом 1000 годин складає близько 83%.

Більш якісні та відповідно дорогі технічні елементи, що впливають на безвідмовність роботи системи будуть збільшувати вірогідність безвідмовної роботи системи в цілому.

2.1.6 Загальний опис системи.

Призначення системи.

Призначенням системи є автоматизація функцій, що виконуються адміністратором сайту при надходженні замовлення.

Метою створення системи є підвищення ефективності та якості роботи компанії по наданню послуг.

Задачі системи полягають в забезпеченні виконання наступних функцій:

- ✓ створення та редагування пакетів послуг, що надає компанія;
- ✓ реєстрація та облік клієнтів;
- ✓ формування замовлення послуг клієнтом;
- ✓ взаємодія з системою управління сервером для автоматизованого виконання замовлення;
- ✓ взаємодія з платіжними системами;
- ✓ передача повідомлень всередині системи.

Опис системи.

Інформаційна система призначена для створення повнофункціональної електронної компанії з автоматизованою системою виконання замовлень та можливістю вибору платіжних систем.

Опис взаємозв'язків інформаційної системи з іншими системами.

При замовленні система взаємодіє з платіжними системами, з якими йде обмін даними про користувача та інформацією про здійснення оплати покупцем.

Взаємодія з платіжними системами залежить від алгоритму роботи системи, але відбуватиметься з використанням протоколів HTTP і HTTPS. Структура даних, що буде описувати взаємодію інформаційної системи з платіжними системами буде давати змогу розширювати систему додаючи до неї взаємодію з іншими платіжними системами незалежно від методів їх взаємодії з інформаційною системою.

Після здійснення платіжних операцій система здійснює взаємодію з системою управління сервером засобами API, що має вказана система.

Опис підсистем.

Система складається з наступних підсистем:

- ✓ підсистема електронної крамниці, яка забезпечує введення та відображення пакетів послуг, що надає компанія;
- ✓ підсистема передачі повідомлень, яка забезпечує внутрішню передачу повідомлень між користувачами системи;
- ✓ підсистема платежів, яка забезпечує здійснення операцій продажу товарів покупцеві через одну з підключених платіжних систем;
- ✓ підсистема управління аккаунтами, яка забезпечує взаємодію з системою управління сервером для автоматизованого створення та редагування клієнтських аккаунтів на сервері.

2.2 Рішення з організаційного забезпечення.

2.2.1 Опис організаційної структури.

Інформаційна система розробляється для розширення систем оплати та автоматизації виконання замовлень. Організаційна структура компанії, що буде супроводжувати систему, зображена на рисунку 2.1.

Користувачами системи будуть адміністратор та клієнти. При введенні системи в дію організаційна структура компанії не зміниться.

Директор здійснюють загальне керівництво компанією на підставі графіків планування та звітів, що представляються системою.

Адміністратор здійснює адміністрування пакетів послуг що надає компанія, займається модерацією клієнтів, здійснює технічну підтримку та в разі надходження замовлень, що не можуть бути автоматизовані, – виконує їх.

2.2.2 Інструкція користувача.

Для нормального функціонування системи, всі операції, що проводяться у ній користувачами, повинні відповідати Інструкції користувача, приведеної в Додатку Г.

2.2.3 Заходи щодо організаційного забезпечення.

Ефективна робота системи управління діяльністю компанії вимагає організаційних заходів, цілями яких є підтримка даних про процеси, що відбуваються, в актуальному стані та грамотній своєчасній реакції користувачів на інформацію, що їй надається системою.

2.2.4 Організаційно-кадрові заходи.

Для підтримки апаратних і програмних засобів у робочому стані, а також рішення поточних задач настроювання, установки і відновлення загальносистемного програмного та апаратного забезпечення, на підприємстві повинна бути введена штатна одиниця системного адміністратора, або для цих цілей застосовуватися аутсорсінг (outsourcing – надання ІТ-функцій сторонніми організаціями).

2.3 Рішення з інформаційного забезпечення.

2.3.1 Опис інформаційного забезпечення системи.

Склад інформаційного забезпечення

Інформаційне забезпечення системи складається з бази даних про пакети послуг, даних інформації про клієнтів та їхні замовлення.

Організація інформаційного забезпечення

Для забезпечення здатності масштабувати систему та підтримання високої швидкості доступу до даних, данні системи будуть зберігатися у базі даних. База даних повинна знаходитись в реляційній СУБД, яка повинна

підтримувати стандарт ANSI SQL. Це забезпечить сумісність системи з розповсюдженими СУБД. Рекомендована СУБД – MySQL.

База даних розташовується на жорсткому диску сервера баз даних. Доступ до бази даних повинен бути обмеженим адміністратором системи і можливим лише з веб-сервера, для запобігання несанкціонованого втручання.

Організація збору і передачі інформації

Введення інформації в базу даних системи повинно здійснюватися за допомогою програмного забезпечення системи.

При встановленні системи адміністратор вводить данні для аутентифікації його як адміністратора системи, всі інші данні в систему вводять користувачі у процесі роботи .

Всі користувачі вводять інформацію у систему віддалено, використовуючи робочі станції і Web-оглядач.

Побудова системи класифікації і кодування

Для кодування об'єктів у системі будуть використані цілі натуральні числа. Для кодування кожного об'єкту будуть використані власні нумерації.

Організація внутрішньо-машинної інформаційної бази.

З аналізу функціональної моделі системи можна побудувати концептуальну модель даних системи.

Найбільш розповсюдженим засобом представлення концептуальної моделі даних є діаграми "сутність-зв'язок". З їхньою допомогою визначаються важливі для предметної області об'єкти (сутності), їхні властивості (атрибути) і відносини один з одним (зв'язки). ERD безпосередньо використовуються для проектування реляційних баз даних.

Для побудови концептуальної моделі даних системи необхідно виділити сутності і визначити їх атрибути.

У системі що розробляється можна виділити наступні сутності:

- ✓ пакет послуг;
- ✓ аккаунт клієнта;

- ✓ замовлення;
- ✓ система оплати;
- ✓ оплата;
- ✓ повідомлення;
- ✓ рахунок клієнта.

Наведені сутності складаються з атрибутів.

Сутність "Пакет послуг" складається з:

- ✓ код пакета;
- ✓ назва;
- ✓ короткий опис;
- ✓ повний опис;
- ✓ місячна вартість/

Сутність "Аккаунт клієнта" складається з:

- ✓ код клієнта;
- ✓ ім'я;
- ✓ прізвище;
- ✓ електронна адреса;
- ✓ країна;
- ✓ місто;
- ✓ адреса;
- ✓ телефон;
- ✓ коментарі;
- ✓ дата реєстрації.

Сутність "Замовлення" складається з:

- ✓ код замовлення;
- ✓ код клієнта;
- ✓ код пакету послуг;
- ✓ код рахунку;
- ✓ дата замовлення;
- ✓ дата підтвердження;
- ✓ дата завершення замовлення;
- ✓ коментарі;
- ✓ статус.

Сутність "Системи оплати" складається з:

- ✓ код системи оплати;
- ✓ назва системи оплати;
- ✓ опис системи оплати;
- ✓ ім'я файлу системи оплати;
- ✓ параметри системи оплати;
- ✓ статус системи оплати.

Сутність "Оплата" складається з:

- ✓ код оплати;
- ✓ код клієнта;
- ✓ код замовлення;
- ✓ код системи оплати;
- ✓ дата оплати;
- ✓ сума оплати.

Сутність "Повідомлення" складається з:

- ✓ код повідомлення;
- ✓ код клієнта;
- ✓ ім'я клієнта;
- ✓ дата повідомлення;
- ✓ код попереднього повідомлення;
- ✓ тема повідомлення;
- ✓ текст повідомлення;
- ✓ статус прочитання;
- ✓ статус завершення листування.

Сутність "Рахунок клієнта" складається з:

- ✓ код рахунку;
- ✓ код аккаунту клієнта;
- ✓ код пакету послуг;
- ✓ дата початку замовлення;
- ✓ дата завершення замовлення;
- ✓ додаткова інформація.

Зв'язки між цими сутностями зображено на схемі "сутність-зв'язок", яка наведена на рисунку 2.7.

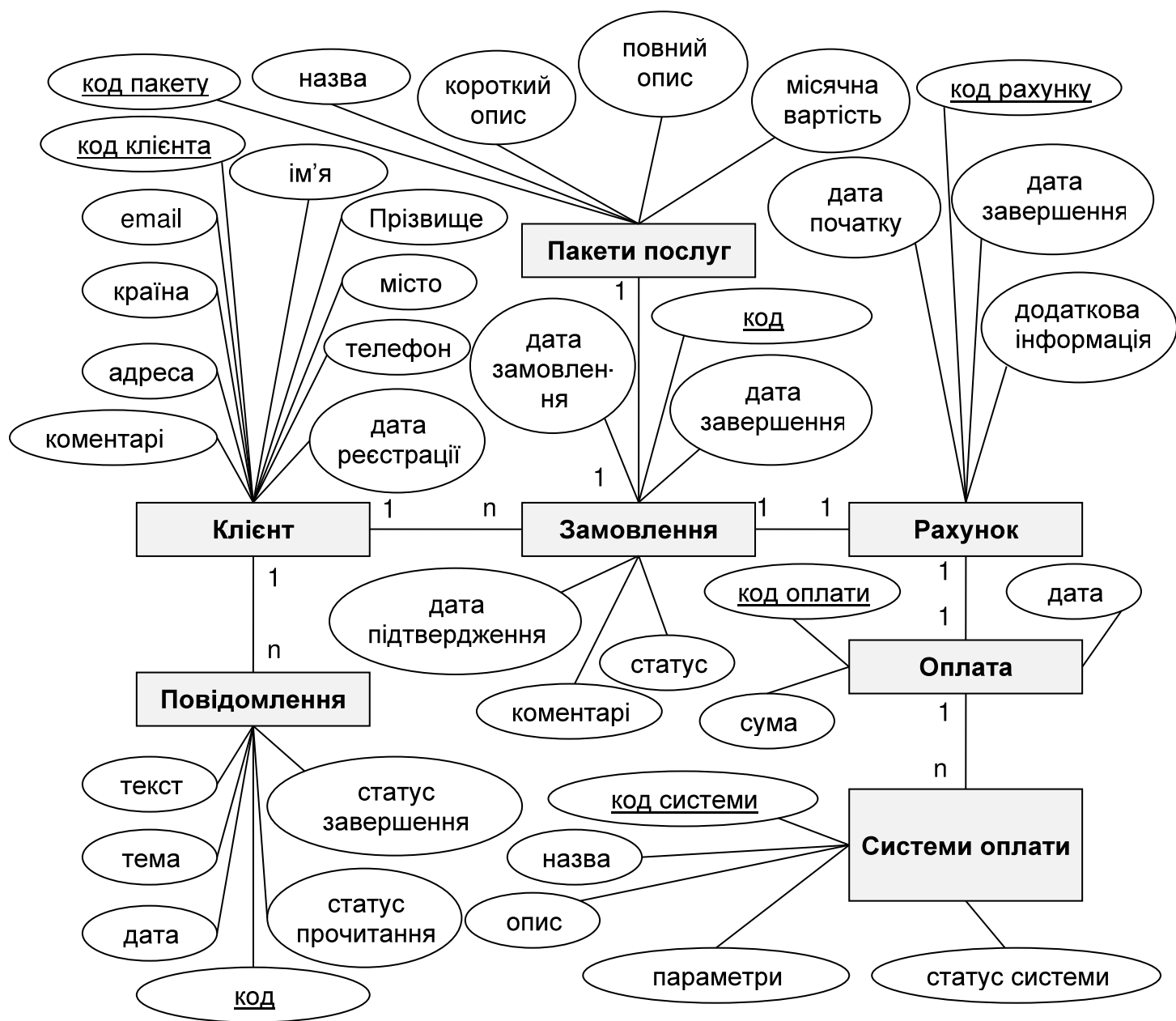


Рисунок 2.7 – Діаграма "Сутність-зв'язок" системи

2.3.2 Перелік вхідних сигналів і даних.

Перелік вхідних сигналів.

Склад та данні вхідних сигналів залежать від системи оплати, яка застосовується для оплати замовлення. Ці сигнали система оплати передає інформаційній системі як данні про параметри платежу та його результат. Стандартними для всіх систем є номер, сума платежу, результат платежу.

Після оплати система взаємодіє з системою управління сервером від якої отримує дані про виконання операцій на сервері (створення або редагування аккаунту). Отриманими даними є результат виконання операцій.

Базуючись на цих даних система змінює стан замовлення на відповідний

до результату оплати та виконання операцій на сервері.

Сигнали передаються до системи через протокол HTTP методами GET або POST.

Перелік вхідних даних

Вхідним для системи є особисті данні клієнтів та каталог пакетів послуг.

2.3.3 Перелік вихідних сигналів і даних.

Перелік вихідних сигналів

Вихідні сигнали системи представляють собою параметри для систем оплати та дані для системи управління сервером.

До систем оплати зазвичай передаються номер замовлення, сума до сплати, номер клієнта у системі оплати та специфічні для кожної системи оплати параметри.

До системи управління сервером передаються номер та дані клієнта, обраний пакет послуг.

Перелік вихідних документів

Вихідні документи будуть таблиці зі статистикою замовлень, що виведені на екрані монітора, з можливістю роздрукування цих даних.

2.3.4 Відомість машинних носіїв інформації.

