

МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ  
НАЦІОНАЛЬНИЙ УНІВЕРСИТЕТ КОРАБЛЕБУДУВАННЯ  
ІМЕНІ АДМІРАЛА МАКАРОВА

Навчально-науковий інститут комп'ютерних наук та управління проектами  
Кафедра інформаційних управляючих систем та технологій

"Допущений до захисту"  
Завідувач кафедри ІУСТ

  
к.т.н, доц. Михелєв І.Л.

"\_\_" \_\_\_\_ 2024 р.

**КВАЛІФІКАЦІЙНА РОБОТА**

на здобуття ступеня вищої освіти "магістр"  
(освітньо-кваліфікаційний рівень)

на тему **Системний аналіз технологій проектування розподілених систем та розробка інформаційної системи управління завданнями на ІТ-підприємстві**

Виконав: студент 6 курсу, групи 6145м  
напряму підготовки (спеціальності)  
124 – "Системний аналіз".

(шифр і назва напряму підготовки, спеціальності)

  
Калугіна К.Д.

(прізвище та ініціали)

Керівник Ажищев В.Ф. 

(прізвище та ініціали)

м. Миколаїв – 2024 р.

**МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ**  
**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 124 – "Системний аналіз"

Освітня програма "Системний аналіз"

**"ЗАТВЕРДЖУЮ"**

Гарант освітньої програми

д.ф-м.н, проф. Ушкац М.В.

" \_\_\_\_ " \_\_\_\_\_ 2024 р.

**ЗАВДАННЯ**  
**НА КВАЛІФІКАЦІЙНУ РОБОТУ**  
на здобуття ступеня вищої освіти "магістр"

Студенту Калугіна Катерина Дмитрівна

1. Тема проекту (роботи) Системний аналіз технологій проектування розподілених систем та розробка інформаційної системи управління завданнями на ІТ-підприємстві  
керівник проекту (роботи) Ажищев Віктор Федорович,  
затвержені наказом вищого навчального закладу від " 08 " 10 2024 року № 1055-уч.

2. Строк подання студентом проекту (роботи) 15 грудня 2024 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, англійською), зміст, перелік умовних означень, символів, одиниць та термінів (при необхідності).

- Вступ.

- Аналіз предметної області і постановка задачі.

- Розробка проекту автоматизованої системи.

- Реалізація проекту автоматизованої системи (технічна документація по проекту).

- Розділ з охорони праці.

- Висновки.
- Список використаної літератури.
- Додатки: загальний опис підсистеми, керівництво користувача, керівництво адміністратора, програма і методика випробувань, графічні інтерфейси, код програми.

#### 5. Перелік графічного матеріалу

- 5.1 Організаційна структура підприємства.
- 5.2 Діаграма варіантів використання.
- 5.3 Варіанти концепції системи.
- 5.4 Розміщення комплексу технічних засобів.
- 5.5 Логічна модель бази даних.
- 5.6 Діаграма класів системи.
- 5.7. Екранні форми.

#### 6. Консультанти розділів проекту (роботи)


Розділ	Прізвище, ініціали та посада консультанта	Підпис, дата	
		завдання видав	завдання прийняв
1.	Ажищев В.Ф.	05.09.2023	22.12.2023
2.	Ажищев В.Ф.	23.12.2023	15.06.2024
3.	Ажищев В.Ф.	15.06.2024	25.09.2024
4.	Ажищев В.Ф.	25.09.2024	14.11.2024
5.	Гурець Н.В.	25.09.2024	14.11.2024
6.	Гурець Н.В.	25.09.2024	14.11.2024

7. Дата видачі завдання 05 вересня 2024 р.


### КАЛЕНДАРНИЙ ПЛАН

№ з/п	Назва етапів дипломного проекту (роботи)	Строк виконання етапів проекту (роботи)	Примітка
1.	Вивчення і опис предметної області	22.12.2024	
2.	Розробка концепції ІС	15.03.2024	
3.	Розробка проектних рішень по ІС	15.06.2024	
4.	Реалізація проекту, розробка робочої документації	25.09.2024	
5.	Вирішення питань з охорони праці	14.11.2024	
6.	Виконання графічної частини роботи	14.11.2024	
7.	Оформлення пояснювальної записки	14.11.2024	
8.	Подання роботи на кафедральний захист	16.11.2024	
9.	Подання роботи рецензенту	20.11.2024	
10.	Подання роботи на державний захист	15.12.2024	

Студент

 Калугіна К.Д.

Керівник проекту (роботи)

 к.т.н., доцент Ажищев В.Ф.

## АНОТАЦІЯ

Дана магістерська робота присвячена системному аналізу технологій проектування розподілених систем та розробці проекту інформаційної системи управління завданнями на підприємстві JGSoft з метою підвищення продуктивності працівників і збереження їхнього часу.

Під час роботи формуються вимоги до системи, розроблюються концепції системи і проводиться вибір найкращої, розробляється технічне завдання. Далі проводиться розробка основних проектних рішень за технічним, інформаційним та програмним забезпеченням, формується робоча документація за програмним забезпеченням.

Дана дипломна робота викладена на 106 сторінках друкованого тексту, містить 13 рисунків, 15 таблиць та має 3 додатки які містять технічне завдання, текст виконаної програми, інструкцію користувача. Список літератури з 11 найменувань.

Робота виконана на українській мові.

**Ключові слова:** системний аналіз; проектування розподілених систем; інформаційні системи; розподілені системи; архітектура систем; бази даних; автоматизація бізнес-процесів; інтерфейси.

## ABSTRACT

This master's work is devoted to the study of distributed systems design technologies and the development of a project of an information management system for tasks at JGSoft to improve the productivity of workers and save their working time.

In the course of work, the system requirements are formed, the system concepts are developed and the best one is selected, the technical task is developed. Further, the development of basic design solutions for technical, informational and software is carried out, working documentation on software is formed.

This thesis is presented on 106 pages of printed text, contains 13 figures, 15 tables and has 3 appendices containing the technical task, the text of the executed program, the instruction. References of 11 items.

The work is done in Ukrainian.

**Keywords:** systems analysis; distributed systems design; information systems; distributed systems; systems architecture; databases; business process automation; interfaces.

# ЗМІСТ

<b>Вступ</b> .....	
<b>1. Дослідження предметної області, формування вимог до інформаційної системи, розробка її концепції і постановка задачі</b> .....	
1.1 Опис предметної області .....	
1.2 Формування вимог до інформаційної системи .....	
1.3 Розробка концепції інформаційної системи .....	
1.4 Техніко-економічне обґрунтування розробки інформаційної системи на основі функціонально-вартісного аналізу .....	
<b>2. Розробка проектних рішень по системі</b> .....	
2.1 Загальносистемні проектні рішення.....	
2.2 Рішення по організаційному забезпеченню .....	
2.3 Рішення по інформаційному забезпеченню .....	
2.4 Рішення по технічному забезпеченню .....	
2.5 Опис алгоритмів функціонування системи .....	
2.6 Рішення по програмному забезпеченню .....	
<b>3. Розробка робочої документації інформаційної системи управління завданнями на підприємстві JGSoft</b> .....	
<b>4. Розрахунок економічної ефективності впровадження інформаційної системи управління завданнями</b> .....	
4.1 Вступ.....	
4.2. Розрахунок витрат на створення та експлуатацію ПЗ інформаційної системи .....	
4.3 Економічна ефективність розробки і впровадження системи .....	
<b>5. Охорона праці</b> .....	
5.1 Вступ.....	
5.2 Аналіз небезпечних і шкідливих факторів у відділі інтерфейсного програмування підприємства JGSoft.....	
5.3 Розрахунок системи штучного освітлення приміщення відділу інтерфейсного програмування .....	
5.4 Розробка заходів щодо зменшення впливу шкідливих факторів на підприємстві JGSoft .....	
<b>6. Охорона навколишнього середовища</b> .....	
6.1 Вступ.....	
6.2 Забруднення атмосферного повітря автотранспортом.....	
6.3 Заходи запобігання забруднення навколишнього середовища .....	
6.4 Висновки .....	
<b>Висновки</b> .....	

**Список літератури**.....

**Додаток А.** – Загальний опис системи управління завданнями на підприємстві JGSoft.....

**Додаток Б.** – Інструкція користувача автоматизованої системи управління завданнями.....

**Додаток В.** – Тексти програмних модулів системи.....

## ВСТУП

Впровадження сучасних інформаційних технологій дозволяє підвищити ефективність функціонування підприємств.

Завданням дипломної роботи являється розробка проекту інформаційної системи управління завданнями, яка призначена для автоматизації роботи працівників підприємства на будь-якому рівні, від начальника відділу до працівника і не тільки. Функції, що автоматизуються системою, пов'язані з організацією робочого часу працівників отримання завдань та іншого документообігу, як то звіт та статистичні показники працівників.

Цілями розробки автоматизованої системи є:

- підвищення ефективності управління підприємством за рахунок оптимізації обігу інформації;

- зменшення ризику помилок при прийнятті рішень через невірогідність інформації про функціонування підприємства;

- зменшення витрат на документообіг.

Для досягнення поставленої мети були вирішені наступні основні задачі:

- проведений аналіз документообігу підрозділів підприємства;

- проведений функціонально-вартісний аналіз, і обрана концепція майбутньої системи;

- досліджений економічний ефект від розробки програмного забезпечення;

- проведені дослідження з екології та охорони праці.

# 1. Дослідження предметної області, формування вимог до інформаційної системи, розробка її концепції і постановка задачі

## 1.1 Опис предметної області.

Інформаційна система управління завданнями призначена для автоматизації роботи працівників підприємства на будь-якому рівні, від начальника відділу до працівника і не тільки. Функції, що автоматизуються системою, пов'язані з організацією робочого часу працівників отримання завдань та іншого документообігу, як то звіт та статистичні показники працівників.

Інформаційна система (ІС) є середовищем, складовими елементами якого є комп'ютери, об'єднані в комп'ютерні мережі, програмні продукти, бази даних, люди, різного роду технічні та програмні засоби зв'язку і т.п. Основна мета інформаційної системи – організація збереження, обробки і передачі інформації. Інформаційна система являє собою людино-комп'ютерне оточення (систему) для обробки інформації.

ІС, що розробляється, є розподіленою, тобто для її функціонування, крім програм-клієнтів, необхідна наявність принаймні одного сервера програм. Програмне забезпечення сучасних ІС має наступні логічні рівні (дивись рисунок 1.1):

- Рівень користувача (інтерфейс і сервіси користувача).
- Прикладний рівень (прикладні сервіси).
- Рівень даних (сервіси доступу до даних і сховище даних).

Опишемо функції кожного рівня.

– Рівень користувача: відображає дані і дозволяє користувачу редагувати їх. Існує два основних типи інтерфейсу: віконний (реалізується засобами підсистеми інтерфейсу користувача операційної системи) і на основі Web. Web-інтерфейси засновані на HTML, DHTML, XHTML чи XML, у результаті чого вони можуть відображатися будь-яким Web-оглядачем на будь-якій платформі.

– Прикладний рівень: тут реалізовані бізнес-правила й обмеження на дані (перевірка коректності даних). І хоча його сервіси використовуються презентаційним рівнем, він не прив'язаний до якого-небудь клієнта – сервіси

прикладного рівня доступні будь-якому клієнту. Бізнес-правила виражаються у формі прикладних алгоритмів, корпоративних правил і т.д. Обмеження на дані гарантують точність і цілісність інформації, що зберігається. Бізнес-правила звичайно реалізуються окремим модулем на централізованому сервері, що дає можливість доступу до нього відразу декільком клієнтам.

– Рівень даних: прикладний рівень не має інформації про те, як і де зберігаються дані, які він обробляє. У цьому питанні він покладається на сервіси доступу до даних, що виконують всю роботу з отримання і передачі даних. Сервіси доступу до даних також реалізуються у виді ізольованих модулів, які знають про місце збереження інформації. Кожен модуль доступу до даних, як правило, відповідає і за цілісність сховища (наприклад, реляційної бази даних). Для багаторівневих програм у якості сховища інформації підходять сервера баз даних (MS SQL Server, Oracle, Sybase, MySQL, PostgreSQL і ін.), необхідні для обслуговування даних у таблицях і високопродуктивної вибірки інформації.

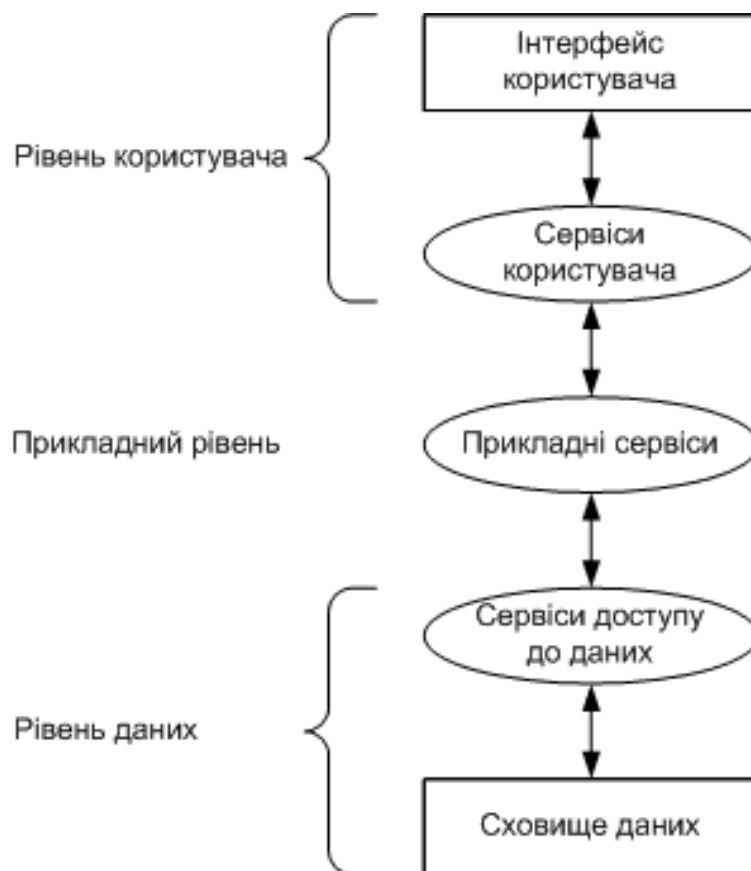


Рисунок 1.1 – Логічна архітектура програмного забезпечення інформаційної системи

Реалізація функцій інформаційної системи неможлива без знання орієнтованих на неї інформаційних технологій. Нижче представлений опис технологій, які використовуються на кожному рівні інформаційної системи, що розробляється.

### **1.1.1 Технології рівня користувача.**

#### *Web-оглядач і Web-інтерфейс.*

Web-оглядач чи Web-броузер – це програма, що формує стандартний запит для Web-сервера відповідно до правил протоколу HTTP (Hyper Text Transfer Protocol), одержує документи від Web-сервера, інтерпретує і відображає їх на екрані користувача.

Фактично Web-оглядач є єдиною програмою, яка необхідна користувачу для роботи з інформаційною системою; він надає відповідний Web-інтерфейс.

Web-інтерфейс широко розповсюджено і пропонує готові засоби відображення. Web-оглядачі, що вільно розповсюджуються, мають прості інтерфейси, що збільшує ефективність розгортання програм.

Найпоширеніші Web-оглядачі: MS Internet Explorer, Safari, Mozilla FireFox, Opera.