На машинних носіях системи містяться:

- ✓ текст програми системи;
- ✓ набір SQL команд для створення структури бази даних.

2.3.5 Опис організації інформаційної бази.

Логічна структура

Виходячи з аналізу концептуальної моделі даних системи, розробимо логічну структуру бази даних. Логічна структура бази даних системи представлена на рисунку 2.8.

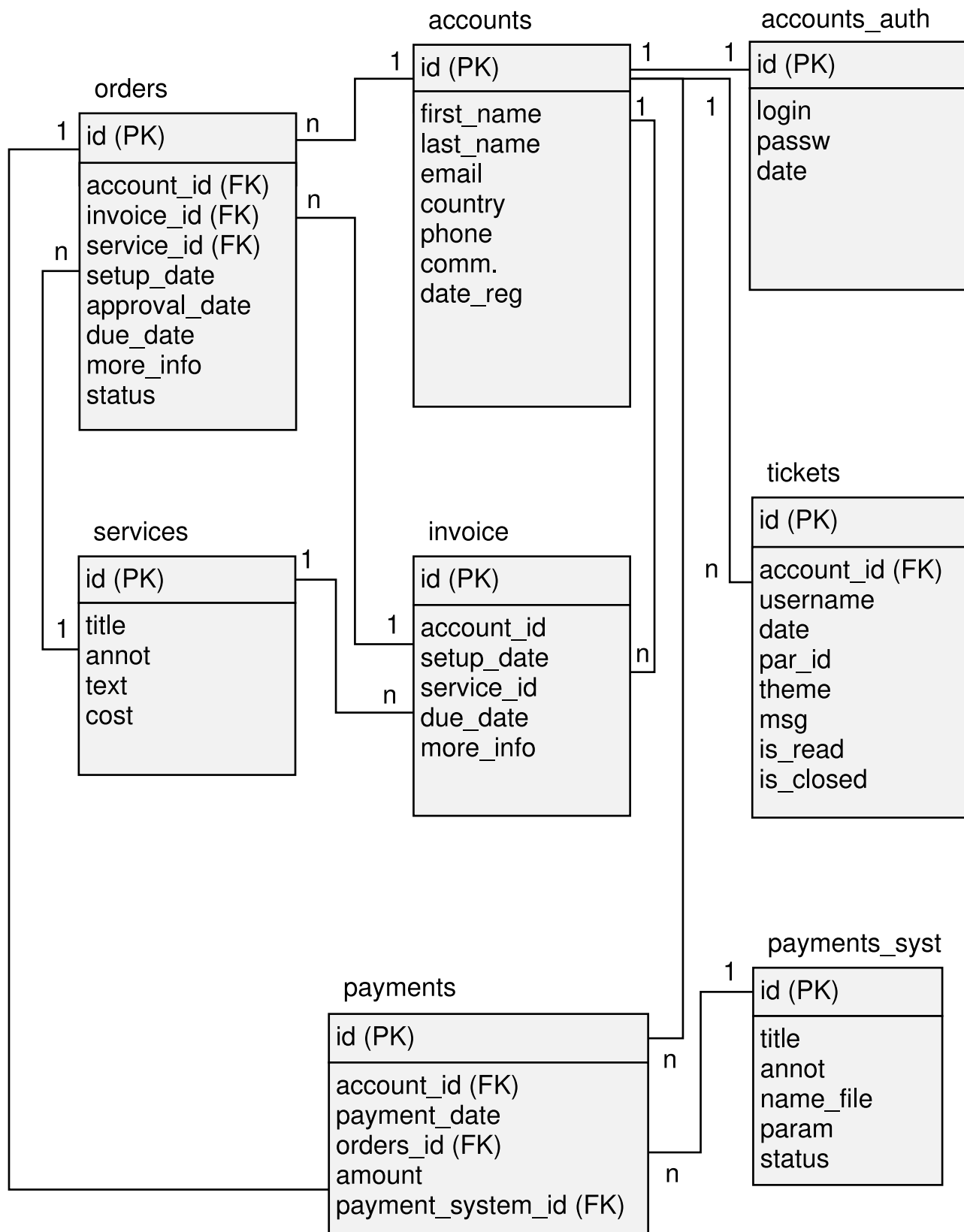


Рисунок 2.8 – Логічна структура бази даних системи

Фізична структура.

Розробимо фізичну структуру кожної таблиці, що входить у базу даних. Типи даних будемо вказувати для СУБД MySQL.

При описі фізичних таблиць бази даних використані стандартні

позначення ключів:

- ✓ РК (первинний ключ);
- ✓ АК (альтернативний ключ);
- ✓ FK (зовнішній ключ).

Усі таблиці бази даних знаходяться в 1 нормальній формі (НФ), оскільки вони є двовимірними, не містять полів, що включають кілька значень, і порожніх ключових полів (РК).

Усі таблиці бази даних знаходяться в 2 НФ, тому що всі їхні поля, що не входять у первинний ключ, зв'язані повною функціональною залежністю з первинним ключем.

Структура таблиці "Пакети послуг" наведена у таблиці 2.6.

Таблиця 2.6 – Структура таблиці "Пакети послуг" ("Services")

Назва поля (українською)	Назва поля (англійською)	Тип даних	Ключ
Код пакета	id	Int	РК
Назва пакету	title	Varchar(100)	АК
Короткий опис	annot	Varchar(255)	
Вартість пакету	cost	Float	
Повний опис	text	Text	

Структура таблиці "Аккаунти" наведена у таблиці 2.7.

Таблиця 2.7 – Структура таблиці "Аккаунти" ("Accounts")

Назва поля (українською)	Назва поля (англійською)	Тип даних	Ключ
Код аккаунту	id	Int	РК
Ім'я	first_name	Varchar(100)	
Прізвище	last_name	Varchar(100)	
Email	email	Varchar(50)	
Країна	country	Varchar(100)	
Місто	city	Varchar(100)	
Телефон	phone	Varchar(100)	
Короткий опис	comm	Varchar(255)	
Дата реєстрації	date_reg	timestamp	

Структура таблиці "Замовлення" наведена у таблиці 2.8.

Таблиця 2.8 – Структура таблиці "Замовлення" ("orders")

Назва поля (українською)	Назва поля (англійською)	Тип даних	Ключ
Код замовлення	id	Int	PK
Код аккаунту	account_id	Int	FK
Дата замовлення	setup_date	timestamp	
Дата підтвердження	approval_date	timestamp	
Код пакету послуг	service_id	Int	FK
Код рахунку	invoice_id	Int	FK
Дата завершення замовлення	due_date	timestamp	
Додаткова інформація	more_info	Varchar(255)	
Статус	status	int	

Структура таблиці "Авторизація" наведена у таблиці 2.9.

Таблиця 2.9 – Структура таблиці "Авторизація" ("accounts_auth")

Назва поля (українською)	Назва поля (англійською)	Тип даних	Ключ
Код	id	Int	PK
Логін	login	Varchar(30)	
Пароль	passw	Varchar(40)	
Дата останньої авторизації	date	timestamp	

Структура таблиці "Рахунок" наведена у таблиці 2.10.

Таблиця 2.10 – Структура таблиці "Рахунок" ("invoice")

Назва поля (українською)	Назва поля (англійською)	Тип даних	Ключ
Код рахунку	id	Int	PK
Код аккаунту	account_id	Int	FK
Код пакету послуг	service_id	Int	FK
Дата початку замовлення	setup_date	timestamp	
Дата завершення замовлення	due_date	timestamp	
Додаткова інформація	more_info	Varchar(255)	

Структура таблиці "Системи оплати" наведена у таблиці 2.11.

Таблиця 2.11 – Структура таблиці "Системи оплати" ("payment_system")

Назва поля (українською)	Назва поля (англійською)	Тип даних	Ключ
Код системи	id	Int	PK
Назва системи	title	Varchar(100)	
Короткий опис	annot	Varchar(255)	
Назва файлу	name_file	Varchar(20)	
Параметри системи	param	Varchar(255)	
Статус	status	Int	

Структура таблиці "Оплата" наведена у таблиці 2.12.

Таблиця 2.12 – Структура таблиці "Оплата" ("payments")

Назва поля (українською)	Назва поля (англійською)	Тип даних	Ключ
Код оплати	id	Int	PK
account_id	account_id	Int	FK
Дата оплати	payment_date	timestamp	
Сума	amount	Float	
Код замовлення	orders_id	Int	FK
Код системи оплати	paymet_system_id	Int	FK

Структура таблиці "Оплата" наведена у таблиці 2.12.

Таблиця 2.13 – Структура таблиці "Повідомлення" ("tickets")

Назва поля (українською)	Назва поля (англійською)	Тип даних	Ключ
Код повідомлення	id	Int	PK
Код клієнта	account_id	Int	FK
Ім'я клієнта	username	Varchar(100)	
Дата повідомлення	date	Timestamp	
Код попереднього повідомлення	par_id	Int	FK
Тема повідомлення	theme	Varchar(100)	
Текст повідомлення	msg	text	
Статус прочитання	is_read	int	
Статус завершення листування	is_closed	int	

Організація ведення інформаційної бази.

Для створення бази буде необхідним виконання SQL коду для створення таблиць бази даних. Текст коду створення таблиць наведено у додатку.

Для захисту бази від руйнування рекомендується створювати копії бази раз на день, та копіювати їх на інший носій.

2.3.6 Каталог бази даних.

У базі даних зберігається інформація про наступні об'єкти:

- ✓ пакети послуг;
- ✓ заробітки компанії;
- ✓ права доступу адміністратора;
- ✓ зони доступу клієнтів;

- ✓ клієнт;
- ✓ замовлення;
- ✓ склад замовлення;
- ✓ деталі оплати замовлення;
- ✓ системи оплати;
- ✓ параметри системи оплати;
- ✓ повідомлення;
- ✓ система управління клієнтськими аккаунтами.

2.3.7 Склад вихідних даних.

Вихідними даними системи є параметри платежу, що передаються системі оплати для переведення грошей за продукт, а також дані які передаються системі управління сервером(WHM).

Вихідні дані до систем оплати.

Параметри залежать від кожної окремої системи оплати.

Розглянемо деякі з них на прикладі: WebMoney, Transfer та PayPal.

WebMoney Transfer.

Система WebMoney Transfer приймає данні про оплати на один URL, але має великий набір даних, що можуть передаватися.

Форма передачі запиту з веб-сайту власника товарів в сервіс Web Merchant Interface через веб-браузер покупця повинна мати наступні атрибути і полючи:

Action – <https://merchant.webmoney.ru/lmi/payment.asp>.

Method – POST.

Fields – параметри передані у формі, описані нижче:

✓ LMI_PAYEE_PURSE – гаманець власника товарів. Обов'язковий параметр. Гаманець власника товарів, на який покупець повинний зробити платіж. Формат – буква і 12 цифр. На даний час допускається використання гаманців Z-, R-, E- і D-типів.

✓ LMI_PAYMENT_AMOUNT. Сума платежу. Обов'язковий параметр. Сума платежу, що власник товарів бажає одержати від покупця. Сума повинна бути більше нуля, дробова частина відокремлюється крапкою.

✓ LMI_PAYMENT_NO. Внутрішній номер покупки власника товарів. У

цьому полі власник товарів задає номер покупки у відповідності зі своєю системою обліку. Незважаючи на те, що параметр не є обов'язковим, рекомендовано завжди задавати його. Бажано використовувати унікальний номер для кожного платежу, що дозволить швидко одержати стосовну до нього інформацію через інші інтерфейси системи WebMoney Transfer. Номер повинний представляти собою ціле число без знака не більше 2147483647.

✓ LMI_PAYMENT_DESC. Призначення платежу. Опис товару або послуги. Формується власником товарів. Якщо він присутній, додається в призначення платежу в операцію переказу WM. Максимальна довжина – 255 символів.

✓ LMI_SIM_MODE. Режим тестування. Додаткове поле, що визначає режим тестування. Діє тільки в режимі тестування і може приймати одне з наступних значень:

- 0 чи не встановлене: для всіх тестових платежів сервіс буде імітувати успішне виконання;
- 1: для всіх тестових платежів сервіс буде імітувати виконання з помилкою (платіж не виконаний);
- 2: близько 80% запитів на платіж будуть виконані успішно, а 20% – не виконані.

✓ LMI_RESULT_URL. Заміна Result URL. Це поле дозволяє власнику товарів тимчасово змінити параметр "Result URL", встановлений власником товарів на сторінці настройки сайту Web Merchant Interface. Якщо в налаштуваннях установлений прапор "Дозволяти використовувати URL, передані у формі", то передане значення замінює значення параметра "Result URL", встановлене в настройках на сайті Web Merchant Interface. У протилежному випадку завжди використовується значення, встановлене в настройках на сайті Web Merchant Interface. Формат цього поля повинний строго відповідати значенню параметра "Result URL".

✓ LMI_SUCCESS_URL. Заміна Success URL. Це поле дозволяє власнику товарів тимчасово змінити параметр "Success URL", встановлений їм на сторінці настройки сайту Web Merchant Interface. Якщо в настройках установлений прапор "Дозволяти використовувати URL, передані у формі", то передане значення замінює значення параметра "Success URL", встановлене в настройках на сайті Web Merchant Interface. У протилежному випадку завжди

використовується значення, встановлене в настройках на сайті Web Merchant Interface. Формат цього поля повинний строго відповідати значенню параметра "Success URL".