#### *HTML.*

Найпоширеніший і простий спосіб створення Web-інтерфейсу – використання HTML (HyperText Markup Language, Мова розмітки гіпертексту). Web-інтерфейс складається з набору HTML-документів, статичних чи генерованих динамічно. HTML являє собою набір команд, що називаються тегами, які розташовуються між фрагментами тексту й усередині них. Команди HTML дозволяють структурувати документ, виділяючи в ньому частини тексту (заголовки різних рівнів, абзаци, перерахування і т.д.), які логічно розрізняються. У результаті кожен Web-оглядач може формувати текст документа таким чином, щоб найкращим способом відобразити його на конкретному дисплеї. Для надання документам більшої виразності текст звичайно форматується з використанням збільшених розмірів шрифту

заголовків, застосуванням напівжирного і курсивного накреслень для важливих термінів, виділенням пунктів списків і т.д. Мова HTML дозволяє також включати в документи ілюстративну графіку, що може бути відображена програмами перегляду, що основані на використанні графічного інтерфейсу користувача

Однією з найважливіших властивостей HTML є можливість включення в документ гіпертекстових посилань. Ці посилання дозволяють користувачу завантажити новий документ на свій комп'ютер, просто клацнувши вказівником миші в тім місці екрана, де розташоване посилання. Будь-який документ може містити посилання на інші документи. Областю документа, що використовується як посилання, може служити слово, група слів, графічне зображення чи навіть заданий фрагмент зображення.

Іншою не менш важливою властивістю HTML є форми. HTML-форми дають розробнику Web-інтерфейсу можливість змішування традиційних елементів, властивих друкованим сторінкам, з інтерфейсом користувача сучасної віконної системи. Механізм форм дозволяє розміщати серед форматowanego тексту з ілюстраціями кнопки, меню, що розкриваються, текстові поля й інше. Ці елементи інтерфейсу дають користувачам, що переглядають сторінки, можливість передачі відгуку, перетворюючи їх із простих глядачів в учасників, що мають право голосу. Головне обмеження HTML-форм пов'язано з тим, що Web-оглядачі змушені здійснювати нове з'єднання з сервером при кожному отриманні HTML-сторінки чи передачі даних форми.

HTML сам по собі не дозволяє створювати інтерактивні документи, тобто такий Web-інтерфейс є статичним. Існує ще одна технологія (DHTML), що дозволяє Web-сторінкам бути більш динамічними. DHTML (Dynamic HTML, динамічний HTML) дозволяє змінювати вид сторінки Web і її стиль після того, як документ завантажився.

Ключовим поняттям DHTML є шари. У звичайному HTML-документі, всі елементи так чи інакше впливають один на одного при формуванні його зовнішнього вигляду. Тобто абзац тексту "зміщує" наступні елементи на деяку

відстань униз, картинка, вирівняна вправо, "змушує" текст обтікати її і так далі. Всі елементи разом утворюють так званий HTML-потік чи потік документа (document flow). Шари у свою чергу – це частини сторінки, що не входять в основний HTML-потік. Для кожного з них створюється свій потік, що ніяк не впливає на основний. При цьому вміст шарів відображається, як правило, над (зверху) основним змістом сторінки. Шари можна пересувати і робити невидимими. Таким чином, за допомогою шарів можна створювати різні інтерактивні ефекти (наприклад, деревоподібне меню як у провіднику Windows).

Одне з обмежень DHTML полягає в тому, що сторінки, створені з застосуванням технології DHTML, можна переглядати тільки в четвертій і більш пізніх версіях Web-оглядачів MS Internet Explorer і Netscape Navigator.

DHTML заснована на технологіях CSS, JavaScript і HTML.

CSS.

CSS (Cascading Style Sheets, каскадні таблиці стилів) – ще один стандарт, розроблений W3C. Існують CSS Level 1 і CSS Level 2. Далі мова йде про CSS2.

CSS визначив правила опису стилів візуального відображення елементів HTML-документів. Основні особливості CSS:

– CSS – це мова, що дозволяє приєднувати стилі до будь-яких структурованих документів. На сьогодні такими є HTML-документи і XML-документи.

– CSS поширив поняття стилю відображення на друкувальні пристрої, синтезатори мови й інші пристрої відображення документів.

Ціль створення каскадних таблиць стилів полягала в тому, щоб відокремити структуру документа (описану мовою HTML чи XML) від правил його відображення на різних пристроях (що задаються таблицями CSS). За допомогою стилів можна визначити властивості елементів (текст, гіпертекстове посилання, абзац, заголовок, рисунок, таблиця й інші) у HTML-документі.

Існує три способи завдання стилів у HTML-документі: зовнішні таблиці стилів, внутрішні таблиці стилів і таблиці стилів елементів.

## *JavaScript*

Мова JavaScript – інтерпретуєма об’єктно-базована мова високого рівня, що дозволяє писати програми, які виконуються в середовищі Web-оглядача. В даний час JavaScript є найбільш широко розповсюдженою мовою в Web для написання сценаріїв, що виконуються на стороні клієнта.

JavaScript дозволяє вбудовувати команди в HTML-документи. Команди JavaScript можуть виконуватися при завантаженні документа, при заповненні полів форм, при наведенні вказівника миші на елемент документа й інше. Мова JavaScript має досить широкі можливості. Вона не дозволяє працювати на рівні машинних кодів, однак дозволяє отримати доступ до багатьох можливостей Web-оглядачів, HTML-документів, а іноді й операційної системи, у якій працює Web-оглядач. На відміну від Java чи C, програми на JavaScript не потрібно компілювати, а оглядачу не потрібно завантажувати віртуальну машину для виконання програмного коду.

Одна з причин, згідно якої Web-розробники використовують JavaScript – можливість виконання на стороні клієнта багатьох функцій, що раніше виконувалися винятково на стороні сервера. Кращим прикладом є перевірка HTML-форм. За допомогою JavaScript елементи форми можна перевірити до того, як користувач передасть інформацію Web-серверу. Це приводить до зменшення обсягу транзакцій HTTP.

JavaScript широко використовується для створення DHTML-документів. JavaScript дозволяє переміщатися по дереву (ієрархії) елементів HTML-документа, змінювати його властивості і багато чого іншого.

Крім JavaScript існує ще одна мова для вирішення аналогічних задач – VBScript. Однак вона, на відміну від JavaScript, не отримала такого ж широкого поширення.

### 1.1.2 Технології прикладного рівня.

#### *Web-сервер.*

Web-сервер (HTTP-сервер) – програма, що сприймає запити клієнтів (наприклад, Web-оглядачів), які приходять, як правило, на стандартний порт TCP/IP (порт 80), і виконує в найпростішому випадку пересилання HTML-документів клієнту. Web-сервер є програмою, яка дозволяє службі Internet, що називається WWW (World Wide Web, Всесвітня інформаційна мережа), знаходити і використовувати гіпертекстові і графічні документи на різних серверах.

Фактично Web-сервер передає інформацію користувачу, який знаходиться на віддаленому комп'ютері. Користувач переглядає інформацію за допомогою Web-оглядача (MS Internet Explorer, Netscape Navigator і інші) і відсилає нові запити Web-серверу.

– Web-сервер як правило має вбудовану систему аутентифікації по логіну та пароллю, яку можна використовувати для обмеження доступу до ресурсів сервера.

– Найбільш розповсюджені Web-сервера: Apache Server, MS IIS, Netscape Enterprise Server.

#### *CGI.*

CGI (Common Gateway Interface) – це набір правил, згідно з якими програми на стороні сервера можуть через Web-сервер пересилати дані клієнтам. CGI є перевіреним і універсальним способом створення розподілених систем у Internet. Головна задача, яку він вирішує, – це забезпечення викликів віддалених процедур між клієнтом і сервером з використанням сокетів TCP/IP.

Крім CGI існують ще кілька подібних технологій – ISAPI/NSAPI, Servlet і JSP. Ці чотири технології призначені для вирішення приблизно тих самих задач і концептуально дуже схожі. Всі вони дозволяють розширити можливості Web-серверів шляхом розробки програм, зареєстрованих на цьому сервері.

#### *Servlet і JSP.*

Технологія Servlet була розроблена компанією Sun Microsystems, і була

призначена для того щоб серверна частина отримуючи запит від клієнта у якості відповіді повернула сгенеровану HTML сторінку яка і відображає дані. Servlet – це спеціальний Java клас, який реалізує необхідні API для отримання запитів від WEB клієнта та після обробки надання результату. Сам по собі Servlet не чого робити не вмє він є частиною системи яка знаходиться у контейнері (Java web сервер) який вмє отримати запит і передати його необхідному додатку. Технологія JSP (JSPx) є "надбудовою" над сервлетами і призначена для більш наглядного конструювання інтерфейсу. JSP є наслідником Servlet і являє собою комбінацію Java коду та HTML, на зразок PHP.

### *Java web сервери.*

Існує декілька Java web серверів зокрема такі, JBoss, Tomcat, Glassfish та ін. Майже усі вони виконують однакові задачі:

- - компіляція JSP сторінок;
- - зв'язок з базою даних;
- - обмін даними між програмою та клієнтом;
- - обмін даними між програмою та БД.

### **1.1.3 Технології рівня даних.**

#### *Мова SQL.*

SQL – це скорочена назва структурованої мови запитів (Structured Query Language). SQL – це мова програмування, яка використовується для виконання запитів і модифікації даних, а також для управління СУБД (системами управління базами даних). SQL працює тільки з базами даних одного певного типу, які називаються реляційними.

Стандарти SQL затвержені ANSI і ISO. Хоча в кожній СУБД використовується власний діалект SQL, більшість цих діалектів задовольняють стандарту ANSI SQL. Таким чином, для сумісності програмного забезпечення з різними СУБД, необхідно строго дотримуватися стандарту ANSI SQL.

Мова SQL містить оператори, які підрозділяються на дві категорії: мова опису даних (Data Definition Language, DDL) і мова маніпулювання даними

(Data Manipulation Language, DML). DDL використовується для створення і управління такими об'єктами, як бази даних, таблиці і представлення. Оператори DDL містять команди CREATE, ALTER і DROP для кожного з об'єктів. DML використовується для управління даними в об'єктах баз даних за допомогою таких операторів, як SELECT, INSERT, UPDATE і DELETE.

#### *Сервер баз даних.*

Сервер баз даних – це окрема програма, яка виконується як окремий процес (сервіс). Сервер передає обрану з бази інформацію по каналу клієнту. Саме сервер працює з даними, піклується про їхнє розміщення на диску.

Найбільш розповсюджені сервера баз даних: MS SQL Server, Oracle, Sybase, MySQL, PostgreSQL.

#### *EJB.*

EJB – ще одна технологія від Sun яка призначена для реалізації моделі у Java програмі. Вона реалізує зв'язок з БД не залежно від реалізації, та представлення у вигляді ява об'єктів. EJB поділяється на дві складові:

- EntityBeans (класи які є об'єктним представленням інформацією з БД);
- SessionBeans (класи та інтерфейси які контролюють, змінюють, і тд EntityBeans).

#### **1.1.4 Динаміка роботи інформаційної системи.**

Маючи загальне уявлення про технології, які використовуються для розробки системи, можна описати динаміку роботи системи.

- Користувач запускає Web-оглядач на віддаленому комп'ютері-клієнті і вводить адресу Web-сервера інформаційної системи.
- Web-оглядач надсилає запит Web-серверу.
- Web-серверу надсилає запит Java-серверу.
- Java-сервер передає запит відповідному додатку (системі).
- Система отримав запит і передає його відповідному сервлету (JSP).
- Servlet (JSP) обробив запит звертається (якщо необхідно) до SessionBean.
- SessionBean звертається до Java-сервера з проханням надати зв'язок з

необхідним сервером БД та проводить аутентифікацію користувача по логіну і паролю та вибирає потрібну базу.

- Отримав зв'язок SessionBean проводить необхідний запит і отримав результати повертає їх контролеру (Servlet або JSP).

- Контролер отримав результат генерує необхідну HTML сторінку та повертає її і передає її Web-серверу.

- Web-сервер додає HTTP-заголовок до отриманого документа і відправляє його Web-оглядачу.

- Web-оглядач приймає HTML-документ і відображає його на екрані користувача.

- Користувач отримує Web-інтерфейс і необхідну інформацію. Він може надіслати новий запит Web-серверу.

Таким чином, вимальовується схема взаємодії технологій (рисунок 1.2)

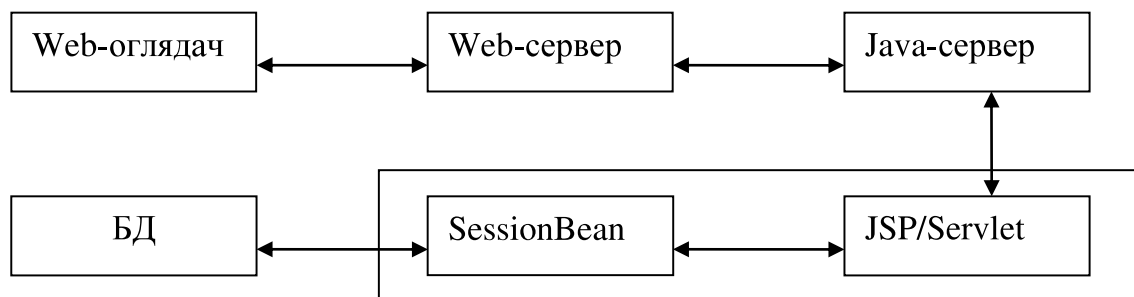


Рисунок 1.2 – Схема взаємодії технологій інформаційної системи

## 1.2 Формування вимог до інформаційної системи.

### 1.2.1 Характеристика об'єкта автоматизації.

Об'єктом авторизації є підприємство JGSoft, пов'язане з розробкою програмного забезпечення. Підприємство має потребу у створенні єдиної бази даних співробітників інформації про них, а також системи управління внутрішнього документообігу, створення та отримання завдання його перенаправлення та отримання інформації про нього, також система яка проводить статистичний аналіз.

### **1.2.2 Цілі, критерії й обмеження створення інформаційної системи.**

Метою розробки інформаційної системи є збільшення продуктивності і поліпшення умов праці працівників шляхом автоматизації функцій, які вони виконують.

Основні критерії створення інформаційної системи:

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

– Економічний критерій. Автоматизована інформаційна система створюється для підприємства безкоштовно. Вартість інформаційної системи буде складатися з витрат на придбання необхідного програмного, технічного й іншого видів забезпечення.

– Критерій мобільності. Користувач повинен мати можливість працювати із системою як з комп'ютера локальної мережі, так і з будь-якого комп'ютера, що має доступ до Internet.

– Критерій безпеки. Система повинна бути безпечною з погляду збереження й обробки даних. Втрата даних може привести до серйозних збоїв у роботі системи. Необхідний захист системи від несанкціонованого втручання.

### **1.2.3 Функції та задачі інформаційної системи.**

Задачею інформаційної системи, що створюється, є автоматизація наступних функцій викладача:

– Облік відділків та їх працівників.

– Облік завдань та інформацію про їхній статус.

– Формування звітів.

– Усі інші вимоги не є головними і можуть змінюватися під час розробка або експлуатації.

Для створення інформаційної системи необхідно розробити єдину базу даних для збереження інформації про відділи та їх співробітників, про отриманні ними завдання і іншу. Також повинно бути передбачене програмне

забезпечення для адміністрування системи.

#### **1.2.4 Висновки і пропозиції.**

Таким чином, необхідно розробити автоматизовану інформаційну систему, що дозволить підвищити ефективність та якість роботи працівників, поліпшити умови їхньої праці. Система є глобальною інформаційною системою і вирішує задачі, що пов'язані з трудовою діяльністю. Інформаційна система повинна включати базу даних, програмне забезпечення працівника та програмне забезпечення адміністратора, а також відповідне організаційне, технічне та інші види забезпечення.

У перспективі, база даних інформаційної системи може використовуватися різними підрозділами підприємства. Начальник відділу або менеджер може одержувати коректну і своєчасну інформацію про стан кожного завдання та інформацію про співробітників.