✓ LMI_SUCCESS_METHOD. Заміна методу виклику Success URL. Це поле дозволяє власнику товарів тимчасово змінити параметр "Метод виклику Success URL", встановлений їм на сторінці настройки сайті Web Merchant Interface. Якщо в настройках встановлено прапор "Дозволяти використовувати URL передані у формі", то передане у формі значення замінює значення параметра "Метод виклику Success URL", встановлене у настройках на сайті Web Merchant Interface. У противному випадку завжди використовується значення, встановлене в настройках на сайті Web Merchant Interface. Це поле повинне приймати значення 0, 1 чи 2, що відповідає значенням параметра "Метод виклику Success URL" – "GET", "POST" чи "LINK".

✓ LMI_FAIL_URL. Заміна Fail URL. Аналогічно до параметра LMI_SUCCESS_URL – заміна Success URL. Аналогічно до LMI_SUCCESS_METHOD – заміна методу виклику Success URL.

✓ LMI_FAIL_METHOD. Заміна методу виклику Fail URL.

Додаткові параметри власника товарів визначається власником товарів. Усі параметри форми, що не мають у назві префікса "LMI_", обробляються сервісом Web Merchant Interface автоматично і передаються на веб-сайт власника товарів після виконання платежу.

PayPal.

Система PayPal приймає наступні параметри у якості вхідних параметрів:

✓ business. Електронна адреса власника товарів або його ідентифікатор. У випадку оплати на головний рахунок – теж саме що й receiver_email.

✓ receiver_email. Головний e-mail адрес власника товарів. У випадку оплати на неголовний адрес власника товарів, receiver_email все одно буде головним і на ньому будуть зберігатися гроші.

✓ item_name. Назва продуктів. Передається для відображену формі оплати переліку товарів, за які йде оплата. Можна використовувати декілька продуктів додаючи до назви поля цифри. Наприклад item_name1, item_name2.

✓ item_number. Номер продукту. Вказує номер продукту, за які створюється оплата. Як і item_name може бути декілька полів. Наприклад

item_number1, item_number2.

✓ quantity. Кількість продукту. Вказує кількість продукту, за яку створюється оплата. Як і item_name може бути декілька полів. Наприклад item_number1, item_number2.

✓ invoice. Номер рахунка. Задається як унікальний номер операції. За цим номером покупець матиме змогу переглядати свої замовлення.

✓ custom. Довільне значення, що передається системі і буде повернуто нею, але не буде доступне для покупця.

✓ memo. Коротке повідомлення. Буде відображено покупцю під час покупки.

✓ mc_currency. Валюта. Валюта у якій здійснюється оплата.

✓ mc_gross. Сума платежу. Загальна сума до сплати.

Вихідні дані до системи управління сервером.

Система управління сервером (WHM) приймає наступні параметри:

✓ username. Ім'я клієнта для аккаунту.

✓ domain. Доменне ім'я.

✓ plan. Назва пакету для аккаунту.

✓ featurelist. Найменування функцій в списку, який буде використовуватись при створенні нового пакету.

✓ password. Пароль для доступу клієнта.

✓ quota. Квота дискового простору.

✓ contactemail. Контактна поштова адреса.

✓ Maxftp. Кількість FTP-аккаунтів.

✓ maxsql. Кількість MySQL баз даних.

2.3.8 Інструкція з формування і ведення бази даних.

Правила підготовки даних.

Для введення в базу використовується каталог пакетів послуг, що надає компанія.

Також в базу вводяться особисті данні клієнтів.

Всі ці данні вводяться у відповідні форми, що надає система. Форми надаються у вигляді html форм. Зміни інформації вносяться у процесі введення, або шляхом натискання кнопки редагування у інтерфейсі.

Порядок і засоби заповнення бази даних.

Первинне заповнення бази складається з встановлення значень параметрів системи і відбувається при створенні таблиць даних системи. Редагування та зміна цих даних можлива лише адміністратором системи з відповідного інтерфейсу.

Головне заповнення бази відбувається адміністратором системи при введенні даних про пакети послуг, які компанія пропонує. Введення даних відбувається шляхом заповнення html форм у браузері та відправки їх на сервер. Після передачі даних на сервер, скрипт який відповідає за прийом та обробку даних зберігає данні у базі даних.

Введення особистих даних клієнтів відбувається шляхом заповнення html форм у браузері та відправки їх на сервер. Після передачі даних на сервер, скрипт який відповідає за прийом та обробку даних зберігає данні у базі даних.

Процедури зміни і контролю бази даних.

Зміна даних у системі відбувається шляхом редагування даних у html формах, що представлені у браузері, через відповідні пункти меню. Після зміни даних натискається кнопка відправлення даних на сервер. Після передачі даних на сервер скрипт який відповідає за прийом та обробку даних зберігає данні у базі даних.

Редагування особистих даних клієнтів відбувається шляхом заповнення html форм у браузері та відправки їх на сервер. Після передачі даних на сервер скрипт який відповідає за прийом та обробку даних зберігає данні у базі даних.

Порядок і засоби відновлення бази даних.

Для захисту бази від несанкціонованого доступу треба вірно встановити права користувачів бази. Для даної системи необхідно створити одного користувача бази даних, використовуючи якого система буде здійснювати обмін інформацією між базою даних та веб-сервером системи. Для цього користувачу необхідно надати повні права на базу даних системи, але обмежити його доступ, дозволивши його тільки з веб-серверу. Таким чином, для доступу до бази треба бути ще й зареєстрованим користувачем на веб-сервері.

Окрім цього доцільно обмежити доступ до серверу бази даних ззовні, дозволивши лише консольні з'єднання ssh, що використовують шифрування

даних.

Для захисту бази від руйнування рекомендується створювати резервні копії бази даних. Це створюється шляхом використання сервісів бази даних та операційної системи. Резервні копії рекомендовано створювати раз на день та зберігати їх на окремому носії інформації, а за можливістю, й на іншому технічному засобі.

Для відновлення бази треба буде видалити залишки зруйнованої бази та встановити данні з останнього вірного збереження даних.

2.4 Рішення з технічного забезпечення.

2.4.1 Схема автоматизації.

Схема автоматизації містить:

- ✓ спрощене зображення системи;
- ✓ засоби технічного забезпечення, що беруть участь у процесі, відображеному на схемі, за винятком допоміжних приладів і апаратури (джерела живлення, реле, магнітні пускачі);
- ✓ функціональні зв'язки між засобами технічного забезпечення;
- ✓ зовнішні функціональні зв'язки засобів технічного забезпечення з іншими технічними засобами.

Технічне забезпечення система складається з веб-сервера. Схема автоматизації зображена на рисунку 2.9.

Також для адміністрування системи необхідний консольний доступ до серверів через протокол SSL.

2.4.2 Опис комплексу технічних засобів.

Загальні положення.

Технічне забезпечення системи складається з веб-сервера. Для використання системи необхідні клієнтські комп'ютери з мінімальними потребами, так як для використання необхідно лише браузер, рекомендовано, з підтримкою JavaScript.

Розміщення веб-серверу та серверу баз даних на одному серверному комп'ютері зумовлено обмеженням бюджетом проекту на стадії запуску системи, оскільки на початковому етапі одразу великої кількості клієнтів бути не може,

відповідно немає необхідності їх розділяти. З часом, коли кількість клієнтів буде зростати та досягне певного рівня, то ресурс комп'ютеру буде вичерпаний і з'явиться необхідність розділити сервер БД з веб-сервером.

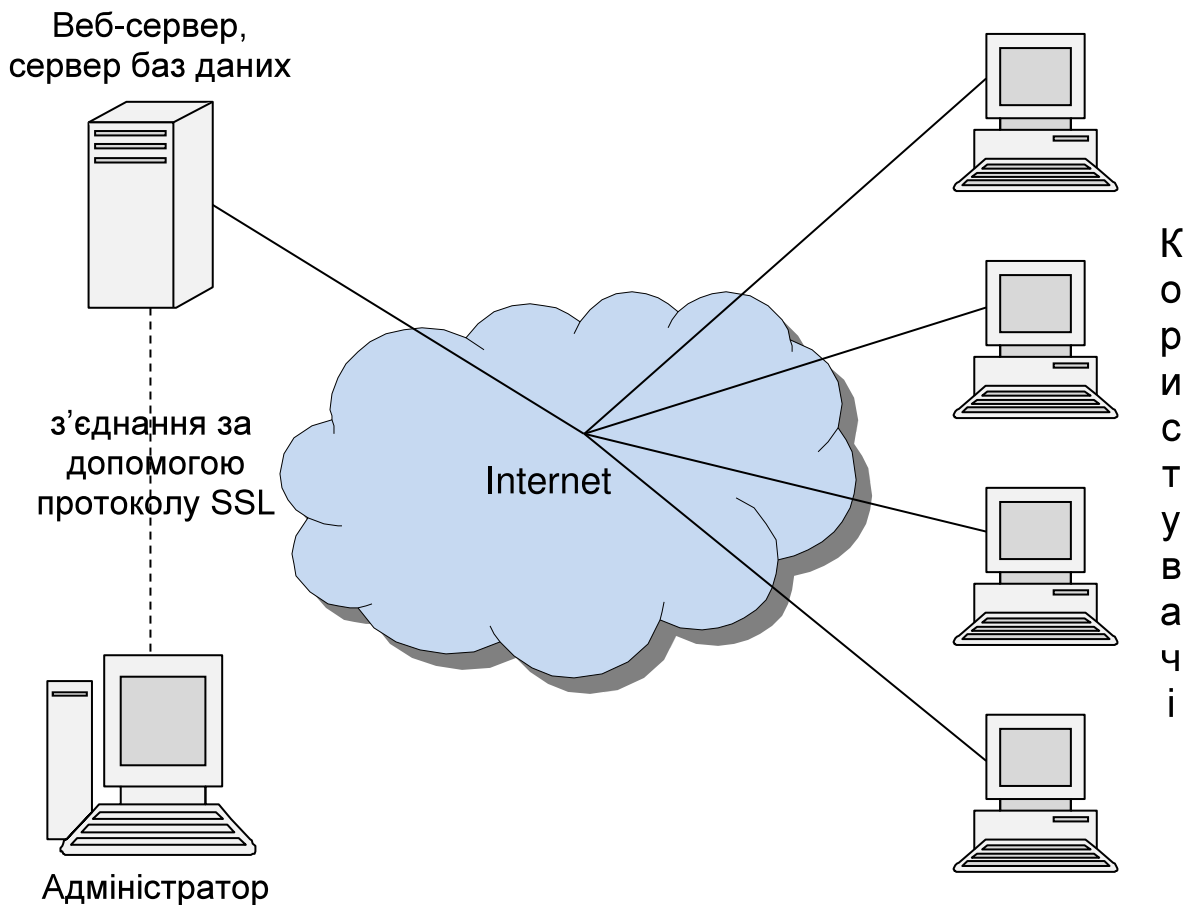


Рисунок 2.9 – Схема автоматизації системи

Структура комплексу технічних засобів.

Технічне устаткування може бути розміщене на технічній площі одного з Internet провайдерів для забезпечення високошвидкісного доступу до серверу.

Для збільшення надійності системи від несанкціонованого доступу рекомендовано для сервера баз даних дозволити доступ лише з веб-сервера.

Засоби обчислювальної техніки.

Технічні засоби сервера повинні задовольняти вимогам операційної системи і СУБД до устаткування.

Технічне забезпечення веб-сервера та клієнтських комп'ютерів наведено у попередньому розділі.

2.5 Рішення з математичного забезпечення. Опис алгоритму.

2.5.1 Призначення і характеристика.

Алгоритм призначено для обробки замовлень клієнтів. За допомогою алгоритму відбувається процес оплати замовлення клієнтом та взаємодія з системою управління сервером. В процесі функціонування, алгоритм впливає на сутності "замовлення" та "оплата" додаючи туди нові данні. В цих сутностях відображено фінансові данні клієнта. Використання цього алгоритму відбувається при переході покупця до оплати замовлення, а потім виклик необхідних функцій API, що надає система управління сервером. Вхідними даними будуть данні про замовлення та клієнта. Формат даних, що передаються, повинен чітко відповідати типам, які має таблиця пакетів послуг, наведена у фізичній моделі.

2.5.2 Використана інформація.

Обробка замовлення

Вхідними даними є :

- ✓ код пакету, `services.id`, натуральне число;
- ✓ назва пакету, `services.title`, рядок тексту до 100 символів;
- ✓ вартість пакету, `services.cost`, дійсне додатне число;
- ✓ короткий опис, `services.annot`, рядок тексту до 255 символів;
- ✓ повний опис, `services.text`, рядок тексту.

2.5.3 Результати розв'язання.

В результаті виконання створюється замовлення та визначається результат його оплати. Данні будуть збережені у базі даних у таблицях `orders`, `invoice` та `payments`.

2.5.4 Алгоритм розв'язання.

Обробка замовлення складається з декількох етапів. В залежності від окремої платіжної системи ці етапи можуть несуттєво відрізнятись. Головні етапи обробки замовлення:

1. формування замовлення;
2. збереження його у базі;

3. перехід покупця до платіжної системи;
4. прийом повідомлення про результат оплати;
5. повернення покупця до крамниці.

Принцип роботи різних платіжних систем відрізняється обробкою пунктів 4 і 5. Деякі з існуючих платіжних систем передають код підтвердження платежу при поверненні, а деякі лише повідомлення про платіж.

Платіжні системи відрізняються типом можливих засобів, за допомогою яких покупець може здійснювати оплату. Усі платіжні Internet-системи, залежно від способу розрахунків, поділяють на:

- ✓ кредитні (які працюють з кредитними картками);
- ✓ дебетові (які працюють з електронними чеками і цифровою готівкою).

Алгоритм роботи наступний: спочатку інформаційна система авторизується у системі управління як управляючий аккаунт, що має певні права на створення аккаунтів та редагування пакетів. Після авторизації викликається потрібна функція в відповідності з замовленням. За отриманням результатів роботи функцій наша інформаційна система за допомогою системи повідомлень надає клієнту інформацію щодо виконання замовлення та переводє замовлення у відповідний статус.

3. Розробка робочої документації інформаційної системи хостингової компанії.

У технічному завданні (Додаток А) на розробку системи визначені вимоги до документації. Відповідно до цих вимог були розроблені наступні документи:

- ✓ SQL-дамп бази даних.
- ✓ Текст програми.
- ✓ Посібник користувача.
- ✓ Інструкція з формування і ведення бази даних.
- ✓ Посібник адміністратора з установки програмного забезпечення на серверах.
- ✓ Програма і методика випробувань.

4. Розрахунок економічного ефекту від впровадження ІС.

4.1 Загальні положення.

На даний момент сучасні обчислювальні засоби широко використовуються в повсякденній діяльності людини. Сьогодні обчислювальна техніка дозволяє вирішувати найрізноманітніші задачі: від звільнення людини від монотонної ручної праці до розробки складних технічних розрахунків, які не під силу виконати одній людині. Сучасні засоби програмування дозволяють вирішувати в домашніх умовах задачі, що стояли в недавньому минулому перед цілими колективами наукових співробітників.

Розвиток нових технологій і ріст застосування обчислювальної техніки привели до широкого застосування ЕОМ для вирішення проблем, пов'язаних з обробкою інформації, її контролем і аналізом. Сьогодні на основі засобів обчислювальної техніки розробляються і впроваджуються різні автоматизовані інформаційні системи. Успіх у впровадженні цих систем, їхня роль в інтенсифікації розвитку народного господарства нашої країни багато в чому залежить від фахівців із системотехніки, які знають методику аналізу і проектування цих систем, можливості обчислювальної техніки і які володіють математичними методами, що використовуються при постановці та вирішенні задач. Інформаційна система хостінгової компанії (далі система) призначена для автоматизації робочого процесу приватного підприємства. Впровадження системи приведе до значної економії, що виражається в наступних показниках:

- ✓ підвищення продуктивності праці;
- ✓ вивільнення робочого часу;
- ✓ вивільнення працівників для виконання іншої роботи.

Найбільш важливим моментом для розробника, з економічної точки зору, є процес формування ціни системи.

Створення автоматизованої системи вимагає одноразових витрат на її розробку, придбання необхідних технічних засобів і поточних витрат на функціонування системи. Економія від функціонування автоматизованої системи визначається з урахуванням витрат на її експлуатацію. Відношення цієї економії до витрат на створення автоматизованої системи характеризує економічну ефективність капітальних вкладень. Економічні показники визначаються за оптовими цінами, що діють на момент розрахунку, тарифами і ставками заробітної плати.

4.2 Розрахунок витрат на створення і впровадження системи.

Витрати на розробку ПЗ системи складаються з витрат на зарплату розробника, на амортизацію ЕОМ, на якій виконується розробка, на експлуатацію цієї ЕОМ, на засоби розробки та витрат на матеріали і

комплектуючі.

Припустимо розробка програмного забезпечення виконується програмістом, місячний оклад якого складає 15000 грн. Додаткова заробітна плата складає 20 % від основної. Виходячи з цього, основна і додаткова заробітна плата розроблювача системи 18000 грн./міс., а вартість сучасної ПК складає 24000 грн. (приведена вартість машини на базі Intel Core i5: Intel Core i5-7400 (3.0-3.5 ГГц)/RAM 16 ГБ/HDD 1 ТБ/nVidia GeForce GTX1050 Ti, 4 ГБ/без ОД/LAN/без ОС. При вартості кіловат-години електроенергії рівної 1,68 грн. (для населення, яке споживає понад 100 кВт·год до 600 кВт·год електроенергії на місяць), розраховується вартість розробки програми.

Витрати на допоміжні матеріали приведені в табл. 4.1.

Таблиця 4.1 – Витрати на допоміжні матеріали

Пункти витрат	Сума грн.
Папір	100,00
Заправлення картриджа до принтера	120,00
Канцтовари	180,00
Література	2000,00
Непередбачені витрати	500,00
Разом:	2800,00

Вартість програми розрахуємо по формулі:

$$C_{np} = (Z_{zn} + Z_{cz} + Z_{z2} + Z_e) \cdot T + Z_m, \quad (4.1)$$

де T – тривалість розробки, міс.

Z_{zn} – основна і додаткова заробітна плата обслуговуючого персоналу, грн.

Z_{cz} – відрахування на соціальні заходи (38 % від основної і додаткової заробітної плати), грн.

Z_{z2} – загальногосподарські витрати (10 % від основної заробітної плати), грн.

Z_e – витрати на електроенергію.

Z_m – витрати на основні і допоміжні матеріали.

Z_e при споживанні потужності 0.5 кВт, тривалості роботи на місяць, рівної $22 \cdot 6 = 132$ години, вартості кіловат-години електроенергії 1,68 грн обчислимо:

$$Z_e = 132 \cdot 1,68 \cdot 0,5 = 110,88 \text{ грн.}$$

Таблиця 4.2. – Витрати на розробку системи.

Найменування витрат	Одиниця виміру	Кількість
Тривалість розробки	Місяць	1
Основна і додаткова заробітна плата	Гривня	18000
Відрахування на соціальні заходи	Гривня	6840
Загальногосподарські витрати	Гривня	1500
Витрати на допоміжні матеріали	Гривня	2800
Витрати на електроенергію	Гривня	110,88

Відповідно до формули 4.1 вартість програми:

$$C_{np} = (18000 + 6840 + 1500 + 110,88) \cdot 1 + 2800 = 29250,88 \text{ грн.}$$

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

$$A_{об} = 24000 \cdot 0,6 = 14400 \text{ грн.}$$

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

$$B_m = 24000 \cdot 0,06 = 1200 \text{ грн.}$$

Річний обсяг робіт ПК у годинах визначається в такий спосіб:

$$\Phi_m = 264,5 \cdot T_z, \quad (4.2)$$

де T_z – середнє місячне завантаження устаткування (близько 4 годин);

264,5 – середня кількість робочих днів у році.

Отже, річний обсяг роботи ПЕОМ складе:

$$\Phi_m = 264,5 \cdot 4 = 1058 \text{ годин.}$$

Витрати на електроенергію $З_e$ при 1058 годинах роботи устаткування в рік складуть

$$З_e = 1058 \cdot 1,68 \cdot 0,5 = 888,72 \text{ грн.}$$

Експлуатаційні витрати для ПК за рік складуть:

$$З_{зр} = A_{об} + B_m + \Phi_m = 14400 + 1200 + 888,72 = 16488,72 \text{ грн.}$$

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

$$З_{се} = C_{np} + З_{зр} = 29250,88 + 16488,72 = 45739,60 \text{ грн.}$$

4.3 Економічна ефективність розробки і впровадження ІС.

Основним показником економічної ефективності функціонування системи є підвищення ефективності керування інформацією у вигляді зниження витрат на керування при одночасному збільшенні швидкості і якості одержання потрібного результату.

Крім багатьох інших негативних ефектів, ручна обробка інформації спричиняє наступні негативні економічні ефекти:

- ✓ високі витрати на складування паперових документів (сейфи, шафи, папки й ін.);
- ✓ підвищені витрати на канцтовари;
- ✓ витрати, пов'язані з роботою виявлення раніше допущених помилок (людський фактор).

До числа основних факторів, що визначають приріст прибутку в зв'язку з упровадженням програми, відносяться:

- ✓ підвищення продуктивності праці;
- ✓ вивільнення робочого часу.

Крім того, не піддається прямій грошовій оцінці підвищення оперативності керування, якість одержуваних результатів, поліпшення організації праці і т.д.

Обов'язковою умовою визначення економічної ефективності програми є порівнянність усіх показників у часі, за цінами й іншими нормами, використовуваними для визначення показників, за змістом і колом елементів витрат.

Визначимо пряму економічну ефективність, ґрунтуючись на тому, що впровадження програмного продукту вивільняє 0,7 працівника (за експертною оцінкою фахівців підприємства).

Зарплата 0,7 працівника в рік складає:

$$15000 \cdot 12 \cdot 0,7 = 126000 \text{ грн.}$$

Річний економічний ефект розраховується по формулі:

$$\mathcal{E}_{\text{год}} = \Delta C_n - E_n \cdot k,$$

де ΔC_n – вивільнені кошти після впровадження програмного продукту мінус експлуатаційні витрати: $\Delta C_n = 126000 - 16488,72 = 109511,3$ грн.;

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

k – одноразові витрати на впровадження продукту (45739,60).

$$\mathcal{Q}_{\text{од}} = 109511,3 - 0,6 \cdot 45739,60 = 82067,52 \text{ грн.}$$

Строк окупності системи розраховується по формулі:

$$T = \frac{k}{\Delta C} = \frac{45739,60}{109511,3} \approx 0,42 \text{ років.}$$

Отже строк окупності системи складає приблизно 5 місяців.

5. Охорона праці.

5.1 Вступ.

Охорона праці – система законодавчих актів, соціально-економічних, організаційних, технічних, гігієнічних, лікувально-профілактичних заходів і засобів забезпечення безпеки, збереження здоров'я і працездатності людини в процесі праці.

Як відомо, цілком безпечних і нешкідливих виробництв не існує, тому основною задачею охорони праці є зведення до мінімуму імовірність поразки чи захворювання працюючого з одночасним забезпеченням комфорту при максимальній продуктивності праці.

Комплекс організаційних і технічних заходів і засобів, що запобігають нещасні випадки, що виникають при впливі на працюючого небезпечних виробничих факторів, називається технікою безпеки. Виробнича санітарія містить у собі комплекс організаційних, гігієнічних і санітарно-технічних заходів, що запобігають виробничі шкідливості.

Турбота про охорону праці являється справою державної важливості. 14 жовтня 1992 року був прийнятий Закон України "Про охорону праці", у якому закріплені найважливіші положення в області охорони праці, і який поширюється на всіх суб'єктів підприємницької діяльності незалежно від форм власності. Основними пунктами в цьому законі є: стаття про гарантію прав громадян на охорону праці, стаття про відшкодування моральної шкоди, статті про охорону праці жінок, неповнолітніх, інвалідів.

У Законі "Про охорону праці" зазначені органи, покликані здійснювати нагляд і контроль за дотриманням законодавства про працю і правил по охороні праці. До них відносяться спеціально уповноважені на те державні органи й інспекції, що не залежать у своїй діяльності від адміністрації підприємств, установ, організацій і їхніх вищих органів.

Державне керування органами охорони праці здійснюють: Кабінет Міністрів України, державний комітет України по нагляду за охороною праці, міністерство й інші централізовані органи, місцеві ради народних депутатів і їхні виконавчі комітети. Державний нагляд за дотриманням законодавства й інших

нормативних актів по охороні праці здійснюють: державний комітет України по нагляду за охороною праці, державний комітет України по атомній і радіаційній безпеці, органи державного пожежного нагляду, органи і служби санітарно епідемічного нагляду. А трудові колективи в особі своїх уповноважених і профспілки здійснюють суспільний контроль за охороною праці.

Вищий нагляд за точним виконанням законів про працю міністерствами і відомствами, підприємствами, установами, громадянами і посадовими особами покладений на Генерального Прокурора України.

Велике значення в створенні здорових і безпечних умов праці має стандартизація. Вона дозволяє вжити діючих заходів по підвищенню технічного рівня й упорядкуванню розробки нормативно-технічної документації по безпеці праці. З 1972 року створюється Система стандартів по безпеці праці (ССБТ), що являє собою комплекс великого числа взаємозалежних стандартів, спрямованих на забезпечення безпеки праці. Ця система встановлює загальні вимоги і норми по видах небезпечних і шкідливих виробничих факторів, загальні вимоги безпеки до виробничого устаткування і процесів, вимоги до засобів захисту працюючих, методи оцінки безпеки праці.

5.2 Аналіз небезпечних і шкідливих виробничих факторів, характерних для приміщення відділу.

Реальні виробничі умови характеризуються, як правило, наявністю деяких небезпечних і шкідливих виробничих факторів.

До небезпечних виробничих факторів відносяться фактори, вплив яких приводить до травми, фактори, що приводять до професійного захворювання працюючого. По природі дії небезпечні і шкідливі фактори за ДСТ 120.003-74 діляться на:

- ✓ фізичні;
- ✓ хімічні;
- ✓ біологічні;
- ✓ психологічні.

Впровадження обчислювальної техніки в усі сфери народного господарства привело до значного розширення парку ПЕОМ. У зв'язку з цим

виникла проблема організації робочих місць користувачів ПЕОМ відповідно до вимог нормативів охорони праці.

Робота з персональним комп'ютером зв'язана з низкою шкідливих і небезпечних факторів, що можуть впливати на організм людини.