### **1.3 Розробка концепції інформаційної системи.**

#### **1.3.1 Опис результатів вивчення об'єкта автоматизації.**

Інформаційна система розробляється як система управління робітниками та завданнями на підприємстві JGSoft. Для створення системи потрібна розробка бази даних інтерфейсу користувача та адміністратора а також контролеру управління внутрішнього документообігу, створення та отримання завдання його перенаправлення та отримання інформації про нього, також створення статистичного аналізу.

#### **1.3.2 Аналіз вимог до інформаційної системи.**

Крім функціональних вимог до інформаційної системи (облік успішності студентів, облік відвідування студентів, організація розкладу викладача), необхідно врахувати вимоги до системи в цілому:

- Простий і зручний інтерфейс користувача.
- Доступ до системи через Internet з будь-якого комп'ютера глобальної

мережі.

- Багатокористувальна робота.
- Висока масштабуємість системи.
- Висока продуктивність.
- Надійність.
- Конфіденційність інформації.

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

- Web-інтерфейс.
- Багаторівневу архітектуру.
- Сучасні стандарти в області інформаційних технологій (протоколи передачі даних, стандарти збереження даних і т.д.).
- Сучасні серверні платформи і програмне забезпечення.
- Сучасні рішення в області інформаційних технологій по розробці розподілених систем.
- Ефективні методи доступу до даних.
- Забезпечити простоту модернізації, можливість повторного використання програмного забезпечення (коду).

### **1.3.3 Вибір логічної архітектури програмного забезпечення інформаційної системи.**

Логічна архітектура програмного забезпечення інформаційної системи являє собою концептуальний опис структури програмного забезпечення системи. Кожна система має три рівні: рівень користувача, прикладний і рівень даних. Крім того, в усіх системах мається презентаційне програмне забезпечення (ПЗ), ПЗ обробки бізнес-правил і даних, а також ПЗ, що відповідає за збереження інформації. Відрізняються ж архітектури тільки організацією програмного забезпечення (програмного коду), індивідуальної для конкретної системи.

Багато сучасних інформаційних систем побудовані на базі дворівневої

архітектури «клієнт/сервер». При цьому клієнтське програмне забезпечення відповідає за обробку і відображення даних. Самі ж дані централізовано зберігаються на серверах, до яких підключаються клієнти. Часто час використання системи обмежується тривалістю такого з'єднання.

Клієнт-серверні системи добре працюють тільки в контрольованих середовищах, коли число користувачів передбачуване (це необхідно для виділення ресурсів). Однак ця архітектура стає неефективною, при великому числі користувачів або якщо їхнє число невідомо. Крім цього, масштабуємість таких систем невисока, тому що кількість безпосередніх з'єднань із сервером обмежена. Невеликі і можливості повторного застосування, оскільки користувачі прив'язані до визначеного формату бази даних. А тому що клієнтське програмне забезпечення містить блок обробки інформації, загальний обсяг програмного забезпечення досить великий (Такий тип клієнтських програм іноді називають «товстим» клієнтом). У цьому випадку при зміні алгоритмів обробки даних приходиться встановлювати нове програмне забезпечення на кожен клієнтський комп'ютер.

Можна отримати невелике поліпшення шляхом переміщення прикладних алгоритмів і блоків обробки даних на сервери даних (наприклад, за допомогою збережених процедур сервера баз даних). Таку архітектуру іноді називають 2,5-рівневою. Масштабуємість подібних систем трохи краща, але все рівно мала для виконання вимог інформаційної системи. Крім того, можливість повторного використання програмного забезпечення (коду) залишається на невисокому рівні.

Масштабуємість і ступінь повторного використання можна помітно поліпшити, додавши в архітектуру системи третій рівень. У такій багаторівневій архітектурі (її також називають N-рівневою) усі рівні – рівень користувача, прикладний і рівень даних – логічно розділені (рисунок 1.3). Багаторівнева архітектура задовольняє вимогам до інформаційної системи, що розробляється.

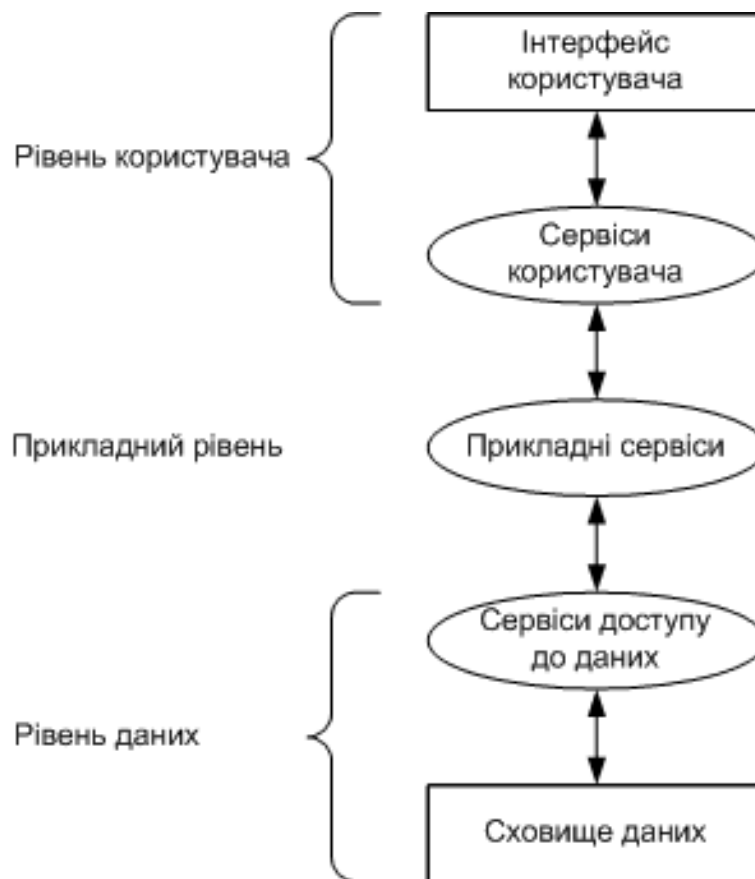


Рисунок 1.3 – Логічна архітектура програмного забезпечення інформаційної системи

Опишемо функції кожного рівня.

*Рівень користувача:* відображає дані і дозволяє користувачу редагувати їх. Існує два основних типи інтерфейсу: віконний (реалізується засобами підсистеми інтерфейсу користувача операційної системи) і на основі Web. Web-інтерфейси засновані на HTML, DHTML, XHTML чи XML, у результаті чого вони можуть відображатися будь-яким Web-оглядачем на будь-якій платформі.

*Прикладний рівень:* тут реалізовані бізнес-правила й обмеження на дані (перевірка коректності даних). І хоча його сервіси використовуються презентаційним рівнем, він не прив'язаний до якого-небудь клієнта – сервіси прикладного рівня доступні будь-якому клієнту. Бізнес-правила виражаються у формі прикладних алгоритмів, корпоративних правил і т.д. Обмеження на дані гарантують точність і цілісність інформації, що зберігається. Бізнес-правила звичайно реалізуються окремим модулем на централізованому сервері, що дає можливість доступу до нього відразу декільком клієнтам.

*Рівень даних*: прикладний рівень не має інформації про те, як і де зберігаються дані, які він обробляє. У цьому питанні він покладається на сервіси доступу до даних, що виконують всю роботу з отримання і передачі даних. Сервіси доступу до даних також реалізуються у виді ізольованих модулів, які знають про місце збереження інформації. Кожен модуль доступу до даних, як правило, відповідає і за цілісність сховища (наприклад, реляційної бази даних). Для багаторівневих програм у якості сховища інформації підходять сервера баз даних (MS SQL Server, Oracle, Sybase, MySQL, PostgreSQL і ін.), необхідні для обслуговування даних у таблицях і високопродуктивної вибірки інформації.

### **1.3.4 Вибір фізичної архітектури програмного забезпечення інформаційної системи.**

Виходячи з логічної архітектури програмного забезпечення, можна розробити його фізичну архітектуру. Фізична архітектура являє собою опис структури програмного забезпечення системи з погляду наявних у системі фізичних компонентів: серверів і клієнтських комп'ютерів.

Відповідно до описаної логічної архітектури, можна одержати дві фізичні архітектури, які так чи інакше задовольняють вимогам до системи. Перша фізична архітектура є триланковою (рисунок 1.4), друга – чотириланковою (рисунок 1.5).

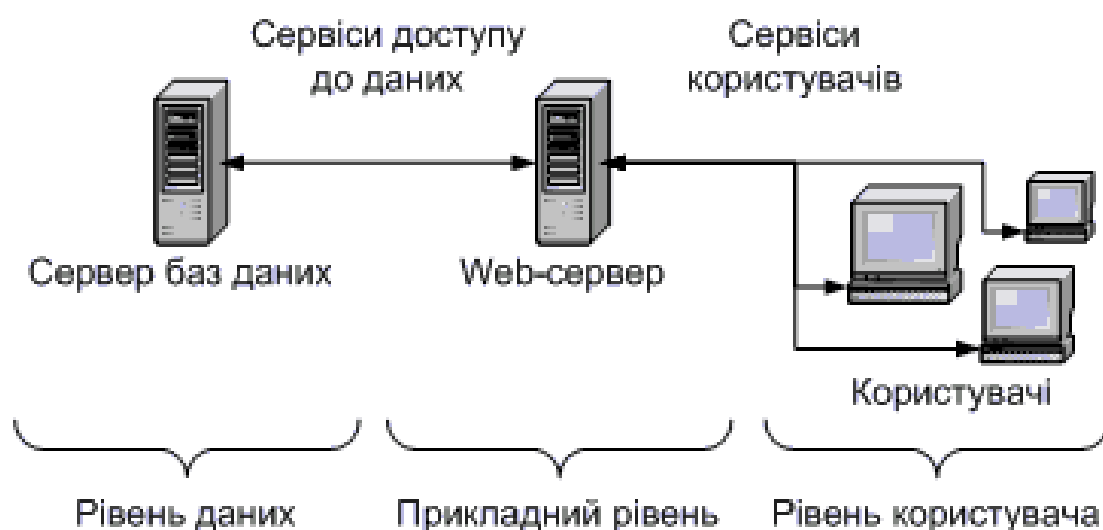


Рисунок 1.4 – Триланкова фізична архітектура програмного забезпечення інформаційної системи

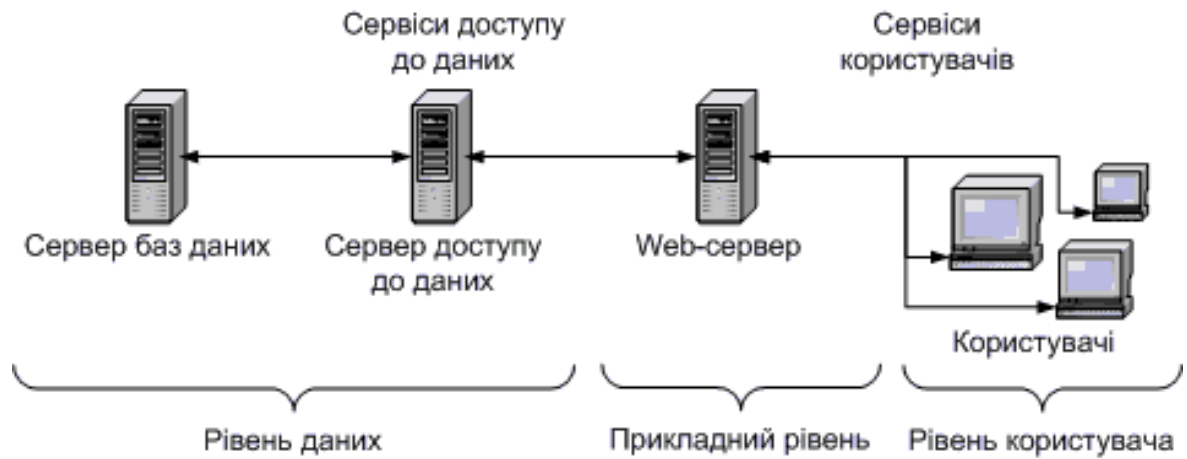


Рисунок 1.5 – Чотириланкова фізична архітектура програмного забезпечення інформаційної системи

Порівняємо отримані фізичні архітектури для інформаційної системи.

Переваги триланкової архітектури над чотириланковою:

- Краща продуктивність при невеликому числі користувачів.
- Порівняльна простота адміністрування системи.

Переваги чотириланкової архітектури над триланковою:

- Краща продуктивність при великому числі користувачів.
- Краща масштабуємість.
- Краща надійність.
- Кращі можливості по модернізації і повторному використанню.

Таким чином, чотириланкова архітектура є більш перспективною. Вона в більшій мері задовольняє вимогам до інформаційної системи.

### 1.3.5 Вибір технологій для розробки інформаційної системи.

Відповідно до обраної фізичної архітектури для інформаційної системи, система є розподіленою і має чотири ланки: сервер баз даних, сервер доступу до даних, Web-сервер і клієнт.

Розглянемо наступні технології для створення розподілених систем:

- DCOM/COM+, CORBA (об'єктно-орієнтовані технології).
- CGI, ISAPI/NSAPI, Servlets і інші.

Всі ці технології тим чи іншим способом організують виклик віддалених

процедур (методів) чи його аналог.

Взаємодію між клієнтами і Web-сервером раціонально організувати за допомогою CGI, ISAPI/NSAPI і/або Servlets, тому що ці технології орієнтовані на Web-інтерфейс. Використання DCOM/COM+, CORBA чи RMI у цьому випадку серйозно збільшило б вимоги до програмного забезпечення клієнтських машин, що суперечить вимогам до інформаційної системи (доступ до системи через Internet з будь-якого комп'ютера глобальної мережі). Усі зазначені технології (CGI, ISAPI/NSAPI, Servlets) вирішують однакові задачі і мають приблизно однакові характеристики по продуктивності. Servlets є більш гнучкою та одною з найсучасніших технологій, а також реалізується на мові високого рівня Java яка не залежить від операційної системи. Тому і обираємо цю технологію.

Для взаємодії серверів та доступу до БД потрібно застосовувати більш розвинуті технології.

Найбільш перспективною й універсальною технологією є CORBA, але вона теж має недоліки. CORBA – технологія досить молода і перебуває в стадії доведень і ліквідації «дитячих хвороб». Вона «сира» просто в силу свого надзвичайно швидкого розвитку. Не встигають розробники виправити помилки в поточній версії, потрібно створювати нову внаслідок появи нових можливостей. Також, гідним образом реалізовані не всі стандартні сервіси CORBA. Навіть найкраща специфікація не врятує положення при відсутності її доступної реалізації. Крім того, за CORBA (на відміну від того ж COM) треба платити.

Основний недолік DCOM/COM+ пов'язаний з необхідністю використовувати Windows-платформу. DCOM/COM+ на практиці довела свою придатність для розробки інформаційних систем. Це безкоштовна технологія і на ній засновано багато інших технологій (наприклад, OLE DB – найефективніший/найпродуктивний доступ до будь-яких даних). Ця технологія цілком задовольняє вимогам до інформаційної системи, перевірена на практиці, є безкоштовною, тому раціонально використовувати її для взаємодії серверів. Так як ми обрали Servlets то логічно обрати технологію яка реалізується на Java

тому наш вибір це EJB, крім цього EJB має багато переваг серед своїх конкурентів.

Як говорилося раніше, на рівні користувача необхідно використовувати Web-інтерфейс (HTML/DHTML, CSS, JavaScript, DOM, Java-аплети). Клієнтське програмне забезпечення – Web-оглядач.