За даними Всесвітньої організації охорони здоров'я, професійна діяльність зв'язана з роботою на ПЕОМ може в окремих випадках приводити до порушень з боку зорового аналізатора, кістково-м'язовим (змушена поза) порушенням, нервово-емоційною напругою при роботі, шкірним захворюванням і ін. Високий відсоток (92%) складають скарги на зорове стомлення. Дослідження показали, що професія користувача ЕОМ являє собою модель розумової праці, що виконується в одноманітній позі в умовах обмеження загальної м'язової активності при рухливості рук, при високій напрузі зорових функцій і нервово-емоційній напрузі в умовах впливу різноманітних фізичних факторів. Дослідження умов праці показали, що на більшості підприємств користувачі ЕОМ працюють при несприятливому мікрокліматі, зниженій вологості повітря, нераціональному освітленні. Підвищений шум на робочому місці, унаслідок роботи комп'ютерів і принтерів, шкідливо впливає на органи слуху і на весь організм людини через центральну нервову систему, викликаючи швидку стомлюваність, головний біль, безсоння.

У зв'язку з використанням клавіатури можуть виникнути захворювання рук, спини, пліч і шиї.

Отже, підводячи підсумок, при роботі із ЕОМ можна виділити наступні фактори:

✓ фізичні – низькочастотне електромагнітне випромінювання моніторів, яскраві відблиски на екрані, підвищена яскравість світла монітора, недостатня освітленість робочого місця, запиленість повітря поблизу екрана дисплея через його електризацію, підвищення рівня шуму, підвищена напруженість електричних і магнітних полів, підвищення рівня статичної електрики, небезпека поразки електричним струмом;

✓ психофізіологічні – тривалі статичні навантаження м'язів рук, шиї, спини, зап'ясть, пліч, ніг, постійна напруга зору, нервово-емоційні перевантаження (монотонність виконуваних дій, розумова перенапруга).

Встановлено наступні санітарні норми і правила:

- ✓ мікроклімат у робочій зоні приміщення (до 2 м над рівнем підлоги) для легкої роботи (витрати енергії до 150 ккал/година);
- ✓ температура 17-22 С°;
- ✓ відносна вологість не більш 75%;
- ✓ швидкість руху повітря не більш 0.3 м/с.;
- ✓ припустимі коефіцієнти пульсації штучного освітлення в залежності від системи освітлення і характеру виконуваної роботи не повинні перевищувати 10-20%;
- ✓ освітленість при комбінованому освітленні 150-400 лк, при загальному освітленні – 200 лк, коефіцієнт запасу – 1.5 (для люмінесцентних ламп);
- ✓ рівень шуму – 50 дБА в кімнаті програмістів і до 65 дБА в машинному залі;
- ✓ гранично припустима напруженість електричної складової електромагнітного поля 10 В/м, магнітної складової – 0.3 А/м;
- ✓ гранично припустима напруженість електростатичного поля при тривалому впливі на людину на робочому місці (до 1 години) – 60 кВ/м.

5.3 Розрахунок системи освітлення для приміщення відділу.

Рациональне освітлення приміщень – один з найбільш важливих факторів, від яких залежить ефективність трудової діяльності людини. Гарне освітлення необхідне для виконання більшості задач оператора. Для того, щоб спланувати раціональну систему освітлення, враховується специфіка робочого завдання, для якого створюється система освітлення, швидкість і точність, з яким це робоче завдання повинне виконуватися, тривалість його виконання і різні зміни в умовах виконання робочих операцій.

Розрахунок освітленості будемо робити для приміщення, що має наступні характеристики:

- довжина приміщення $A=16$ м;
- ширина приміщення $B=6$ м;
- висота $h=4$ м;
- число вікон 3;
- кількість робочих місць 3.

У приміщенні, де знаходиться робоче місце оператора використовується змішане освітлення, тобто сполучення природного і штучного освітлення. У якості природного – бічне освітлення через вікна. Штучне освітлення використовується при недостатнім природному освітленні. У даному приміщенні використовується загальне штучне освітлення.

Розрахунок його здійснюється по методу світлового потоку з урахуванням потоку, відбитого від стін і стелі. Нормами для даних робіт установлена необхідна освітленість робочого місця $E_n=300$ лк (середня точність роботи з розрізнення деталей розміром від 1 до 10 мм).

Загальний світловий потік визначається за формулою

$$F_{заг} = \frac{E_n \cdot S \cdot z_1 \cdot z_2}{V},$$

де E_n – нормована освітленість ($E_n=300$ лк);

S – площа приміщення $S = A \cdot B = 16 \cdot 6 = 96$ м²;

$z_1=1,5$ – коефіцієнт, що враховує старіння ламп і забруднення світильників;

$z_2=1,1$ – коефіцієнт, що враховує нерівномірність висвітлення приміщення;

V – коефіцієнт використання світлового потоку; визначається в залежності від коефіцієнтів відображення від стін, стелі, робочих поверхонь, типів світильників і геометрії приміщення.

$$F_{заг} = \frac{300 \cdot 96 \cdot 1,5 \cdot 1,1}{0,34} = 139764,7 \text{ лм/}$$

Виберемо з таблиці коефіцієнт використання світлового потоку по наступним даним:

- ✓ коефіцієнт відображення побіленої стелі РП=70%;
- ✓ коефіцієнт відображення від стін, пофарбованих у світлу фарбу RCT=50%;
- ✓ коефіцієнт відображення від підлоги, покритого лінолеумом темного кольору RP=10%;

✓ індекс приміщення: $i = \frac{A \cdot B}{h(A + B)} = \frac{16 \cdot 6}{4(16 + 6)} = 1,1.$

Таким чином, коефіцієнт $V=0,34$.

Для організації загального штучного освітлення виберемо лампи типу ЛБ40. Люмінесцентні лампи мають ряд переваг перед лампами накаливання: їхній спектр ближче до природного; вони мають велику економічність (більше світловіддача) і термін служби (у 10-12 разів). Поряд з цим маються і недоліки: їхня робота супроводжується іноді шумом; гірше працюють при низьких температурах; їх не можна застосовувати у вибухонебезпечних приміщеннях; мають малу інерційність.

Для нашого приміщення люмінесцентні лампи підходять.

Світловий потік однієї лампи ЛБ40 складає не менш $F_l = 2810$ лм. Число N ламп, необхідних для організації загального освітлення визначається по формулі:

$$N = \frac{F_{\text{заг}}}{F_l} = \frac{139764,7}{2810} = 50.$$

Вибираємо світильники ПВЛ-1, 2x40 Ут.

Таким чином, щоб забезпечити світловий потік $F_{\text{заг}} = 139764,7$ лм треба використовувати 25 світильників по 2 лампи ЛБ40 у кожному.

Електрична потужність однієї лампи ЛБ40 $W_l = 40$ Вт.

Потужність всієї освітлювальної системи

$$W_{\text{заг}} = W_l \cdot N = 40 \cdot 50 = 2000 \text{ Вт.}$$

5.4 Розробка заходів щодо забезпечення сприятливих умов праці при роботі з персональним комп'ютером.

Сьогодні фахівці в області ергономіки вже зрозуміли, що не можна знайти ідеальне положення, у якому можна перебувати і працювати протягом усього робочого дня. Для більшості людей комфортабельним робочим місцем повинне бути таке, котре можна пристосувати не менш чим для двох позицій, при цьому положення крісла, монітора повинні щораз відповідати виконуваний роботі і звичкам. Багато хто вважають, що для роботи на комп'ютері більше підходять вертикальне і злегка похиле положення. Можливо, щоб крісло було злегка нахилене вперед.

Дисплей. Положення тіла звичайно відповідає напрямку погляду; дисплеї, розташовані занадто низько чи під неправильним кутом, є основними

причинами сутулості. Відстань від дисплея до очей може варіюватися в залежності від характеру виконуваної роботи – 40-70 см. При роботі з текстом відстань від екрана до очей повинна лише небагато перевищувати відстань між книгою й очима і складати 40-45 см. Необхідно давати відпочинок очам і час від часу їх просто закривати. Не можна допускати, щоб очі увесь час були сфокусовані на одній відстані, працюючи з текстом, рекомендується установлювати великий шрифт. Дуже важливо, щоб екран монітора не мерехтів. Частота регенерації зображення повинна бути не менш 72 Гц, тому що при більш низькій частоті помітне мерехтіння, хоча люди з особливо чутливим зором можуть помітити його і при більш високій частоті. Їм доведеться підібрати графічну плату і монітор з частотою регенерації 85 Гц і вище.

Рекомендується встановлювати на екран монітора спеціальні скляні поляризаційні фільтри з заземленням. Необхідно уникати того, щоб термінал був звернений екраном убік вікна, оскільки інтенсивна освітленість області зору може затопити потоками світла очі і розмити зображення на сітківці. Якщо приходить сидіти поруч з вікном, то треба розташуватися під прямим кутом до нього, причому екран дисплея повинний бути перпендикулярний шибці – цим виключаються відблиски на екрані.

Для зниження впливу низькочастотного електромагнітного випромінювання монітора на ЕЛТ рекомендується вибирати монітор, що задовольняє стандарту MPR II, чи TCO 92-95. У протилежному випадку необхідно установити на екран захисний фільтр, найбільш сучасні моделі, затримують до 99 % електромагнітного випромінювання.

Форма спинки крісла повинна повторювати форму спини працюючого. Крісло необхідно установити на такій висоті, щоб не почувався тиск на куприк (крісло розташоване занадто низько) на стегна (крісло розташоване занадто високо). Фахівці з ергономіки, вважали, що кут між стегнами і хребтом повинний складати 90°, однак недавно проведені дослідження показали, що більшість людей переважно сидять трохи відкинувшись.

Клавіатуру необхідно установити так, щоб не треба було до неї тягтися і нахилятися вперед. При зміні положення тіла (наприклад, з вертикального на похиле) обов'язково треба перемінити положення клавіатури і дисплея. Може

виявитися корисною регульована підставка клавіатури, завдяки якій можна без усякої напруги працювати з маніпулятором "миша". Можна поставити клавіатуру і на коліна. Руки при роботі повинні бути прямими в зап'ястях і зігнуті в ліктях приблизно під прямим кутом. Пальці також повинні бути злегка зігнуті. Удари по клавішах не повинні бути занадто сильними.

Зручна висота столу особливо важлива в тому випадку, коли на ньому розташовується клавіатура. Якщо в клавіатури немає підставки, а висоту столу не можна змінити (і він занадто високий), то треба вище підняти сидіння крісла, а під ноги підставити лавочку: чи що-небудь інше. Якщо стіл занадто низький, необхідно підкласти що-небудь під його ніжки.

Якщо при роботі часто приходиться дивитися на документи, треба установити підставку з оригіналом документа в одній площині і на одній, висоті з екраном. Якщо треба частіше дивитися на оригінал, чим на екран, необхідно повернути крісло й екран таким чином, щоб оригінал розміщався прямо перед очима, а комп'ютер ледве збоку.

Щогодини необхідно робити перерву в роботі. У цей час рекомендується робити вправи для зап'ясть.

6. Охорона навколишнього середовища.

6.1 Вступ.

За останні десятиліття кількість шкідливих, відходів підприємств настільки зросло, що природа поступово втрачає здатність до самоочищення. Стало очевидним, що будь-яка нова техніка і технологія, навіть високопродуктивна й економічно вигідна, але не задовольняючим сучасним вимогам охорони праці і навколишнього середовища, не є прогресивною і не відповідає основному напрямку технічного прогресу. Тому охорона навколишнього середовища в даний час здобуває першорядне значення у всіх областях діяльності, людини.

Сучасна політика держав в області охорони навколишнього середовища від забруднення ґрунтується на принципі "забруднив – плати". Уперше на міжнародному рівні даний принцип був висунутий Організацією економічного співробітництва і розвитку в 1972 році. З цього часу цей принцип став активно використовуватися в законодавчій практиці європейських і інших країн світу.

Для постійного контролю за станом навколишнього середовища застосовують різні методи, одним з них є екологічний моніторинг навколишнього середовища.

6.2 Моніторинг як система спостереження і контролю навколишнього середовища.

Моніторингом навколишнього середовища називаються регулярні, виконувані по заданій програмі спостереження природних середовищ, природних ресурсів, рослинного і тварини світу, що дозволяють виділити їхній стан і процеси що відбуваються в них під впливом антропогенної діяльності.

Під екологічним моніторингом варто розуміти організований моніторинг навколишнього середовища, при якому, по-перше, забезпечується постійна оцінка екологічних умов середовища життя людини і біологічних об'єктів (рослин, тварин, мікроорганізмів і т.д.), а також оцінка стану і функціональної цінності екосистеми, по-друге, створюються умови для визначення корегувальних дій у тих випадках, коли цільові показники екологічних умов не досягаються.

У систему моніторингу повинні входити наступні основні процедури:

- ✓ виділення (визначення) об'єкта спостереження;
- ✓ обстеження виділеного об'єкта спостереження;
- ✓ складання інформаційної моделі для об'єкта спостереження;
- ✓ планування спостережень;
- ✓ оцінка стану об'єкта спостереження й ідентифікація його інформаційної моделі;
- ✓ прогнозування зміни стану об'єкта спостереження;
- ✓ представлення інформації в зручній для використання формі і доведення її до споживача.

Основні задачі екологічного моніторингу:

- ✓ спостереження за джерелом антропогенного впливу;
- ✓ спостереження за фактором антропогенного впливу;
- ✓ спостереження за станом природного середовища під впливом факторів антропогенного впливу й оцінка прогнозованого стану природного середовища.

При розробці проекту екологічного моніторингу необхідна наступна інформація:

✓ джерело надходження забруднюючих речовин у навколишнє середовище – викиди забруднюючих речовин в атмосферу промисловими, енергетичними, транспортними й іншими об'єктами; скидання стічних вод у водяні об'єкти; поверхневі змиви забруднюючих і біогенних у поверхневі води суші і моря; внесення на земну поверхню і (чи) у ґрунтовий шар забруднюючих і біогенних речовин разом з добривами і ядохімікатами при сільськогосподарській діяльності; місця поховання і складування промислових і комунальних відходів; техногенні аварії, що приводять до викиду в атмосферу небезпечних речовин чи розливу рідких забруднюючих і небезпечних речовин, і т.д.;

✓ переноси забруднюючих речовин – процеси атмосферного переносу; процеси переносу і міграції у водяному середовищі;

✓ процеси ландшафтно-геохімічного перерозподілу забруднюючих речовин – міграція забруднюючих речовин по ґрунтовому профілю до рівня ґрунтових вод; міграція забруднюючих речовин по ландшафтно-геохімічному сполученню з урахуванням геохімічних бар'єрів і біохімічних круговороті.

Спостереження за цими процесами доцільно проводити періодично на спеціально виділеній системі пунктів: контрольні водозбори – катени – площадки – створи;

✓ дані про стан антропогенних джерел емісії – потужність джерела емісії і місце розташування його, гідродинамічні умови надходження емісії в навколишнє середовище.

6.3 Методи контролю забруднюючих речовин в об'єктах навколишнього середовища.

Сучасні методи контролю хімічних речовин, що забруднюють навколишнє середовище – це по суті фізико-хімічні методи. Іноді їх поєднують терміном "інструментальні методи аналізу". Дана тема величезна, тому ми розглянемо лише найбільш важливі з фізико-хімічних методів, що оптимально сполучають у собі цілий ряд якостей: високу точність і відтворюваність результатів аналізу, високу чутливість і, незважаючи на ці тверді вимоги, приступність апаратури і можливість швидкого її освоєння.

Газова хроматографія. В основу методу газової хроматографії покладений наступний принцип: аналіз суміші речовин у результаті розподілу компонентів між фазами, що не змиваються, одна з яких рухлива – інертний газ (азот, гелій і ін.), інша – нерухома (висококип'яча рідина чи тверда фаза).

Цей метод має два варіанти: газоабсорбційна і газорідинна хроматографія.

Поділ компонентів суміші відбувається в хроматографічному стовпчику. Хроматографічні стовпчики: набивні (довжина – 1-3м, діаметр – 4мм, матеріал – скло, сталь і ін.) і капілярні (довжина – до 50м, матеріал – скло, кварц).

Вибір нерухомої фази (Нф). Ефективність стовпчика (здатність розділяти складні суміші на окремі компоненти) залежить від розміру часток, на які нанесена рідка фаза. Вона зростає при використанні однорідних часток малого розміру. Для стандартних набивних стовпчиків оптимальний розмір часток 0,12-0,17 мм. Необхідно враховувати їхню близькість до аналізованих з'єднань. Для аналізу полярних компонентів застосовують полярні фази, для аналізу неполярних компонентів – менш полярні чи цілком неполярні.

Неполярні фази для газоадсорбційної хроматографії силікагель, оксид

алюмінію, цеоліти, полімерні сорбенти (наприклад, полисорб, поропак і ін.).

Рухливі фази: азот, гелій, аргон, пари води.

Детектори. Історія розвитку газової хроматографії – це історія появи і розвитку детекторів для хроматографії. Застосовуються кілька типів детекторів.

1. Детектор теплопровідності (ДТП) чи катетометр. Принцип його дії заснований на розходженні теплопровідностей аналізованої речовини і газу-носія.

2. У детекторі іонізаційно плазмовому (ПІД чи ДІП) використовується залежність електропровідності простору між електродами від числа іонізованих часток, що знаходяться в ньому, що утворюються у водневому полум'ї під дією термічних і окисних процесів при влученні в нього молекул аналізованої речовини.

3. Електронно-загарбний детектор (ЕЗД), чи детектор по захопленню електронів, як і ДІП, заснований на залежності електропровідності проміжку між електродами і числимо іонів, що знаходяться в цьому проміжку, що зв'язано з числом молекул, що надходять у детектор. Однак механізм і спосіб утворення іонів принципово відрізняються від такого у випадку ДІП – іони утворюються в результаті взаємодії молекул аналізованої речовини і потоку електронів у камері детектора в результаті бета-розпаду радіоактивної речовини.

4. Детектор термоіонний (ДТІ) за принципом дії аналогічний ДІП. Однак додатково у водневе полум'я безупинно надходить потік іонів лужних металів (калій, натрій, цезій). У їхній присутності різко зростає ефективність іонізації з'єднань, що містять азот, фосфор, хлор і ін. ДТІ застосовують для визначення ФОС і азотовмісних з'єднань.

5. Полум'яно-фотометричний детектор (ПФД) селективний і має підвищену чутливість стосовно з'єднань, що містять сірку.

Високоєфективна рідинна хроматографія (ВЕРХ) – хроматографічний метод, що дозволяє розділити висококип'ячі рідини і (чи) тверді речовини, що важко або недоцільно визначати методом газорідинної хроматографії, наприклад поліциклічні ароматичні вуглеводні, амінокислоти, ПАВ, пестициди, лікарські препарати, вуглеводи й ін.

Тонкошарова хроматографія (ТШХ). Поділ відбувається на спеціальних

пластинках для тонкошарової хроматографії. Нерухома фаза в ТШХ: силікагель, оксид алюмінію, іонообмінні смоли з добавками крохмалю і гіпсу.

Аналізовану суміш наносять на стартову лінію мікрошприцом чи мікропіпеткою. Пластинку чи папір з нанесеною пробою поміщають у закриту камеру, що містить розчинник, який переміщається по шару сорбенту (чи по паперові) під дією капілярних сил. Компоненти суміші переміщаються разом з розчинником з різними швидкостями. По закінченні поділу пластинку чи папір виймають з камери, випаровують розчинник, обробляючи струменем теплого повітря. Обумовлені речовини з'являються на хроматограмі у виді плям у результаті обробки спеціальним реактивом (наприклад, нігдрином при аналізі амінокислот) чи методом флюоресценції. Зміст аналізованого компонента пропорційно площі плям. Кількісну оцінку чи проводять безпосередньо на пластинці за допомогою планіметра, чи шляхом зняття пофарбованої плями з хроматограми екстракції речовини розчинником і визначення його зміст фотометричним чи методом за допомогою денситометра.

При аналізі катіонів колонку для поділу заповнюють сульфонованими катіонами низкою ємності, а гнітючу колонку-аніонітом високої ємності. У якості елюентів використовують розчини HCl HNO_3 , гідро хлориду піридину й ін.

В останні роки розвивається іонна хроматографія без придушення фонового сигналу елемента і з різними способами детектування: фотометричний, атомно абсорбційний, іонометричний.

Достоїнства методу: низька межа визначення – $1 \cdot 10^{-6}$ мг/мол, селективність, можливість одночасного визначення неорганічних і органічних іонів експресність, широкий діапазон обумовлених концентрацій.

Застосовують вітчизняний хроматограф “Колір-300Б”, кондуктометричний детектор, мікропроцесор. Межа виявлення по хлориду натрію – $3 \cdot 10^{-6}$ мг/мол.

Хроматомасс-спектрометрія (ХМС) – це в сутності газова хроматографія з мас-спектрометром як детектор (наприклад, МІ-1201). Даний метод дозволяє розшифровувати склад складних сумішей, що містять сотні невідомих компонентів, і визначати їх по одній пробі.

Поляррографія. Це один з електрохімічних методів аналізу. Поляррограма – залежність сили струму від величини прикладеної напруги на електроди. При

цьому методі не відбувається фізичного поділу суміші на окремі компоненти. Як катод найчастіше застосовують ртутний електрод, що капає, (РКЕ), поверхня якого безупинна обновляється, що дозволяє одержувати полярограми і проводити аналіз з високою відтворюваністю результатів.

Пряме визначення можливе лише при наявності речовин, здатних відновлюватися на РКЕ: іони металів, органічні сполуки, що містять галоїд-, нітро-, карбонільні з'єднання, пероксиди, дисульфідів, і т.д. Це трохи обмежує можливості методу, однак при визначенні поліграфічних активних з'єднань дозволяє досягти високої селективності визначення без попереднього поділу складних сумішей на окремі компоненти.

Основні типи полярографії – постійна-токова (класична) і перемінно-токова. Остання має різні назви (підрозділи): у залежності від форми амплітуди перемінного струму – квадратно-хвильова.; у залежності від полярності електрода, що використовують як індикаторний, – катодна (відновлення) чи анодна (окислювання). Останню іноді називають вольтамперометрією. В анодній полярографії на відміну від катодної використовують тільки твердий електрод (наприклад, графітовий). Застосовують фоновий чи індиферентний електроліт (називаний просто – тло), тобто розчин кислоти, солі, буферний розчин більш складного складу, у якому розчиняють аналізовану пробу.

ВИСНОВКИ

У результаті виконання даної дипломної роботи проведено розширення засобів здійснення електронних платежів і забезпечення режиму реального часу роботи клієнта з замовленнями у інформаційній системі хостинг-компанії "MyHost".

Для досягнення вказаних результатів вирішено наступні задачі:

- ✓ проведено аналіз сучасних інформаційних технологій, які використовуються для розробки інформаційних систем;
- ✓ сформувано вимоги до системи;
- ✓ розроблено концепцію системи;
- ✓ проведено техніко-економічне обґрунтування розробки системи;
- ✓ розроблено загальносистемні рішення, рішення з інформаційного, організаційного, технічного і програмного забезпечення;
- ✓ розроблено необхідне програмне забезпечення системи;
- ✓ розроблено відповідну документацію.

Впровадження створеної інформаційної системи дозволяє підвищити ефективності контролюючих та управлінських дій хостингової компанії, знизити вартості послуг компанії, що розширить коло потенційних клієнтів, дає можливість виходу компанії на міжнародний ринок.

СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ

1. Безопасность жизнедеятельности. Под редакцией Белова Н.А. – М.: Знание, 2000. – 364с.
2. Вендров А.М. Практикум по проектированию программного обеспечения экономических информационных систем: Учеб. пособие. – М.: Финансы и статистика, 2002. – 192 с.
3. ЄСКД "Загальні вимоги до текстових документів" (ДОСТ 2.105-95).
4. Жидецький В.Ц., Джигірей В.С., Мельников О.В. Основи охорони праці. Навчальний посібник. – Вид. 4-те, доповнене. – Львів: Афіша, 2000. – 350с.
5. Жидецький В.Ц. Охорона праці користувачів комп'ютерів. Навчальний посібник. – Вид. 2-ге, доповнене. – Львів: Афіша, 2000. – 176с.
6. Кастаньетто Дж., Рават Х., Шуман С., Сколло К., Велиаф Д. Профессиональное PHP программирование. – Пер. с англ. – СПб: Символ-Плюс, 2001. – 912 с.
7. Коржинский С.Н. Настольная книга Web-мастера: Эффективное применение HTML,CSS, JAVASCRIPT. – СПб: Издательство "КноРус", 2000. – 320 с.
8. Макарова М.В. Електронна комерція: Посібник для студентів вищих навчальних закладів. – К.: Видавничий центр "Академія", 2002. – 272 с.
9. Проектування інформаційних систем. Навчальний посібник (затв. МОН України) // Литвин В.В., Шаховська Н.Б. / За ред. Пасічника В.В. – Львів. Магнолія 2006, 2011. – 380с.
10. Стив Суэринг, Тим Конверс, Джойс Парк. PHP и MySQL. Библия программиста. – К.: Диалектика, 2010. – 912 с.

Технічне завдання**1 Загальні положення.**

Необхідно розробити проект інформаційної системи автоматизованого управління замовленнями з можливістю вибору платіжних систем хостингової компанії.

Назва розробки: "Проект інформаційної системи хостингової компанії "MyHost".

2 Підстава для розробки.

Підставою для розробки інформаційної підсистеми є завдання на дипломну роботу "Розробка інформаційної системи хостингової компанії "MyHost".

3 Призначення і мета створення інформаційної системи.

Призначенням системи є автоматизація функцій, що виконують адміністратори компанії при надходженні замовлень та розширення переліку способів оплати.

Метою створення системи є підвищення ефективності і якості роботи компанії по наданню послуг.

4 Вимоги до інформаційної підсистеми.**4.1 Вимоги до функціональних характеристик.**

Інформаційна система повинна володіти рядом функціональних характеристик, які можна згрупувати в такий спосіб:

1. Користування клієнтами:

- ✓ реєстрація клієнтів;
- ✓ створення замовлень;
- ✓ аутентифікація клієнтів;
- ✓ підтримка різних систем оплати;
- ✓ вибір клієнтом систем оплати та встановлення параметрів кожної системи оплати;

- ✓ перехід клієнта на сайт платіжної системи для оплати свого замовлення;
- ✓ перегляд клієнтом переліку замовлень та інформування про оплату замовлень;

- ✓ зміна клієнтом стану замовлення;

- ✓ конфіденційність передачі даних по комп'ютерній мережі.