### **1.3.6 Розробка варіантів концепції інформаційної системи.**

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

- серверна платформа і серверне програмне забезпечення;
- система управління базами даних;
- клієнтська платформа і клієнтське програмне забезпечення;
- засоби розробки;
- мережне середовище;
- технічні засоби.

Від вибору перерахованих компонентів залежить ефективність інформаційної системи. Розглянемо кожен компонент інформаційної системи.

#### *Серверна платформа і серверне програмне забезпечення.*

На Web-сервері повинен бути встановлений HTTP-сервер. Найрозповсюдженими HTTP-серверами, що задовольняють зазначеним вимогам і мають високу продуктивність, є Apache Server. На Web-сервері повинна бути встановлена ОС яка підтримує цей сервер.

Так як використовуються технології організовані на Java то операційна система на сервері доступу до БД може бути будь яка з перелічених:

- - Macintosh;
- - MS Windows;
- - Linux.

Також потрібний Java WEB сервер встановлений на сервері будь який з:

- JBoss;
- Sun Application Server;
- GlassFish;
- Tomcat.

На сервері баз даних повинна бути встановлена серверна система управління базами даних (СУБД).

*Система управління базами даних.*

Необхідна сучасна серверна реляційна СУБД, що має високу продуктивність, здатна працювати з великими обсягами інформації і підтримує стандарт SQL. Крім того, СУБД повинна працювати під управлінням Windows. Серверними СУБД, що задовольняють зазначеним вимогам, на сьогодні є:

- MS SQL Server.
- MySQL.
- PostgreSQL .
- Sybase.
- Oracle (Standard Edition чи Enterprise Edition).

*Клієнтська платформа і клієнтське програмне забезпечення*

На клієнтських машинах може використовуватися будь-яка операційна система з можливістю встановлення віддаленого мережного з'єднання і графічним інтерфейсом. Операційні системи, що рекомендуються – MS Windows, Linux, тощо.

Програмне забезпечення, з яким буде працювати користувач, являє собою Web-оглядач з підтримкою HTML, JavaScript, CSS (наприклад, MS Internet Explorer, FireFox, Opera тв. інш.).

*Мережне середовище.*

Підприємство має локальну комп'ютерну мережу Ethernet, яку можна використовувати як мережне середовище інформаційної системи. Можливо, потрібно буде внести зміни в конфігурацію існуючої комп'ютерної мережі. Основна вимога до мережі – швидкість передачі даних не менше 10 Мбіт/с.

### *Технічні засоби*

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

- Мікропроцесор класу Pentium 2600 МГц і вище.
- Оперативна пам'ять RAM 512 Мб і більше.
- Жорсткий диск з 20 Гб вільного місця і більше.
- Мережна карта Ethernet (не менше 10 Мбіт/с).

Технічні засоби клієнтських машин повинні відповідати вимогам операційної системи, що використовується. Конфігурація комп'ютера користувача може бути наступною:

- Мікропроцесор класу Pentium 300 МГц і вище.
- Оперативна пам'ять RAM 64 Мб і більше.
- Жорсткий диск з 4,3 Гб вільного місця і більше.
- Мережна карта Ethernet.

### *Засоби розробки.*

У якості засобу розробки є об'єктно-орієнтована мова програмування Java це обґрунтовується по-перше незалежністю від платформи а по-друге безкоштовністю.

### *Варіанти концепції інформаційної системи.*

Таким чином, розглянувши компоненти системи і вибравши найбільш оптимальні, можна визначити кілька варіантів концепції інформаційної системи (див. таблицю 1.1).

Таблиця 1.1 – Варіанти концепції інформаційної системи

№	ПЗ Web-сервера (ОС, HTTP-сервер)	ПЗ сервера доступу до даних (ОС)	ПЗ сервера баз даних (ОС, СУБД)
1	MS Windows Server, MS IIS	MS Windows Server, GlassFish	MS Windows Server, InterBase
2	Linux, Apache Server	MS WindowsServer GlassFish	MS Windows Server, MS SQL Server
3	MS Windows 2000 Server, MS IIS 5.0	Linux, Jboss	MS Windows Server, MySQL
4	Linux, Apache Server	Linux, Jboss	Linux, MySQL

Крім того, кожен варіант концепції системи має відповідне технічне (описане вище) і організаційне забезпечення. Причому, і технічне, і організаційне забезпечення будуть однаковими для усіх варіантів, оскільки обране інформаційне забезпечення має приблизно однакові вимоги до апаратних ресурсів і обслуговування.

При виборі варіантів концепції інформаційної системи, були враховані наступні особливості:

- На Web-сервері і сервері доступу до даних повинна працювати операційна система з повною підтримкою Java, тобто Windows, Linux, Macintosh, так як з фінансової точки зору Linux є безкоштовною ОС то цей вибір є найкращим. Сервер баз даних може працювати під управлінням як Windows, так і UNIX. Якщо сервер баз даних не буде виділений (наприклад, сполучений із сервером доступу до даних), то необхідне використання Linux.

- Краще на всіх серверах встановлювати однакову операційну систему, тому що їх легше буде підтримувати (складно буде знайти фахівця, який буде добре знати кілька серверних операційних систем).

- СУБД MySQL ідеально підходить так як платформно-незалежний до того ж безкоштовний.

- Використання ОС Windows нераціонально, оскільки її могутні можливості не будуть використані повною мірою, а витрати на її придбання досить великі.

### **1.3.7 Вимоги, що впливають на якість інформаційної системи.**

Якість інформаційної системи і якість програмного забезпечення не піддаються точному визначенню і виміру. На даний момент, не існує ні метрології якості інформаційних систем, ні реальних методів оцінки якості програмного забезпечення, ні методів оцінки якості процесу проектування, розробки і впровадження.

Проте, задача поставлена і вимагає вирішення. Складемо список вимог (критеріїв), що можуть впливати на якість інформаційної системи (частина з них взята зі стандарту ISO-9126):

- Здатність підвищити ефективність працівників.
- Функціональність.
- Відповідність призначенню.
- Точність.
- Повнота інформації.
- Здатність взаємодіяти із середовищем.
- Відповідність нормам.
- Безпека (захист від взлому даних і інших злочинних дій).
- Надійність.
- Зрілість ("обкатаність").
- Стійкість до відмов.
- Стійкість до помилок користувача.
- Здатність відновлюватися після збоїв.
- Масштабуємість.
- Придатність до використання.
- Зрозумілість.
- Вивчаємість.
- Зручність і простота в роботі.
- Ефективність.
- Продуктивність і час відгуку.
- Споживання ресурсів.
- Супроводжуємість.
- Аналізуємість (діагностика причин помилок і зіставлення з первісним кодом).
- Придатність до змін.
- Стабільність.
- Тестуємість.

Потрібно відзначити, що виконання всіх зазначених вимог ще не гарантує якість інформаційної системи. Також, невідповідність системи якій-небудь вимозі не говорить про те, що система неякісна. Як уже говорилося вище, не

існує формальних методів оцінки якості інформаційної системи.

## **1.4 Техніко-економічне обґрунтування розробки інформаційної системи на основі функціонально-вартісного аналізу.**

### **1.4.1 Постановка задачі для функціонально-вартісного аналізу.**

Для вибору найбільш ефективного варіанта концепції інформаційної системи управління завданнями 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;
- вартісний аналіз.

Обрані варіантами концепції системи представлені в таблиці 1.2.

Таблиця 1.2 – Варіанти концепції інформаційної системи

№	ПЗ Web-сервера (ОС, HTTP-сервер)	ПЗ сервера доступу до даних (ОС)	ПЗ сервера баз даних (ОС, СУБД)
1	MS Windows Server, MS IIS	MS Windows Server, GlassFish	MS Windows Server, InterBase
2	MS Windows Server, Apache Server	MS Windows Server, JBoss	MS Windows Server, MS SQL Server
3	Linux, Apache Server	Linux, GlassFish	Linux, MySQL
4	Linux, Apache Server	Linux, JBoss	Linux, MySQL

Конфігурація кожного сервера може бути наступною:

- Мікропроцесор класу Pentium 2600 МГц і вище.
- Оперативна пам'ять RAM 512 Мб і більше.
- Жорсткий диск з 20 Гб вільного місця і більше
- Мережна карта Ethernet (не менше 10 Мбіт/с).

Конфігурація комп'ютера користувача може бути наступною:

- Мікропроцесор класу Pentium 300 МГц і вище.

- Оперативна пам'ять RAM 64 Мб і більше.
- Жорсткий диск з 4,3 Гб вільного місця і більше.
- Мережна карта Ethernet.

Як мережне середовище можна використовувати кабелі типу «вита пара» категорії 3, 5 і вище. Активне мережне обладнання повинне підтримувати як мінімум швидкість 10 Мбіт/сек.

Організаційне забезпечення передбачає обслуговування системи протягом строку окупності. Розрахунок строку окупності приведений в організаційно-економічному розділі даної роботи.

#### **1.4.2 Функціональний аналіз.**

Якщо оцінювати варіанти реалізації функцій з погляду надання необхідної функціональності (для реалізації системи), то всі вони надають необхідну функціональність повною мірою. Таким чином, говорити про недоліки чи переваги окремих варіантів зараз передчасно. Система, реалізована по кожному з цих варіантів, буде працювати успішно. Для того, щоб оцінити різні варіанти необхідно визначити основні параметри системи. Основними параметрами системи відповідно до вимог замовника є:

- Функціональність (відповідність призначенню, точність, повнота інформації, безпека).
- Ефективність (продуктивність і час відгуку, споживання ресурсів).
- Надійність (стійкість до відмов, стійкість до помилок користувача, здатність відновлюватися після збоїв).
- Масштабуємість.
- Придатність до використання (зрозумілість, вивчаємість, зручність і простота в роботі).
- Супроводжуємість (аналізуємість (діагностика причин помилок і зіставлення з первісним кодом), придатність до змін, стабільність, тестуємість).

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 варіанті системи і відповідним параметрам. Експертами в даному випадку можуть бути викладачі кафедри інформаційних технологій НУК, оскільки вони є фахівцями в області інформаційних технологій, фахівцями з інформаційної системи, автоматизації функцій якої присвячена дана робота, і потенційними користувачами інформаційної системи, що розробляється. Усереднені експертні оцінки (10 бальна шкала) по кожному варіанту системи і відповідним параметрам представлені в таблиці 1.3.

Таблиця 1.3 – Основні параметри системи

Параметр	Оцінка в балах по варіанту №, $B_i$			
	1	2	3	4
Функціональність	10	10	10	10
Ефективність	7	7	8	9
Надійність	8	8	8	8
Масштабуємість	9	9	9	10
Супроводжуємість	8	8	7	8

На наступному етапі функціонального аналізу потрібно визначити вагові коефіцієнти для кожного з параметрів. У даному випадку вони також визначаються експертним шляхом. Значення вагових коефіцієнтів для кожного з параметрів приведені в таблиці 1.4. Насправді, сума усіх вагових коефіцієнтів повинна дорівнювати одиниці, але для наочності проведемо розрахунок з ненормалізованими коефіцієнтами, а потім розділимо отриманий результат (коефіцієнт технічного рівня) на суму вагових коефіцієнтів (5.0).

Таблиця 1.4 – Вагові коефіцієнти для параметрів системи

Параметр	Коефіцієнт, $\varphi_i$
Функціональність	0.3
Ефективність	0.2
Надійність	0.3
Масштабуємість	0.1
Супроводжуємість	0.1

Після визначення вагових коефіцієнтів  $\varphi_i$  і бальних оцінок  $B_i$  для всіх

параметрів системи, необхідно визначити коефіцієнти технічного рівня для кожного з параметрів за кожним. Результати розрахунку приведені в таблиці 1.5.

За даними таблиці 1.5 розраховується коефіцієнт технічного рівня  $K_{TP}$  для кожного з варіантів по формулі:

$$K_{TP} = \sum_{i=1}^N K_{TP_i},$$

де  $K_{TP_i}$  – показник технічного рівня для  $i$ -го параметра.

Також для нормалізації значення коефіцієнта технічного рівня, отримане по попередній формулі значення потрібно розділити на суму вагових коефіцієнтів, що дорівнює 5.0. Результати розрахунку приведені в таблиці 1.6.

Таблиця 1.5

Параметр	Варіант	Оцінка параметра в балах, $B_i$	Ваговий коефіцієнт, $\varphi_i$	Показник технічного рівня, $K_{TP}$
Функціональність	1	10	0.3	3
	2	10		3
	3	10		3
	4	10		3
Ефективність	1	7	0.2	1.4
	2	7		1.4
	3	8		1.6
	4	9		1.9
Надійність	1	8	0.3	2.4
	2	8		2.4
	3	8		2.4
	4	8		2.4
Масштабуємість	1	9	0.1	0.9
	2	9		0.9
	3	9		0.9
	4	10		1
Супроводжуємість	1	8	0.1	0.8
	2	8		0.8
	3	7		0.7
	4	8		0.8

Таблиця 1.6 – Коефіцієнти технічного рівня для варіантів концепції системи

Варіант	Коефіцієнт технічного рівня, $K_{TP}$
1	7.5
2	7.5
3	8.6
4	9.1

Найкращим на етапі функціонального аналізу є варіант, який має максимальне значення коефіцієнта технічного рівня  $K_{TR}$ , тобто варіант 4.

### **1.4.3 Очікувані результати й ефективність реалізації обраного варіанта концепції системи.**

Тому що обраний варіант реалізації задовольняє всім основним вимогам до системи, очікуються ефективні і високі результати при рішенні поставлених задач:

- створення єдиної комп'ютерної мережі підприємства;
- забезпечення надійного контролю цілісності даних;
- підвищення ефективності виконання програм за рахунок використання потужності сервера з застосуванням технології "клієнт-сервер";
- використання існуючого обладнання у якості клієнтських місць.

### **1.4.7. Умови приймання системи.**

Приймання створеної системи здійснюється представниками виконавця і замовника відповідно до технічного завдання. Складається двосторонній акт, що стає підставою для розрахунків.

## 2. Розробка проектних рішень по системі.

### 2.1. Загальносистемні проектні рішення

#### 2.1.1. Організаційна і функціональна структура.

У процесі свого функціонування система взаємодіє з користувачами з різних підрозділів для виконання ними різних функцій. Схема цієї взаємодії, побудована в нотації варіантів використання системи (use case) показана на рисунку 2.1.

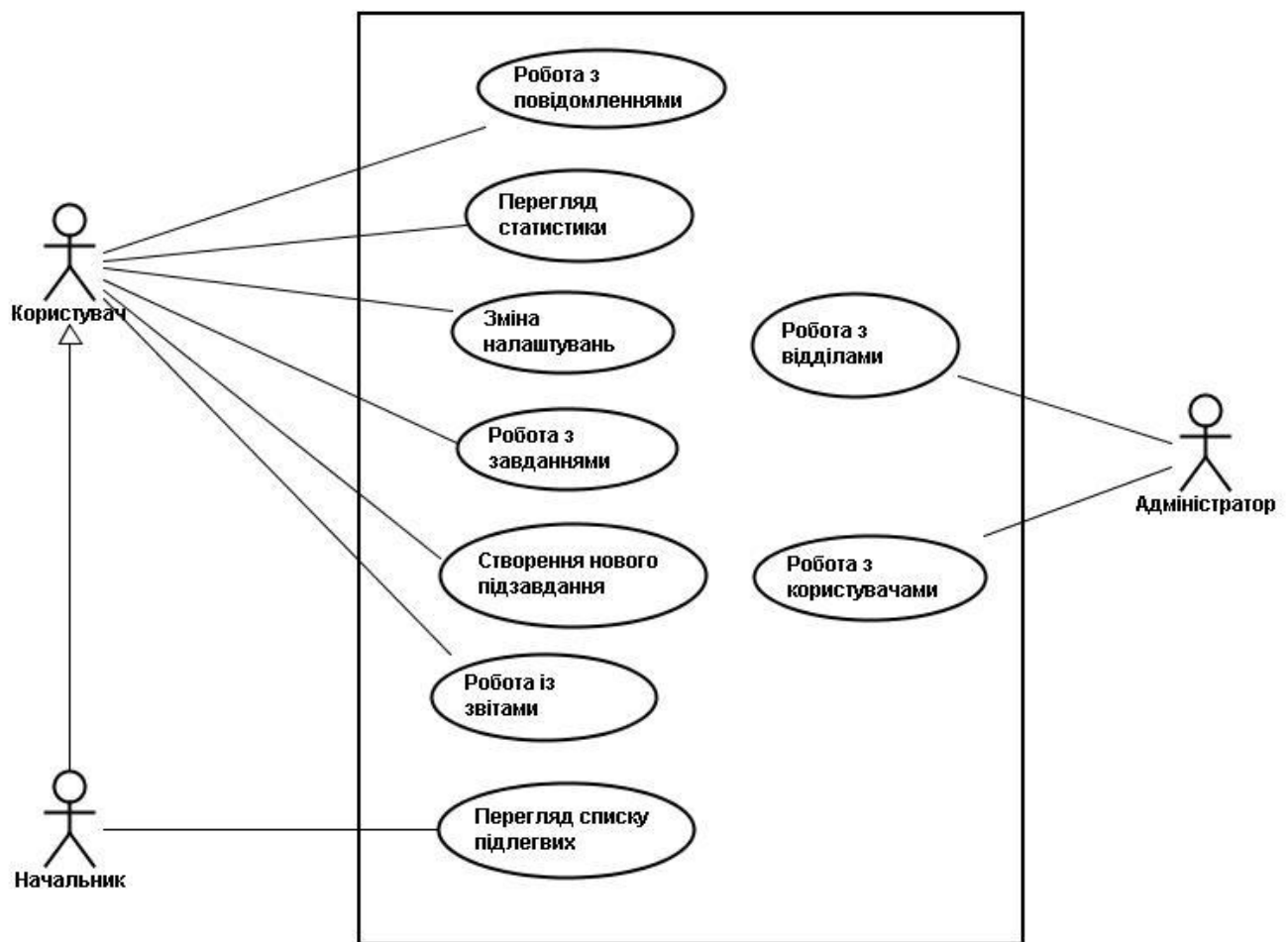


Рисунок 2.1. Схема організаційно-функціональної структури.

У ході своєї роботи, працівники підрозділів використовують наступні функції

Користувач – перегляд та управління особистими повідомленнями, перегляд, створення нових завдань та під-завдань, створення нового звіту та перегляд минулих, перегляд особистої статистики, та налаштувань.

Якщо у користувача є підлеглі, тоді він може переглянути їхні завдання та статистику.

Адміністратор – перегляд та управління (зміна, видалення, створення нового) відділами та користувачами.

### **2.1.2. Опис функцій, що автоматизуються.**

Задачею інформаційної системи, що створюється, є автоматизація наступних функцій користувача:

- Облік відділків та їх працівників.
- Облік завдань та інформацію про їхній статус.
- Формування звітів.

Усі інші вимоги не є головними і можуть змінюватися під час розробка або експлуатації.

Для створення інформаційної системи необхідно розробити єдину базу даних для збереження інформації про відділи та їх співробітників, про отриманні ними завдання і іншу. Також повинно бути передбачене програмне забезпечення для адміністрування системи.

### **2.1.3. Опис постановки задачі.**

На підставі аналізу вимог і обмежень, а також попередніх концепцій реалізації системи, сформулюємо опис постановки задачі. Необхідно розробити систему управління завданнями, що буде задовольняти перерахованим вище функціональним вимогам. Необхідно також забезпечити надійність збереження і безпека доступу до інформації Системи. Система повинна обслуговувати одночасну безперебійну роботу багатьом користувачів, причому зміни, внесені одними користувачами повинні бути доступні всім іншим користувачам негайно.

Програмне забезпечення повинне бути побудоване на платформі SUSE Linux/Apache/MySQL/JBoss/Java/Eclipse і реалізовувати графічний інтуїтивно зрозумілий WEB інтерфейс користувача.

#### 2.1.4. Проектна оцінка надійності системи.

Для оцінки надійності Системи використовується принцип визначення показників надійності Системи за характеристиками надійності комплектуючих елементів, що дозволяє вести розрахунок в процесі проектування Системи, 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 серверної станції, вихід з ладу яких є найбільш ймовірним:

- блок живлення;
- Smart-UPS APC;
- комутатор.

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

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

Середній час роботи блоку живлення ПК:

$T_{cp} = 10\ 000$  годин.

Середній час роботи Smart-UPS APC:

$T_{cp} = 12\ 000$  годин.

Середній час роботи комутатора:

$T_{cp} = 15\ 000$  годин.

Розрахунок буде проводитися для визначення ймовірності безвідмовної роботи кожного з перелічених 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;

$\lambda_0$  – інтенсивність відмов елемента підсистеми протягом середнього часу безвідмовної роботи  $T_{cp}$ .

Інтенсивність відмов елемента підсистеми за середній час безвідмовної роботи  $T_{cp}$  розраховується за формулою  $\lambda_0 = \frac{1}{T_{cp}}$

Для блоку живлення:

$$\lambda_0 = 1 / 10\,000 = 0,0001$$

$$P(t)_{pc} = \exp(-0,0001 * 1000) = 0,920$$

Для UPS:

$$\lambda_0 = 1 / 12\,000 = 0,000083$$

$$P(t)_c = \exp(-0,000083 * 1000) = 0,943$$

Для комутатора:

$$\lambda_0 = 1 / 15\,000 = 0,000066$$

$$P(t)_{mc} = \exp(-0,000066 * 1000) = 0,961$$

Під час визначення надійності підсистеми  $P_n(t)$  через відомі показники надійності її елементів вводяться два припущення: відмови елементів статично незалежні, відмова будь-якого елемента приводить до відмови підсистеми. Прийняті припущення дозволяють використовувати теорему добутку ймовірностей, яка розраховується за формулою

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

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

$n$  – кількість елементів.

Надійність технічного забезпечення Системи протягом 1000 годин роботи визначається добутком ймовірностей безвідмовної роботи блоків живлення, UPS та комутатора.

$$P_n(t) = P(t)_{pc} * P(t)_c * P(t)_{mc} = 0,90 * 0,943 * 0,961 = 0,896$$

Проведений розрахунок показав, що ймовірність безвідмовної роботи Системи протягом 1000 годин складає близько 90%.

### **2.1.5. Загальний опис системи.**

Система являє собою інструмент керування процесом виробництва судів, побудований на основі мережі підприємства малотоннажного суднобудування. Основними функціональними характеристиками системи є:

- ведення загальної інформаційної бази з розподіленим доступом;
- авторизований доступ до даних на основі прав доступу;

- відображення змін, внесених з будь-якого робочого місця;
- можливість створення і редагування завдань і оперативного контролю ходу їхнього виконання.

Система має звичний графічний інтерфейс користувача, що знижує витрати на навчання персоналу, і побудована з максимальним використанням існуючого на підприємстві в даний момент устаткування і програмного забезпечення.

#### **2.1.6. Захист від несанкціонованого доступу.**

Захист системи від несанкціонованого доступу реалізовано на системному рівні. При запуску системи відбувається авторизація користувача. Доступ до даних і функцій дозволяється тільки при правильному введенні користувальницького імені і пароля.

### **2.2. Рішення по організаційному забезпеченню.**

#### **2.2.1. Організаційна структура підприємства.**

У ході впровадження і використання системи, її функціями охоплюються всі суб'єкти підприємства. Схема організаційної структури підприємства при взаємодії із системою ( у виді діаграми варіантів використання) приведена на рисунку 2.2

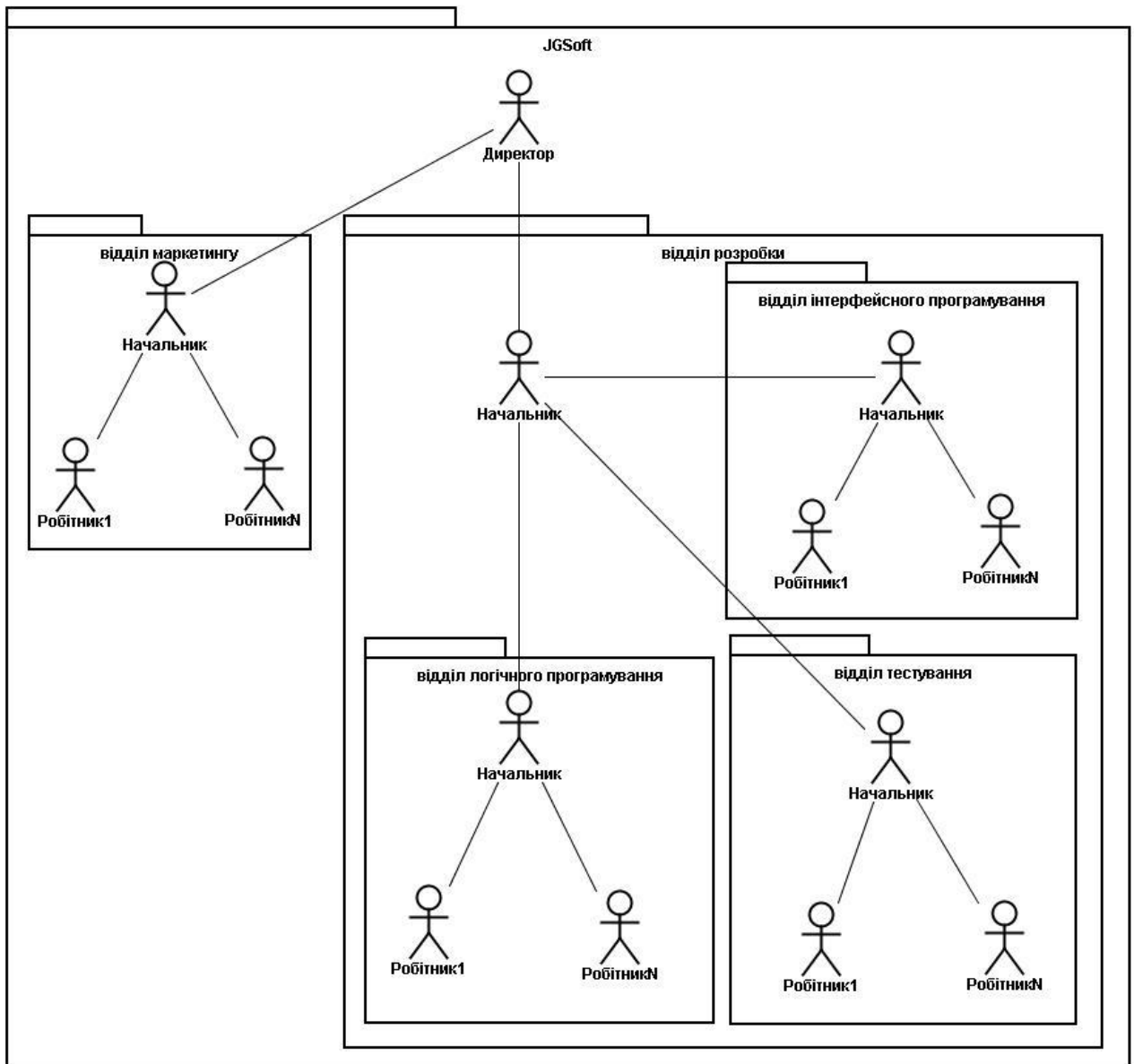


Рисунок 2.2. – Суб'єкти підприємства, які взаємодіють із системою.

### 2.2.2. Інструкція користувача.

Для нормального функціонування системи, всі операції, проведені в ній користувачами, повинні відповідати Інструкції користувача, приведеної в Додатку Б.

### 2.2.3. Заходи щодо організаційного забезпечення.

Ефективна робота системи керування діяльністю підприємства вимагає організаційних заходів, цілями яких є підтримка даних про процеси, що відбуваються, в актуальному стані і грамотній своєчасній реакції користувачів на інформацію, що їй надається системою:

- навчання користувачів роботі в системі згідно виконуваних функцій (посадових обов'язків);
- досвідчена експлуатація Системи з імітацією керування будівництвом об'єкта, що був побудований раніше з максимальним наближенням до реальності (у виді ділової гри);
- мінімізація часу, що проходить між зміною версій документів і зміни їхнього стану (іншими словами, негайна реакція керівництва на зміни у виробничому процесі);
- обов'язкова участь у формуванні даних усіх суб'єктів підприємства; регулярний оперативний контроль зв'язаних процесів керівниками підрозділів і контроль виробничих циклів.

#### **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 й 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 по інформаційному забезпеченню.**

#### **2.3.1. Опис інформаційного забезпечення системи.**

Одну з основ інформаційного забезпечення системи складає визначення її внутрішнього стану, що буде відбито в моделі класів. На даному етапі нас цікавлять довготривалі та постійні сутності бізнес-процесів. Цей вид класів будемо називати класами-сутностями (чи класами предметної області).

Для створення моделі класів складемо словник даних, ґрунтуючись на сутностях, виявлених на етапі бізнесу-аналізу.

*Користувач* – людина яка взаємодіє з системою.

*Начальник* – користувач, на якого є хоча б одне посилання у іншого користувача.

*Звіт* – звіт формується користувачем, та відсилається безпосередньому начальнику.

*Повідомлення* – користувачі можуть обмінюватися повідомленнями.

*Завдання* – завдання сформоване одним користувачем до іншого.

*Дія* – зміна статусу завдання (створено, відхилене, тощо) проходить через створення користувачем "дії".

На даному етапі проектування сутності словника даних стають кандидатами для класів проектованої системи. Модель бізнес-сутностей системи показана на рисунку 2.3.

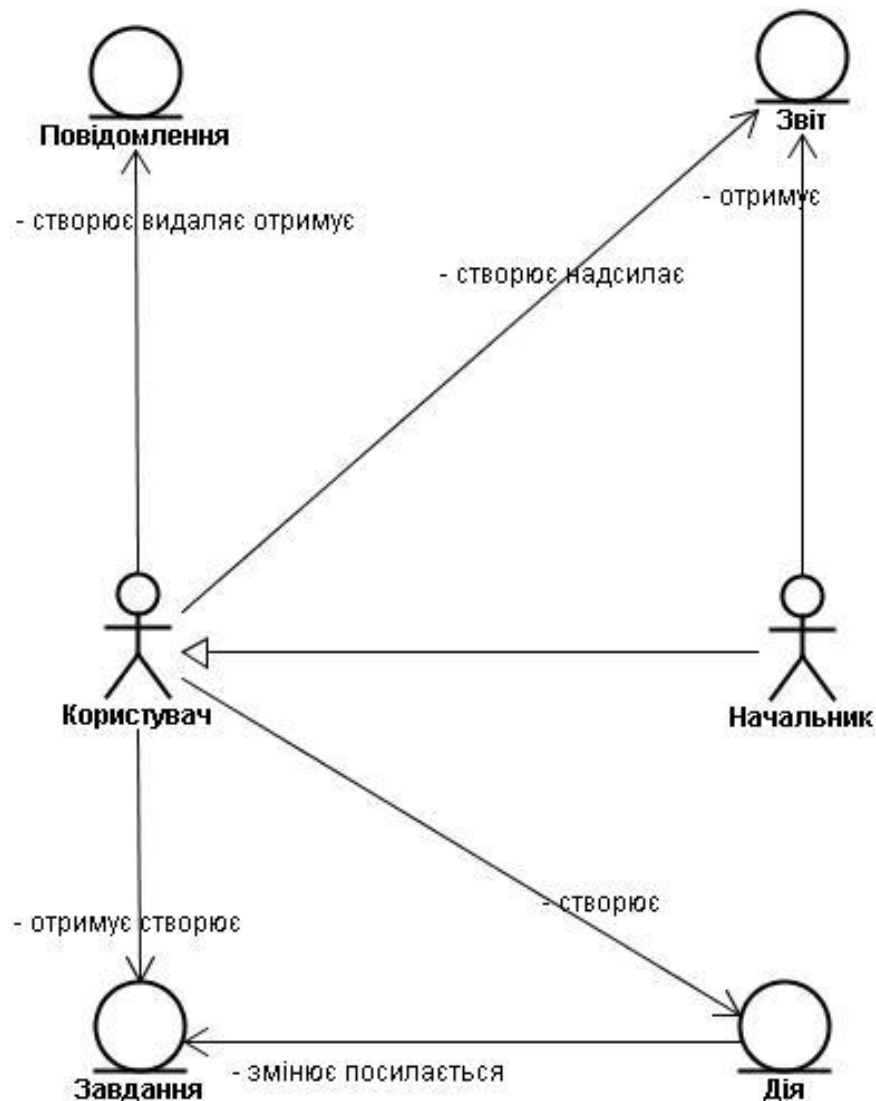


Рисунок 2.3. – Діаграма бізнес-сутностей системи.