2. Надання системи передачі повідомлень:

- ✓ можливість створювати повідомлення для користувачів, що зареєстровані у системі (у відповідності до бізнес-логіки системи);

- ✓ перегляд адміністратором та покупцями повідомлень, що були написані для них та можливість відповідати на ці повідомлення.

3. Взаємодія з системою управління хостингом:

- ✓ автоматичне створення аккаунта клієнта у відповідності до замовленого пакету;

- ✓ редагування аккаунту клієнта.

4. Адміністрування:

- ✓ надання можливості адміністратору створювати нові пакети та редагувати або видаляти існуючі;

- ✓ прийняття замовлень які не можуть бути автоматизовані (послуги, що діють за партнерськими програмами, реєстрація доменного імені, встановлення SSL-сертифікату, тощо);

- ✓ надання можливості адміністратору переглядати замовлення клієнтів та здійснювати контроль оплати;

- ✓ адміністрування клієнтських аккаунтів.

5. Можливість віддаленої роботи з інформаційною системою як у локальній мережі, так і в Інтернет.

4.2 Вимоги до інформаційного забезпечення.

Система повинна бути відкритою, тобто мати властивості мобільності.

Система повинна підтримувати розподілену дворівневу архітектуру клієнт-сервер і мати зручний веб-інтерфейс, який забезпечує доступ до інформації через мережу Інтернет.

Серверні додатки повинні генерувати документи і форми для відображення необхідної інформації мовою HTML. Генерація документів

повинна здійснюватися на основі інформації, що зберігається у базі даних.

Система повинна забезпечувати захист даних шляхом аутентифікації й ідентифікації користувачів, захисту каналів зв'язку (шифрування даних), фізичного захисту устаткування.

В якості сервера бази даних необхідно використовувати серверну СУБД із наявністю засобів веб-доступу до баз даних у сполученні з відносно невисокою ціною рішення. Необхідно використовувати стандартні можливості СУБД із метою підтримки переносимості.

4.3 Вимоги до лінгвістичного забезпечення.

Веб-інтерфейс повинен бути реалізований мовою HTML.

Функціональність на стороні клієнта ("тонкий» клієнт") повинна розроблятися мовою JavaScript.

Серверні додатки необхідно розробляти мовою PHP.

4.4 Вимоги до програмного забезпечення.

Інформаційна система включає веб-сервер, сервер баз даних та клієнтські комп'ютери.

Вимоги до програмного забезпечення веб-сервера:

- ✓ операційна система: Free BSD 8.0;
- ✓ веб-сервер: Apache Server 2.0 і вище, з встановленим PHP у якості модуля веб-сервера (mod_php);
- ✓ СУБД MySQL 5 і вище.

Вимоги до програмного забезпечення клієнтських комп'ютерів:

- ✓ будь-яка операційна система з можливістю установки віддаленого мережного з'єднання і графічним інтерфейсом;
- ✓ будь-який web-оглядач з підтримкою HTML, JavaScript.

4.5 Вимоги до технічного забезпечення.

Технічне забезпечення веб-сервера:

- ✓ процесор: Intel® Core™ i7-920 Quad-Core 4 фізичних ядра, 4 віртуальних, підтримка технології hyper-threading;
- ✓ дискова система: 1500 GB HDD SATA2;

- ✓ оперативна пам'ять: 12 GB DDR3 5;
- ✓ мережна карта Ethernet, що підтримує швидкості передачі 1000 Мбіт/сек.
- ✓ наявність бекап-серверу.

Технічні засоби клієнтських комп'ютерів повинні мати будь-яку операційну систему з веб-браузером. Конфігурація комп'ютера користувача може бути наступною:

- ✓ мікропроцесор: Intel® Core™ i3;
- ✓ оперативна пам'ять RAM 2048 Mb;
- ✓ жорсткий диск: HDD 300 Gb;
- ✓ мережна карта Ethernet.

4.6 Вимоги до документації.

Необхідна наявність наступних документів:

- ✓ технічне завдання;
- ✓ інструкція користувача;
- ✓ опис структури програмного забезпечення і бази даних;
- ✓ інструкція з адміністрування бази даних;
- ✓ програма і методика випробувань;
- ✓ текст програми;
- ✓ дамп бази даних.

4.7 Порядок контролю і прийому системи.

Перевірка документації інформаційної системи робиться замовником з метою зафіксувати відповідність чи невідповідність системи технічному завданню (постановці задачі).

Тестування системи виконує розроблювач протягом створення системи без взаємодії з замовником. Процес тестування представляє собою експлуатацію системи в контрольованих умовах і вивчення отриманих результатів. При цьому перевіряється робота системи з нормальними і помилковими даними і подіями. Тестування розподілених систем ускладнене безліччю взаємозв'язків. Кожна залежність, веб-сторінка, компонент, база даних, а також код графічного інтерфейсу, програмне забезпечення середнього рівня і мережна інфраструктура повинні бути протестовані не тільки на

коректність роботи, але і на сумісність один з одним у різних конфігураціях.

Найкраще для тестування розподілених систем застосовувати метод "знизу-нагору". Спочатку розроблювач повинний перевірити кожен компонент (елемент системи) окремо, потім протестувати взаємодію компонента з іншими компонентами системи на автономному комп'ютері і тільки потім перевірити роботу всієї системи в розподіленому середовищі. Усі ці види тестування виконуються розроблювачем без участі замовника.

Приймання системи виконується замовником. Система вважається придатною, якщо вона задовольняє всім пунктам даного технічного завдання (постановки задачі). Замовник і/або його представники самостійно проводять тестування усіх функцій системи методом "чорної шухляди" у присутності розроблювача. При виявленні помилок у роботі системи складається їхній перелік і обговорюється термін їхнього виправлення розроблювачем. Після цього замовник проводить повторне тестування.

Після виправлення помилок або у випадку, якщо помилки не були знайдені, а також при відповідності системи встановленим у технічному завданні вимогам, підписується акт приймання системи.

SQL-дамп бази даних

```
--
-- Структура таблиці `accounts`
--
CREATE TABLE IF NOT EXISTS `accounts` (
  `is` int(11) NOT NULL auto_increment,
  `first_name` varchar(100) NOT NULL,
  `last_name` varchar(100) NOT NULL,
  `email` varchar(50) NOT NULL,
  `country` varchar(100) NOT NULL,
  `city` varchar(100) NOT NULL,
  `phone` varchar(100) NOT NULL,
  `date_reg` timestamp NOT NULL,
  `comm` varchar(255) NOT NULL,
  PRIMARY KEY (`is`)
);

--
-- Структура таблиці `accounts_auth`
--
CREATE TABLE IF NOT EXISTS `accounts_auth` (
  `id` int(11) NOT NULL auto_increment,
  `login` varchar(30) NOT NULL,
  `passw` varchar(40) NOT NULL,
  `date` timestamp NOT NULL,
  PRIMARY KEY (`id`)
);

--
-- Структура таблиці `invoice`
--
CREATE TABLE IF NOT EXISTS `invoice` (
  `id` int(11) NOT NULL auto_increment,
  `account_id` int(11) NOT NULL,
  `service_id` int(11) NOT NULL,
  `setup_date` timestamp NOT NULL,
  `due_date` timestamp NOT NULL default '0000-00-00 00:00:00',
  `more_info` varchar(255) NOT NULL,
  PRIMARY KEY (`id`)
);

--
-- Структура таблиці `orders`
--
CREATE TABLE IF NOT EXISTS `orders` (
  `id` int(11) NOT NULL auto_increment,
  `account_id` int(11) NOT NULL,
  `setup_date` timestamp NOT NULL,
  `approval_date` timestamp NOT NULL default '0000-00-00 00:00:00',
```



```
`service_id` int(11) NOT NULL,  
`invoice_id` int(11) NOT NULL,  
`due_date` timestamp NOT NULL default '0000-00-00 00:00:00',  
`more_info` varchar(255) NOT NULL,  
`status` int(1) NOT NULL,  
PRIMARY KEY (`id`)  
);
```

```
--  
-- Структура таблицы `payments`  
--
```

```
CREATE TABLE IF NOT EXISTS `payments` (  
  `id` int(11) NOT NULL auto_increment,  
  `account_id` int(11) NOT NULL,  
  `payment_date` timestamp NOT NULL,  
  `amount` float NOT NULL,  
  `orders_id` int(11) NOT NULL,  
  `paymet_system_id` int(11) NOT NULL,  
  PRIMARY KEY (`id`)  
);
```

```
--  
-- Структура таблицы `payment_system`  
--
```

```
CREATE TABLE IF NOT EXISTS `payment_system` (  
  `id` int(11) NOT NULL auto_increment,  
  `title` varchar(100) NOT NULL,  
  `annot` varchar(255) NOT NULL,  
  `name_file` varchar(20) NOT NULL,  
  `param` varchar(255) NOT NULL,  
  `status` int(1) NOT NULL,  
  PRIMARY KEY (`id`)  
);
```

```
--  
-- Структура таблицы `services`  
--
```

```
CREATE TABLE IF NOT EXISTS `services` (  
  `id` int(11) NOT NULL auto_increment,  
  `title` varchar(100) NOT NULL,  
  `annot` varchar(255) NOT NULL,  
  `text` text NOT NULL,  
  `cost` float NOT NULL,  
  PRIMARY KEY (`id`)  
);
```

```
--  
-- Структура таблицы `tickets`  
--
```

```
CREATE TABLE IF NOT EXISTS `tickets` (  
  `id` int(11) NOT NULL auto_increment,  
  `account_id` int(11) NOT NULL,  
  `username` varchar(100) NOT NULL,  
  `date` timestamp NOT NULL,  
  `par_id` int(11) NOT NULL,  
  `theme` varchar(100) NOT NULL,  
  `msg` text NOT NULL,  
  `is_read` int(1) NOT NULL,  
  `is_closed` int(1) NOT NULL,  
  PRIMARY KEY (`id`)  
);
```

Текст програми

Розмір первинного програмного коду всіх модулів дуже великий, тому наводити його повністю недоцільно. Для прикладу, наведемо текст двох модулів: "class_connect.inc" (містить функції роботи з БД) та "services.php" (сервісні функції).

«class_connect.inc»

```
<?php
include("connectdata.inc");

global $mysqldata;
$link = mysql_pconnect($mysqldata['host'],
    $mysqldata['username'],
    $mysqldata['password']);
if(!$link){
    trigger_error("Ошибка при подключении к серверу MySQL.");
    exit();
}
if(!mysql_select_db($mysqldata['db'])){
    $error = mysql_error($link);
    $errno = mysql_errno($link);
    trigger_error("Ошибка при выборе БД". "(сообщение об ошибке:
$error [код ошибки: $errno])");
}

class classConnect{

    function query_array($query){
        $result = mysql_query($query);
        if (!$result){
            trigger_error("Ошибка при выполнении запроса, невозможно
заполнить массив");
        }
        if(mysql_num_rows($result) == 0){
            // echo "Данные отсутствуют!!!";
            return false;
        }

        $variable = array();
        while ($data = mysql_fetch_assoc($result)){
            $variable[] = $data;
        }
        mysql_free_result($result);
        return $variable;
    }

#####
    function get_($query){
        $q_arr= $this->query_array($query);
        return $q_arr;
    }
}
```

```

    }
#####
    function del_($query){
        mysql_query($query);
    }
#####
    function insert_($query){
        if(mysql_query($query)){
            // echo "<br>Данные внесены успешно!";
            return true;
        }
        else{
            echo "<br>Чёто не хо работать!";
            return false;
        }
    }
    function get_mysql_insert_id(){
        $last_id = mysql_insert_id();
        return $last_id;
    }
}

function insertRecord($table, $data){
    $i=0;
    $spt = '';
    foreach($data as $key => $row){
        $set .= $spt."`".$key."` = '".code($row)."'";
        $spt = ', ';
    }
    $q = "INSERT INTO `".$table."` SET ".$set;
    if(mysql_query($q)) return true;
    else return false;
}

function updateRecord($table, $data, $cond){
    $i=0;
    $spt = '';
    foreach($data as $key => $row){
        $set .= $spt."`".$key."` = '".code($row)."'";
        $spt = ', ';
    }

    $spt = '';
    foreach($cond as $key => $row){
        $where .= $spt."`".$key."`".$row." ";
        $spt = 'AND';
    }

    $q = "UPDATE `".$table."` SET ".$set." WHERE ".$where;
    if(mysql_query($q)) return true;
    else return false;
}

?>