Для подальшої деталізації сутностей і зв'язків між ними використовуємо діаграму «сутність-зв'язок» (ERD), що дозволить найбільше повно представити

концептуальну модель даних, а також перейти в наслідку до їх логічного представлення. Побудована на підставі аналізу бізнес-сутностей системи модель у термінах «сутність-зв'язок» у нотації UML приведена на рисунку 2.4.

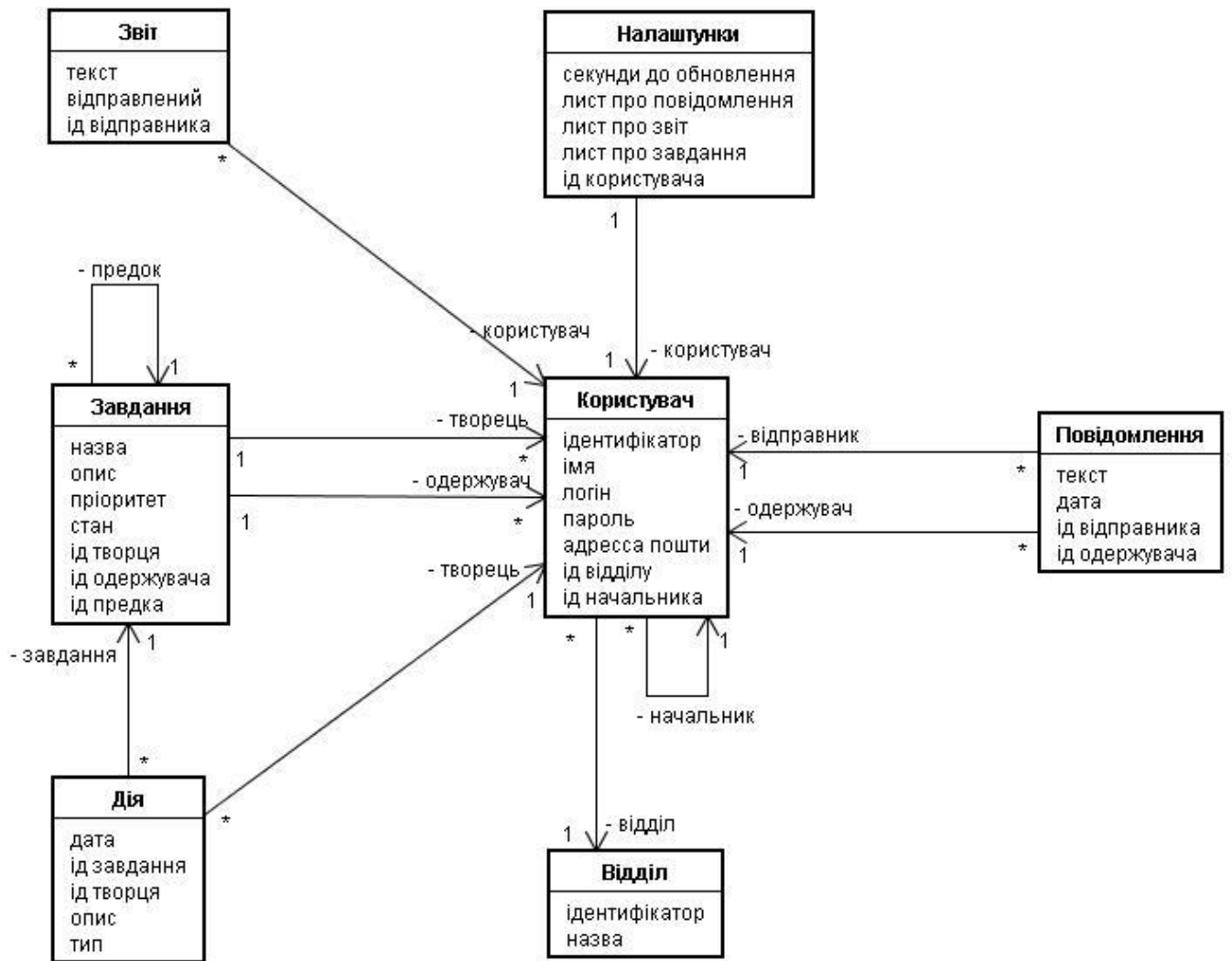


Рисунок 2.4. – Інформаційна модель об’єкту управління.

## 2.4. Рішення по технічному забезпеченню.

### 2.4.1. Схема автоматизації.

Раціональний вибір технічних засобів, необхідних для впровадження автоматизованої системи буде значною мірою визначати загальний чи успіх невдачу проекту. На підставі прийнятих загальносистемних рішень були визначені склад і організація комплексу технічних засобів.

*Вимоги до продуктивності:*

Обчислювальний комплекс повинний забезпечувати достатню продуктивність для рішення передбачуваних задач при бажаній мінімізації фінансових витрат на його придбання. Крім того, потрібно забезпечити спритний час реакції системи при роботі в інтерактивних режимах, що складають значну частину від всіх експлуатаційних режимів. Цим буде забезпечена комфортна робота користувачів. Проектована система не буде пред'являти підвищених вимог до обчислювальної потужності. Створювану систему можна віднести до класу облікових задач, у яких загальна продуктивність системи визначається швидкістю операцій уведення-висновку при неповному залученні потужності центрального процесора. У такий спосіб будь-яке сучасне апаратне рішення з пропонованих на ринку буде відповідати вимогам до продуктивності. Розподіл програмних компонентів майбутньої системи дозволить не збільшити навантаження на існуючі клієнтські робочі місця і не висувати вимог до їх модернізації.

*Вимоги до надійності:*

Розв'язувана задача висуває стандартні вимоги до надійності апаратних засобів без вимоги забезпечення повної відмовостійкості. Основною вимогою в даній групі буде вимога до схоронності даних. Передбачаються наступні заходи для його виконання:

- включення сервера в електромережу через пристрій безперебійного живлення (UPS);

- використання на сервері БД апаратного IDE RAID контролера з можливістю віддзеркалення дисків (дане рішення має досить сприятливу вартість);

- використання спеціального пристрою резервного копіювання; як такий пристрій можливе застосування CD-RW привода.

*Вимоги до експлуатаційних характеристик:*

З перерахованих вимог можна зробити висновок про застосовність для сервера БД і клієнтських робочих місць стандартного апаратного забезпечення на платформі x86. До сервера БД пред'являється традиційний набір вимог для такого рішення.

#### 2.4.2. Структурна схема комплексу обчислювальних засобів.

Проектована програмна система буде виконана за схемою "клієнт-сервер", тому апаратне рішення буде складатися з обчислювальної підсистеми і мережі передачі даних. Принципова схема виконана у виді UML діаграми розміщення наведена на рисунку 2.5.

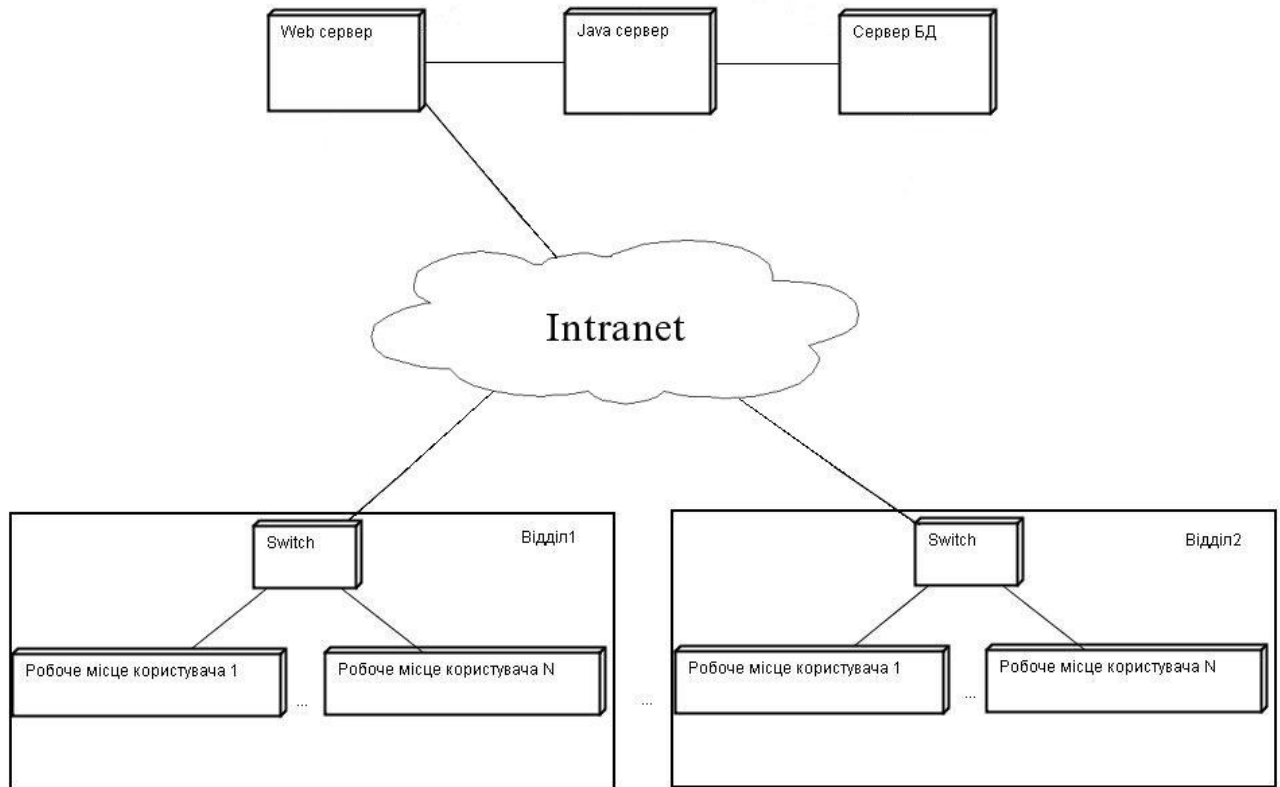


Рисунок 2.5. – Діаграма розміщення комплексу технічних засобів.

#### 2.5. Опис алгоритмів функціонування системи.

Для уточнення алгоритмів функціонування розроблюваної системи використовуємо засобу об'єктно-орієнтованого аналізу й опису поведінки об'єктів. При цьому будемо розглядати як поведінку системи в цілому (для уточнення виду, характеру і порядку взаємодії між класами), так і реалізацію функціональності окремих класів системи. Діаграма класів системи, побудована на підставі аналізу сутностей і операцій над ними, представлена на рисунку 2.4.

Розглянута вище діаграма класів являє собою логічну модель статичного представлення системи, що моделюється. Мова йде про те, що на даній діаграмі

зображуються тільки взаємозв'язки структурного характеру, що не залежать від чи часу реакції системи на зовнішні події. На відміну від цього представлення, загальний алгоритм поведінки системи може бути описаний за допомогою діаграми станів (statechart diagram). Загальний алгоритм поведінки важливий для визначення взаємодій між компонентами системи й уточнення причин, що викликають та чи інша відповідна дія. Алгоритм поведінки системи в нотації діаграми станів приведений на рисунку 2.6.

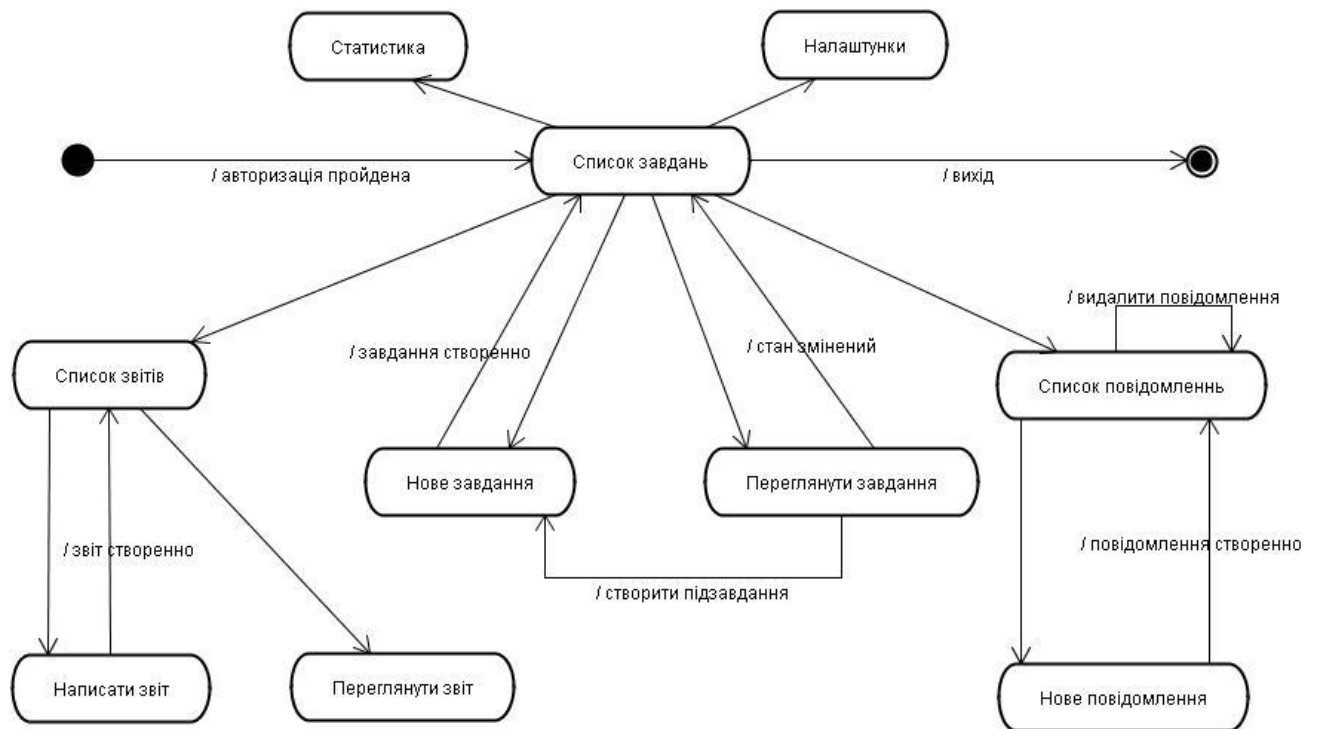


Рисунок 2.6. – Загальний алгоритм поведінки системи.

Після успішного входу в систему користувач бачить екран с двома списками завдань, перший завдання які він отримав, другий які він створив. При переході за посиланням на перегляд отриманого завдання, користувач може відмовитися від нього, завершити або створити нове підзавдання. Якщо ж відбувається перегляд створеного користувачем завдання тоді він може його видалити (якщо стан завдання не відмінний від начального та завдання не має "потомків"), створити підзавдання або якщо завдання було відхилене або завершене він може його схвалити чи навпаки (з указівкою причин). Усі перелічені дії повертають на попередній екран.

З цього ж екрану користувач може перейти:

– на екран де може змінити свої налаштування (кількість секунд для оновлення екрану завдань, отримувати листа при отриманні завдання, звіту або повідомлення);

– на екран статистики де відображується статистика за обраний період, кількість виконаних та отриманих завдань, а також їх перелік, якщо у користувача є підлеглі то він може переглянути їхню статистику теж;

– на екран звітів, де бачить свої звіти, або якщо у користувача є підлеглі то відіслані ними звіти теж, створити новий звіт або відіслати існуючий начальству;

– на екран повідомлень, де бачить повідомлення адресовані йому, або надіслані ним, він може їх видалити, створити нове;

Користувач може вийти з системи, користувач може завершити дію веб-оглядача, але у цілях безпеки рекомендується виходити з системи.

## **2.6. Рішення по програмному забезпеченню.**

### **2.6.1. Структура і функції частин програмного забезпечення.**

Функціональність і структуру проектованої автоматизованої системи поряд з моделлю поведження визначить модель її станів. Будемо будувати модель станів шляхом уточнення моделі бізнес-сутностей, побудованої нами на підставі вимог користувачів до функцій системи й оброблюваних даних. Вхідними даними для даного етапу проектування будуть моделі сутностей, а результатом стане модель специфікації проектних конструкцій розроблювального програмного забезпечення. У відповідності з прийнятим у для даної роботи підходом об'єктно-орієнтованого аналізу і проектування виконаємо подальшу специфікацію проектних рішень у виді діаграм класів. Класи визначають стан пропонованої програмної системи (через значення своїх атрибутів) і, значною мірою, її поведження (через операції чи методи класів). Діаграма бізнес-сутностей приведена на рисунку 2.7.

Сформульовані користувальницькі вимоги тепер приймемо за обмеження. Поряд з моделлю сутностей для проектування класів будемо користатися

побудованими раніше діаграмами прецедентів (рисунок 2.1), у відповідності до рекомендацій методики Rational Unified Process. При цьому модель класів буде відповідати словнику даних, що був отриманий на підставі аналізу прецедентів, і який є текстовим аналогом моделі сутностей.

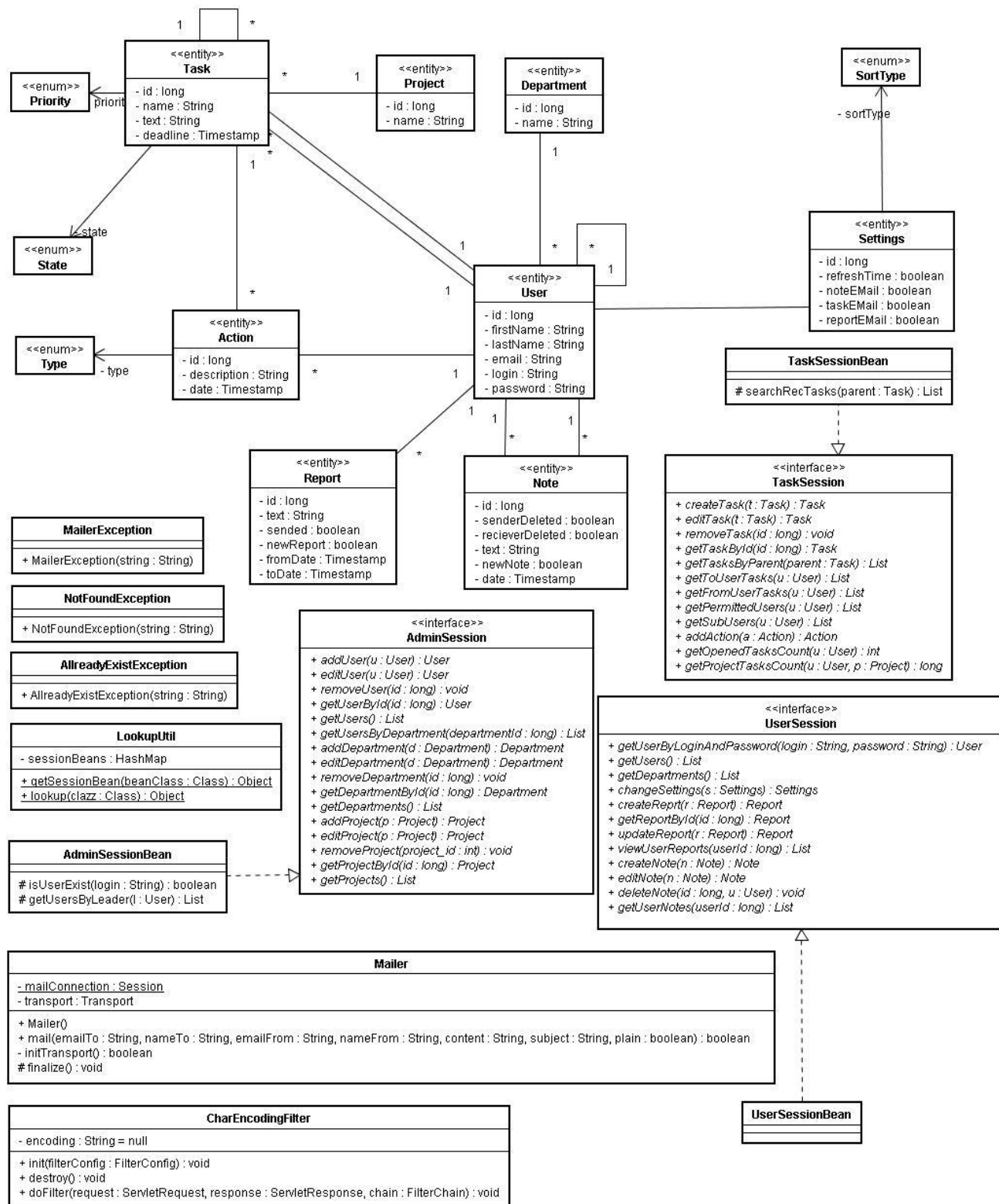


Рисунок 2.7. – Діаграма класів системи

Укажемо відповідальність кожного з приведених на діаграмі класів:

*User (Користувач)* – людина яка взаємодіє з системою.

*Settings (Начальник)* – користувач, на якого є хоча б одне посилання у іншого користувача.

*Report (Звіт)* – звіт формується користувачем, та відсилається безпосередньому начальнику.

*Note (Повідомлення)* – користувачі можуть обмінюватися повідомленнями.

*Task (Завдання)* – завдання сформоване одним користувачем до іншого.

*Action (Дія)* – зміна статусу завдання (створення, відхилення, то що) проходить через створення користувачем "дії".

*Mailer* – клас який займається надсиланням листів.

*MailerException* – виключення екземпляр якого може виникнути при використанні класу *Mailer*.

*CharEncodingFilter* – фільтр який перетворює дані які приходять від *Http* запиту та відповіді.

*LookupUtil* – клас, що реалізує технологію *JNDI* для отримання екземплярів *SessionBean* класів у веб-модулі.

*AdminSession* – інтерфейс описуючий дії адміністратора системи.

*AdminSessionBean* – клас який його реалізує.

*UserSession* – інтерфейс описуючий дії користувача, як то створення звіту, відправлення повідомлення то що.

*UserSessionBean* – клас який його реалізує.

*TaskSession* – інтерфейс описуючий методи для роботи з завданнями.

*TaskSessionBean* – клас який його реалізує.

*NotFoundException* – виключення екземпляр якого створюється якщо не знайдений екземпляр у базі даних.

*AlreadyExistException* – виключення екземпляр якого створюється якщо не може бути створений об'єкт (користувач з тим логіном який вже використовується системою).

### **2.6.2. Логічна модель бази даних.**

Для забезпечення можливості зберігати дані в реляційній базі даних, з моделі класів була отримана логічна модель даних. Для цього серед усіх класів були визначені постійні. Це такі класи, об'єкти яких повинні зберігати свої значення не тільки під час роботи програми. Далі було проведено відображення постійних класів на таблиці моделі даних. При цьому атрибути класів відобразилися в поля реляційних таблиць. Асоціації між класами змінилися в зв'язки між реляційними таблицями з використанням механізму зовнішніх ключів. Для забезпечення можливості підтримки цілісності даних по суттєвостям були використані первинні ключі. Обрана СКБД не підтримує зберігання ієрархії спадкування класів, тому при будівництві логічної моделі даних ієрархії відображалися на таблиці за принципом одна таблиця на клас. Метод згортання типів в супертип при якому ієрархія змінюється в одну таблицю був відхилений із-за великої надмірності даних. Розроблена логічна модель наведена на рисунку 2.8.

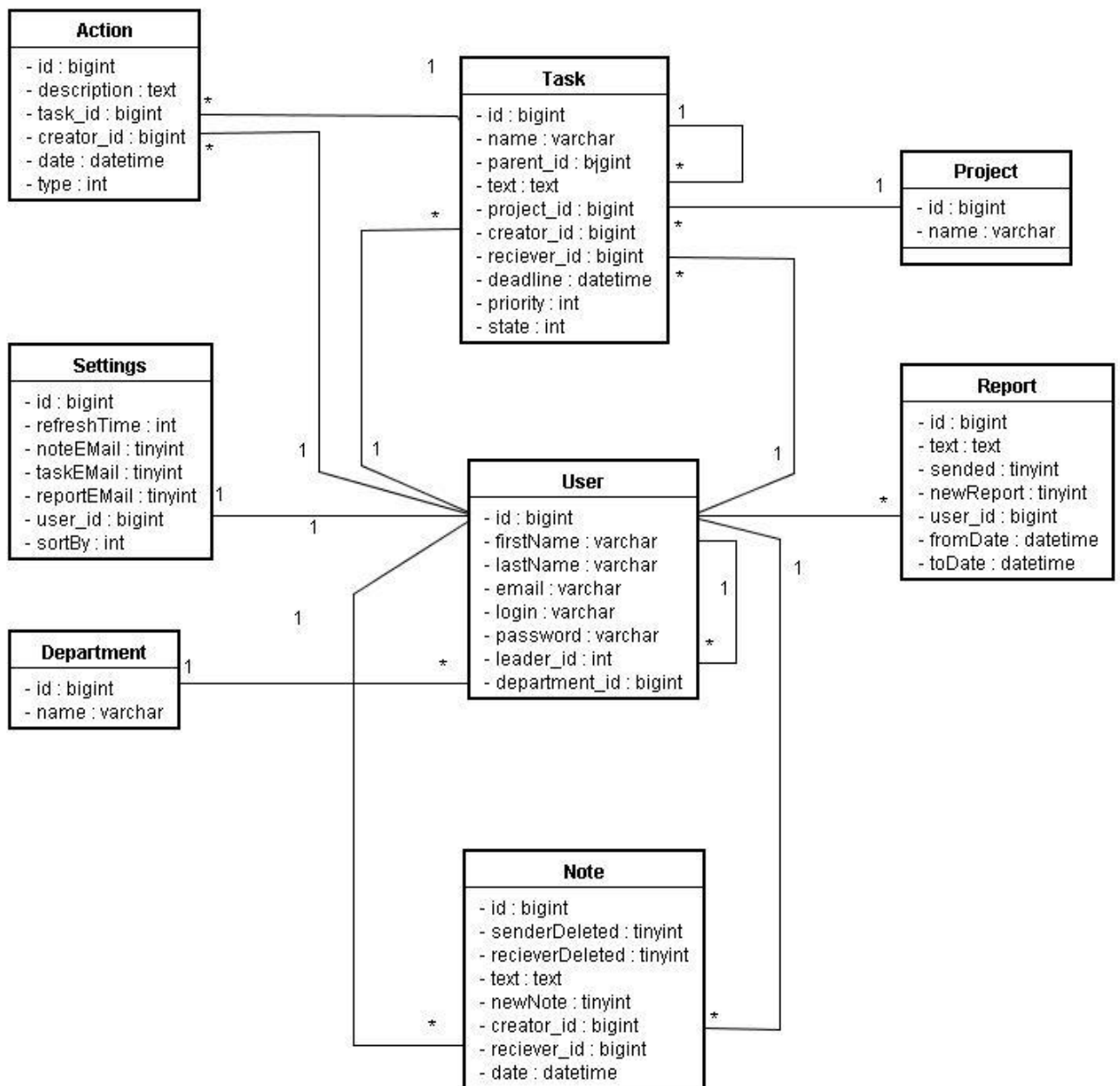


Рисунок 2.8. – Логічна модель бази даних.

### 2.6.3. Методи і засоби розробки програмного забезпечення.

При аналізі і проектуванні Системи застосовувався об'єктно-орієнтований підхід (ООП), виходячи з чого програмне забезпечення системи повинне бути розроблене з його використанням. Основними принципами ООП є спадкування, інкапсуляція і поліморфізм.

Використання ООП у розробці дозволяє:

- використовувати отримані в результаті аналізу і проектування логічні моделі для первинної кодогенерації;

- скоротити час розробки за рахунок повторного використання коду програмних компонентів;

- проводити налагодження і тестування ґрунтуючись на поведженні об'єктів, без знання їхньої внутрішньої структури;

- цілком використовувати потенціал обраних засобів розробки ПО.

Для проектування й аналізу Системи використовувалася CASE-система Rational Rose виробництва компанії Rational Software. На сьогоднішній день це один із самих могутніх CASE-засобів на ринку і де-факто стандарт галузі. Rational Rose використовується на всіх етапах проектування від аналізу до автоматичної генерації коду програмної системи і має засобу для роботи з всіма інструментами, наданими мовою UML.

Основним засобом розробки програмного забезпечення системи є мова програмування високого рівня Java, використовуються такі технології: Servlets, EJB3, JSPx та Struts. У якості середовища був використаний продукт Eclipse.

На Web-сервері повинен бути встановлений HTTP-сервер. Найбільш розповсюдженими HTTP-сервера, що задовольняють зазначеним вимогам і мають високу продуктивність, є Apache Server. На Web-сервері повинна бути встановлена ОС яка підтримує цей сервер.

Так як використовуються технології організовані на Java то операційна система на сервері доступу до БД може бути будь яка з перелічених:

- - Macintosh.

- - MS Windows.

- - Linux.

Також потрібний Java WEB сервер встановлений на сервері будь який з:

- JBoss.

- Sun Application Server.

- GlassFish.

- Tomcat.

На сервері баз даних повинна бути встановлена серверна система управління базами даних (СУБД).

Необхідна сучасна серверна реляційна СУБД, що має високу продуктивність, здатна працювати з великими обсягами інформації і підтримує стандарт SQL-92. Крім того, СУБД повинна працювати під управлінням Windows. Серверними СУБД, що задовольняють зазначеним вимогам, на сьогодні є:

- MS SQL Server.
- MySQL.
- PostgreSQL.
- Sybase.
- Oracle (Standard Edition чи Enterprise Edition).

На клієнтських машинах може використовуватися будь-яка операційна система з можливістю встановлення віддаленого мережного з'єднання і графічним інтерфейсом. Операційні системи, що рекомендуються – MS Windows, Linux, тощо.

Програмне забезпечення, з яким буде працювати користувач, являє собою Web-оглядач з підтримкою HTML, JavaScript, CSS (наприклад, MS Internet Explorer, FireFox, Opera ...).