```

«services.php»

```
<?php
$t = " Services ";

if(isset($_POST['subForm'])){
    $data = $_POST['data'];
    $data['cost'] = str_replace(',', '.', $data['cost']);
    if($_POST['id']==0){
        insertRecord('services', $data);
    }
    else{
        updateRecord('services', $data, array(
'id'=>'.'.intval($_POST['id']) ));
    }
    header("Location: ?act=".$act);
}
elseif(isset($_GET['del'])){
    mysql_query('DELETE FROM `services` WHERE `id`='.$id);
    header("Location: ?act=".$act);
}

$sql = "SELECT * FROM `services` ORDER BY `id`";

$services = $obj->get_($sql);

?>
<a href="javascript:void(0);"
onclick="javascript:getElementById('formAdd').style.display='block'
">&nbsp;Add service</a>
<br><br>

<div id="formAdd">
<form method="post" onsubmit="return checkform( this )">
    Наименование: * <input type="text" id="title"
name="data[title]" size="50" required><br>
    Стоимость ($/year): * <input type="text" id="cost"
name="data[cost]" size="45" required><br>
    <input type="hidden" id="servID" name="id" value="0">
    <div align="center"><input type="submit" name="subForm"
value="Send &raquo;"></div>
</form>
</div>

<table width="600" cellspacing="1" cellpadding="4" border="0">
    <tr class="tab_head">
        <td width="30" valign="middle"
align="center"><b>ID</b></td>
        <td align="center" valign="middle"><b>Наименование</b></td>
        <td width="100" valign="middle"
align="center"><b>Стоимость ($/year)</b></td>
```

```

        <td width="30" valign="middle" align="center"><b>Ред.</b></td>
        <td width="30" valign="middle" align="center"><b>Уд.</b></td>
    </tr>

    <?if( is_array($services) ){?>
        <?foreach($services as $row){
            $bg = ( ($i%2==0) ? '#cccccc' : '#dddddd' );
            $i++;
        ?>
        <tr bgcolor="<?=$bg?>" onmouseover="setPointer(this, 0,
'over', '<?=$bg?>', '#D3E983');" onmouseout="setPointer(this, 0,
'out', '<?=$bg?>', '#D3E983');" valign="middle">
            <td align="center" bgcolor="<?=$bg?>"
height="25"><?=$row['id']?></td>
            <td ><div
id="title<?=$row['id']?>"><?=$row['title']?></div></td>
            <td align="center"><div
id="cost<?=$row['id']?>"><?=$row['cost']?></div></td>
            <td align="center"><a class="edit"
href="javascript:void(0);"
onclick="javascript:serviceEdit('<?=$row['id']?>')"><br></a></td>
            <td align="center"><a onclick="javascript:return
isDel();" class="del"
href="?act=<?=$act?>&del&id=<?=$row['id']?>"><br></a></td>
        </tr>
        <?}?>
    <?}
    else {?>
        <tr><td colspan="4">Сервисов не обнаружено</td></tr>
    <?}?>
</table>

```

Програма і методика випробувань

Об'єкт випробувань.

Назва об'єкта випробувань: "login.php", надалі – "програма". Програма дозволяє користувачеві пройти аутентифікацію та отримати доступ до свого інтерфейсу.

Ціль випробувань.

Перевірка придатності програми до використання за призначенням, її відповідність заданим вимогам і програмним документам. Перевірка реально досягнутих характеристик програми.

Вимоги до програми.

Програма повинна задовольняти наступним функціональним характеристикам:

- ✓ перевірка правильності імені та паролю користувача;
- ✓ надання користувачеві доступу до свого інтерфейсу.

Користувач повинен вказати ім'я та пароль.

Засоби і порядок випробувань.

Вимоги до складу і параметрів технічних засобів:

- ✓ IBM-сумісний комп'ютер з мікропроцесором Intel® Core™ i3;
- ✓ оперативна пам'ять: 2048 Mb;
- ✓ жорсткий диск: HDD 300 Gb.

Вимоги до складу і параметрів програмних засобів :

- ✓ операційна система Free BSD 8.0;
- ✓ встановлене з'єднання з сервером баз даних MySQL 5.

Група людей, що будуть проводити випробування, повинна складатися з автора програми (розроблювача) та представника і/або самого замовника. При наявності відповідної можливості, ця група може бути розширена програмістом, спеціалістом з тестування і фахівцем-проектувальником, що не брали участь у розробці програми.

Попередньо (не пізніше 3 днів) до початку випробувань уточнюється кінцевий склад групи, що буде проводити випробування. Кожен член групи

одержує необхідну документацію до програми (список вимог до програми, опис програми, керівництво користувача, програму і методику випробувань).

Порядок проведення випробувань:

✓ послідовно виконуються контрольні тести і результати заносяться у відповідний журнал;

✓ здійснюється аналіз отриманих результатів і встановлюється відповідність програми заданим вимогам і програмним документам.

Методи випробувань.

Робота програми залежить тільки від двох параметрів, що вводяться користувачем: від імені та паролю користувача.

Іспити будуть проводитися методом "чорної шухляди".

Тести будуть створюватися на основі класів еквівалентності.

Класи еквівалентності виділяються шляхом вибору кожної вхідної умови і розбивкою її на дві або більш груп. Розрізняють два типи класів еквівалентності: правильні класи еквівалентності, що представляють правильні вхідні дані програми і неправильні класи еквівалентності, що представляють всі інші можливі стани умов (помилкові вхідні значення). При цьому існує ряд правил:

✓ один тест може покривати тільки один неправильний клас еквівалентності;

✓ один тест повинний покривати як можна більше число непокритих правильних класів еквівалентності.

Виділені класи еквівалентності для складання тестів мають унікальні номери і приводяться в таблиці Г.1.

Таблиця Г.1 – Класи еквівалентності

Операція, що тестується	Вхідні умови	Правильні класи еквівалентності	Неправильні класи еквівалентності
Перевірка імені та паролю користувача	Ім'я правильне	Так (1)	Ні (2)
	Пароль правильний	Так (3)	Ні (4)
Надання користувачеві доступу до свого інтерфейсу	Належність користувача до інтерфейсу встановлена	Так (5)	Ні (6)

Тести.

Тести складені на основі класів еквівалентності таблиці Г.1. Тести використовують наступні початкові умови:

1. Користувач – serge, його пароль – sf3333db.
2. Користувач – admin, його пароль – KJwedfw3.

Тест 1.

Класи еквівалентності: 1, 3, 5.

Вхідні дані: Ім'я користувача: serge. Пароль користувача: sf3333db.

Результат: Ім'я та пароль користувача розпізнані. Наданий доступ до інтерфейсу користувача.

Тест 2

Класи еквівалентності: 2, 3, 5.

Вхідні дані: Ім'я користувача: andry. Пароль користувача: pass.

Результати: Ім'я та пароль користувача не пізнані. Належність до користувачів системи не встановлена. Видане повідомлення про помилку введення імені чи паролю. Доступу до системи не надано.

Тест 3.

Класи еквівалентності: 1, 4, 5.

Вхідні дані: Ім'я користувача: user1. Пароль користувача: pass2.

Результати: Ім'я та пароль користувача не пізнані. Належність до користувачів системи не встановлена. Видане повідомлення про помилку введення імені чи паролю. Доступу до системи не надано.

Тест 4.

Класи еквівалентності: 1, 3, 5.

Вхідні дані: Ім'я користувача: admin. Пароль користувача: KJwedfw3.

Результати: Ім'я та пароль користувача впізнані. Наданий доступ до адміністрування системи.

Керівництво користувача для покупця

Вступ.

Метою даної роботи є розширення способів здійснення електронних платежів та забезпечення режиму реального часу роботи клієнта з замовленнями у інформаційній системі хостингової компанії "MyHost".

Для користування системою достатньо рівня користувача ПК.

Призначення й умови застосування.

Інформаційна система хостингової компанії призначена для:

- ✓ створення пакетів послуг, що надає компанія;
- ✓ формування замовлення;
- ✓ реєстрації та обліку клієнтів;
- ✓ взаємодії з платіжними системами;
- ✓ передачі повідомлень всередині системи.
- ✓ автоматичного створення клієнтського аккаунту на сервері.

Користувачі повинні працювати на віддалених персональних комп'ютерах мережі Інтернет. Для комфортної роботи з системою, комп'ютер користувача повинен мати відповідне технічне і програмне забезпечення.

Вимоги до програмного забезпечення комп'ютера користувача:

- ✓ будь-яка операційна система з можливістю установки віддаленого мережного з'єднання і графічним інтерфейсом;
- ✓ будь-який web-оглядач з підтримкою HTML, JavaScript.

Вимоги до технічного забезпечення комп'ютера користувача:

- ✓ мікропроцесор класу Pentium 1000 і вище;
- ✓ оперативна пам'ять: RAM 256 Мб і більше;
- ✓ жорсткий диск: 20 Гб і більше;
- ✓ мережна карта.

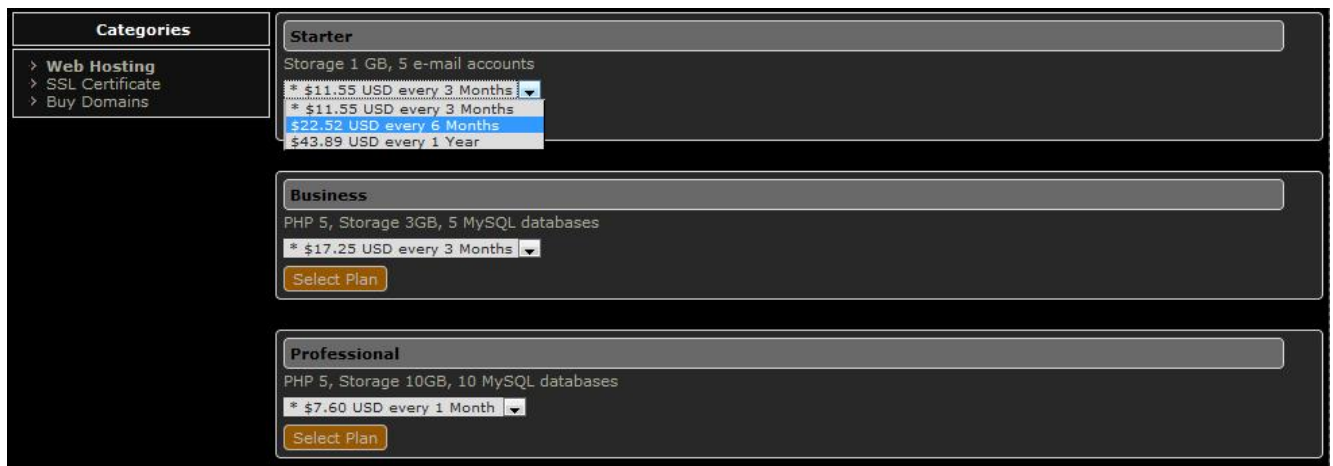
Підготовка до роботи.

Клієнтська машина повинна мати відповідні програмне і технічне забезпечення, описані вище. Користувач повинний мати можливість установити з'єднання з web-сервером системи по протоколу http.

Для початку роботи із системою, користувач повинен запустити web-оглядач і ввести адресу web-сторінки з посиланням на інформаційну систему. Після натискання на посилання, відкриється сторінка з запропонуванням зареєструватися у системі, потім (якщо користувач вже зареєструвався і був успішно аутентифікован) користувач може працювати із системою. Якщо користувач вже раніше був зареєстрований у системі, то він повинен натиснути на посилання з запропонуванням входу до системи для зареєстрованих користувачів, пройти успішно аутентифікацію і працювати в своєму інтерфейсі системи.

Опис операцій.

Спочатку користувач потрапляє на сторінку, де обирає необхідний йому пакет (рисунок Д1).



Categories	Starter
> Web Hosting	Storage 1 GB, 5 e-mail accounts
> SSL Certificate	* \$11.55 USD every 3 Months
> Buy Domains	* \$11.55 USD every 3 Months
	\$22.52 USD every 6 Months
	\$43.89 USD every 1 Year

Business
PHP 5, Storage 3GB, 5 MySQL databases
* \$17.25 USD every 3 Months
Select Plan

Professional
PHP 5, Storage 10GB, 10 MySQL databases
* \$7.60 USD every 1 Month
Select Plan

Рисунок Д1 – Сторінка "Вибір пакету"

Реєстрація користувача у системі відбувається після заповнення ним необхідних даних на сторінці, яка відкривається на екрані його монітора після натискання ним кнопки для придбання якого-небудь пакету у системі. Форма для цієї реєстрації зображена на рисунку Д2.

Після успішної реєстрації, користувач потрапляє на сторінку оплати (рисунок Д3).

New Clients - Fill In All Fields Below

First Name:

Last Name:

Organization:

Optional

Address:

City:

Enter County:

Phone:

E-mail Address:

Password:

Verify Password:

Order Notes:

Terms Of Service Agreement:

I do not agree to the TOS.

View Agreement

Click to checkout now

Рисунок Д2 – Сторінка "Реєстрація"

2CO

2CHECKOUT.COM

(For non-US currency, payment op

Shopping with us is safe.

Guaranteed

Verified by VISA

learn more

MasterCard SecureCode.

learn more

BBBOnline RELIABILITY PROGRAM

Review Order

myDomainKeeper has directed you to 2Checkout.com to complete your purchase.

Qty	Vendor Product ID Name	Price € (EUR)	Subtotal € (EUR)
1	Starter -	Cart purchase: Starter - 33.66	33.66
	Vendor site cart purchase: Cart id Starter - (Qty: 1) Generic		

By completing my purchase, I agree to the [terms of use](#) and [privacy statement](#).

Check out with PayPal

The safer, easier way to pay

OR

CONTINUE CHECKOUT

Safe and Secure Shopping

Рисунок Д3 Сторінка платіжної системи оплати

Аварійні ситуації.

Якщо система працює не коректно, або зовсім не працює, необхідно перевірити з'єднання з Інтернет. Якщо з'єднання існує, то необхідно звернутися по допомогу до адміністраторів системи.

Рекомендації щодо освоєння.

Інформація, що наведена в даному документі, достатня для освоєння використання системи покупцями. При виникненні додаткових питань вони можуть звертатися до адміністратора системи.