### **3. Розробка робочої документації інформаційної системи управління завданнями на підприємстві JGSoft.**

Інформаційна система управління завданнями призначена для автоматизації роботи працівників підприємства на будь-якому рівні, від начальника відділу до працівника і не тільки. Функції, що автоматизуються системою, пов'язані з організацією робочого часу працівників отримання завдань та іншого документообігу, як то звіт та статистичні показники працівників.

Основними функціями, що автоматизуються є:

- Облік відділків та їх працівників.
- Облік завдань та інформацію про їхній статус.
- Формування звітів.

Усі інші вимоги не є головними і можуть змінюватися під час розробка або експлуатації.

При впровадженні програмного забезпечення системи основними цілями є:

- Простий і зручний інтерфейс користувача.
- Доступ до системи через Internet з будь-якого комп'ютера глобальної мережі.
- Багатокористувацька робота.
- Висока масштабуємість системи.
- Висока продуктивність.
- Надійність.
- Конфіденційність інформації.

Для ефективного використання програмного забезпечення важливо впровадити на підприємстві організаційні і технічні заходи, що передбачено в п. 2.2 та п. 2.4. Тому що підвищується рівень складності експлуатованого ПО, необхідно відповідне підвищення кваліфікації співробітників. При роботі з

програмним забезпеченням системи варто керуватися документами "Загальний опис системи управління завданнями" (Додаток А), "Керівництво користувача" (Додаток Б).

## **4. РОЗРАХУНОК ЕКОНОМІЧНОЇ ЕФЕКТИВНОСТІ ВПРОВАДЖЕННЯ ІНФОРМАЦІЙНОЇ СИСТЕМИ УПРАВЛІННЯ ЗАВДАННЯМИ.**

### **4.1 Вступ.**

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, що не під силу виконати одній людині. Сучасні засоби програмування дозволяють вирішувати 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 (управління, проектування, технологічної підготовки виробництва). 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 управління завданнями призначена для автоматизації роботи працівників підприємства на будь-якому рівні, від начальника відділу до працівника і не тільки. Функції, що автоматизуються системою, пов'язані з організацією робочого часу працівників отримання завдань та іншого документообігу, як то звіт та статистичні показники працівників.

Найбільш важливим моментом для розробника, з економічної точки зору, є процес формування вартості системи. Очевидно, що вона являє собою дуже специфічний товар з безліччю особливостей.

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 по діючим на момент розрахунку оптовим цінам, тарифам і ставкам заробітної плати.

#### 4.2 Розрахунок витрат на створення та експлуатацію ПЗ інформаційної 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 4,32 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
<b>Разом:</b>	<b>2800,00</b>

Вартість програми розраховуємо по формулі:

$$C_{np} = (z_{zn} + z_{cz} + z_{ze} + z_e) \cdot T + z_m, \quad (4.1)$$

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.

$Z_{zr}$  – 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 4,32 грн обчислимо:

$$Z_e = 132 \cdot 4,32 \cdot 0,5 = 285,12 \text{ грн.}$$

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  $Z_e$  при 1058 годинах роботи устаткування в рік складуть

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

Експлуатаційні витрати для ПК за рік складуть:

$$Z_{zp} = A_{ob} + B_m + \Phi_m = 14400 + 1200 + 888,72 = 16488,72 \text{ грн.}$$

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

$$Z_{ce} = C_{np} + Z_{zp} = 29250,88 + 16488,72 = 45739,60 \text{ грн.}$$

#### **4.3 Економічна ефективність розробки і впровадження системи.**

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  $\Delta C_n$  – 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).

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

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 – це система законодавчих актів, соціально-економічних, організаційних, технічних і лікувально-профілактичних заходів і засобів, що забезпечують безпеку, збереження здоров'я і працездатність людини в процесі праці.

Охорона праці виявляє і вивчає можливі причини виробничих нещасливих випадків, професійних захворювань, аварій, вибухів, пожеж і розробляє систему заходів і вимог з метою усунення цих причин, і створення безпечних і сприятливих для людини умов праці.

З питаннями охорони праці нерозривно пов'язане і вирішення питань охорони природи.

Складність задач, які стоять перед охороною праці, вимагає використання досягнень і висновків багатьох наукових дисциплін, прямо або побічно пов'язаних із задачами створення здорових і безпечних умов праці.

Оскільки головним об'єктом охорони праці є людина в процесі праці, то при розробці вимог виробничої санітарії використовуються результати досліджень ряду медичних і біологічних дисциплін.

Особливо тісний зв'язок існує між охороною праці, науковою організацією праці, ергономікою, інженерною психологією і технічною естетикою.

Успіх у вирішенні проблем охорони праці здебільшого залежить від якості підготовки фахівців у цій області, від їхнього уміння приймати правильні рішення в складних і мінливих умовах сучасного виробництва.

### **5.2 Аналіз небезпечних і шкідливих факторів у відділі інтерфейсного програмування підприємства JGSoft.**

При організації умов праці необхідно враховувати вплив на працюючих небезпечних і шкідливих виробничих факторів, які можуть привести до травми або іншого раптового різкого погіршення здоров'я та захворювання або зниження працездатності.

Небезпечні і шкідливі виробничі фактори (ДСТ 12.0.003-74) підрозділяються по природі дії на чотири групи: фізичні, хімічні, біологічні і психофізіологічні.

*До небезпечних фізичних факторів відносяться:* машини і механізми, що рухаються; різні підйомно-транспортні пристрої і переміщувані вантажі; незахищені рухливі елементи виробничого устаткування (приводні і передавальні механізми, різальні інструменти, пристосування, що обертаються і переміщуються й ін.); відлітаючі частки оброблюваного матеріалу та інструменту, електричний струм, підвищена температура поверхонь устаткування й оброблюваних матеріалів і т.д.

*Шкідливими для здоров'я фізичними факторами є:* підвищена чи знижена температура повітря робочої зони; висока вологість і швидкість руху повітря; підвищені рівні шуму, вібрації, ультразвуку і різних випромінювань – теплових, іонізуючих, електромагнітних, інфрачервоних і ін. До шкідливих фізичних факторів відносяться також запыленість і загазованість повітря робочої зони; недостатня освітленість робочих місць, проходів і проїздів; підвищена яскравість світла і пульсація світлового потоку.

*Хімічні небезпечні і шкідливі виробничі фактори за характером дії на організм людини підрозділяються на наступні підгрупи:* загальнотоксичні, подразнюючі, сенсibilізуючі (ті, що викликають алергійні захворювання), канцерогенні (ті, що викликають розвиток пухлин), мутагени (ті, що діють на статеві клітини організму). У цю групу входять численні пари і гази: пари бензолу і толуолу, окис вуглецю, сірчистий ангідрид, окисли азоту, аерозолі свинцю та ін., токсичні пили, що утворюються, наприклад, при обробці різанням берилію, свинцюватих бронз і латуней і деяких пластмас зі шкідливими наповнювачами. До цієї групи відносяться агресивні рідини (кислоти, луги), що можуть заподіяти хімічні опіки шкіряного покриву при зіткненні з ними.

*До біологічних небезпечних і шкідливих виробничих факторів відносяться* мікроорганізми (бактерії, віруси й ін.) та макроорганізми (рослини і тварини), вплив яких на працюючих викликає травми або захворювання.

*До психофізіологічних небезпечних і шкідливих виробничих факторів*

відносяться фізичні перевантаження (статичні та динамічні) і нервово-психічні перевантаження (розумові перенапруги, перенапруга аналізаторів слуху, зору та ін.).

Між шкідливими і небезпечними виробничими факторами спостерігається визначений взаємозв'язок. У багатьох випадках наявність шкідливих факторів сприяє прояву травмонебезпечних факторів. Наприклад, надмірна вологість у виробничому приміщенні і наявність струмопровідного пилу (шкідливі фактори) підвищують небезпеку ураження людини електричним струмом (небезпечний фактор).

Рівні впливу на працюючих шкідливих виробничих факторів нормовані граничнодопустимими рівнями, значення яких зазначені у відповідних стандартах системи стандартів безпеки праці і санітарно-гігієнічних правил.

Гранично припустиме значення шкідливого виробничого фактора (за ДСТ 12.0.002-80) – це граничне значення величини шкідливого виробничого фактора, вплив якого при щоденній регламентованій тривалості протягом усього виробничого стажу не приводить до зниження працездатності і захворювання як у період трудової діяльності, так і до захворювання в наступний період життя, а також несприятливо не впливає на здоров'я потомства.

Науково-технічний прогрес вніс серйозні зміни в умови виробничої діяльності працівників розумової праці. Їхня праця стала більш інтенсивною, напруженою, потребує значних витрат розумової, емоційної і фізичної енергії. Це зажадало комплексного рішення проблем ергономіки, гігієни й організації праці, регламентації режимів праці і відпочинку.

На даний час комп'ютерна техніка широко застосовується у всіх галузях діяльності людини. При роботі з комп'ютером людина піддається впливу ряду небезпечних і шкідливих виробничих факторів: електромагнітних полів (діапазон радіочастот: ВЧ, УВЧ і СВЧ), інфрачервоного та іонізуючого випромінювань, шуму і вібрації, статичної електрики та ін.

Робота з комп'ютером характеризується значною розумовою напругою і нервово-емоційним навантаженням операторів, високою напруженістю зорової роботи і досить великим навантаженням на м'язи рук при роботі з клавіатурою

ЕОМ. Велике значення має раціональна конструкція і розташування елементів робочого місця, що важливо для підтримки оптимальної робочої пози людини-оператора.

У процесі роботи з комп'ютером необхідно дотримуватися правильного режиму праці і відпочинку. В противному випадку в персонала відзначається значна напруга зорового апарату з появою скарг на незадоволеність роботою, головні болі, дратівливість, порушення сну, утому і хворобливі відчуття в очах, у попереку, в області шиї і руках.

Розглянемо шкідливі виробничі фактори, які спостерігаються в приміщеннях, де здійснюється робота з електронно-обчислювальною технікою.

### ***Освітлення.***

Правильно спроектоване і виконане виробниче освітлення поліпшує умови зорової роботи, знижує стомлюваність, сприяє підвищенню продуктивності праці, благотворно впливає на виробниче середовище, створюючи позитивний психологічний вплив на працюючого, підвищує безпеку праці і знижує травматизм.

Недостатність освітлення приводить до напруги зору, послабляє увагу, приводить до передчасної стомленості. Надмірно яскраве освітлення викликає осліплення, роздратування і різь в очах. Неправильний напрямок світла на робочому місці може створювати різкі тіні, відблиски, дезорієнтувати працюючого. Усі ці причини можуть привести до нещасливого випадку або профзахворювання, тому настільки важливий правильний розрахунок освітленості.

Існує три види освітлення – природне, штучне і сполучене (природне і штучне разом).

*Природне освітлення* – освітлення приміщень денним світлом, яке проникає через світлові прорізи в зовнішніх конструкціях приміщень. Природне освітлення характеризується тим, що міняється в широких межах в залежності від часу дня, пори року, характеру області і ряду інших факторів.

Штучне освітлення застосовується при роботі в темний час доби і вдень, коли не вдається забезпечити нормовані значення коефіцієнта природного

освітлення (похмура погода, короткий світловий день). Освітлення, при якому недостатнє по нормах природне освітлення доповнюється штучним, називається сполученим освітленням.

Штучне освітлення підрозділяється на робоче, аварійне, евакуаційне, охоронне. Робоче освітлення, в свою чергу, може бути загальним чи комбінованим. Загальне – освітлення, при якому світильники розміщуються у верхній зоні приміщення чи рівномірно стосовно до розташування устаткування. Комбіноване – освітлення, при якому до загального додається місцеве освітлення.

Згідно до Сніп П-4-79 у приміщеннях обчислювальних центрів необхідно застосувати систему комбінованого освітлення.

При виконанні робіт категорії високої зорової точності (найменший розмір об'єкта розрізнення 0,3...0,5 мм) величина коефіцієнта природного освітлення (КЕО) повинна бути не нижче 1,5%, а при зоровій роботі середньої точності (найменший розмір об'єкта розрізнення 0,5...1,0 мм) КЕО повинний бути не нижче 1,0%. В якості джерела штучного освітлення зазвичай використовуються люмінесцентні лампи типу ЛБ або ДРЛ, які попарно поєднуються у світильники, та повинні розташовуватися над робочими поверхнями рівномірно.

Вимоги до освітленості в приміщеннях, де встановлені комп'ютери, наступні: при виконанні зорових робіт високої точності загальна освітленість повинна складати 300 лк, а комбінована – 750 лк; аналогічні вимоги при виконанні робіт середньої точності – 200 і 300 лк відповідно.

Крім того усе поле зору повинно бути освітлене досить рівномірно – це основна гігієнічна вимога. Іншими словами, ступінь освітлення приміщення і яскравість екрана комп'ютера повинні бути приблизно однаковими, тому що яскраве світло в районі периферійного зору значно збільшує напруженість очей і, як наслідок, приводить до їх швидкої стомлюваності.

### ***Параметри мікроклімату.***

Параметри мікроклімату можуть мінятися в широких межах, в той час як необхідною умовою життєдіяльності людини є підтримка сталості температури тіла завдяки терморегуляції, тобто здатності організму регулювати віддачу

тепла в навколишнє середовище. Принцип нормування мікроклімату – створення оптимальних умов для теплообміну тіла людини з навколишнім середовищем.

Обчислювальна техніка є джерелом істотного тепловиділення, яке може привести до підвищення температури і зниження відносної вологості в приміщенні.

У приміщеннях, де встановлені комп'ютери, повинні дотримуватися визначені параметри мікроклімату. У санітарних нормах СН-245-71 встановлені величини параметрів мікроклімату, що створюють комфортні умови. Ці норми встановлюються в залежності від пори року, характеру трудового процесу і характеру виробничого приміщення ( таблиця 5.1).

Таблиця 5.1 – Параметри мікроклімату для приміщень, де встановлені комп'ютери

Період року	Параметр мікроклімату	Величина
Холодний	Температура повітря в приміщенні	22...24 °С
	Відносна вологість	40...60 %
	Швидкість руху повітря	до 0,1 м/с
Теплий	Температура повітря в приміщенні	23...25 °С
	Відносна вологість	40...60 %
	Швидкість руху повітря	0,1...0,2 м/с

### ***Шум і вібрація.***

Шумом називають усякий несприятливо діючий на людину звук. З фізичної точки зору звук являє собою механічні коливання пружного середовища.

Слуховий орган людини сприймає у вигляді чутного звуку коливання пружного середовища, які мають частоту приблизно від 20 до 20000 Гц, але найбільш важливий для слухового сприйняття інтервал від 45 до 10000 Гц.

Сприйняття людиною звуку залежить не тільки від його частоти, але і від інтенсивності і звукового тиску.

Несприятлива дія шуму на людину залежить не тільки від рівня звукового тиску, але і від частотного діапазону шуму, а також від рівномірності впливу

протягом робочого часу.

У результаті несприятливого впливу шуму на працюючу людину відбувається зниження продуктивності праці, збільшується брак у роботі, створюються передумови до виникнення нещасливих випадків. Усе це обумовлює велике оздоровче й економічне значення заходів щодо боротьби із шумом.

У таблиці 5.2 зазначені граничні рівні звуку в залежності від категорії ваги і напруженості праці, які є безпечними у відношенні збереження здоров'я і працездатності.

Таблиця 5.2 – Граничні рівні звуку, дБ, на робочих місцях

Категорія напруженості праці	Категорія важкості праці			
	I Легка	II Середня	III Важка	IV Дуже важка
I Мало напружена	80	80	75	75
II Помірковано напружена	70	70	65	65
III Напружена	60	60	–	–
IV Дуже напружена	50	50	–	–

### *Електромагнітне й іонізуюче випромінювання*

Більшість учених вважає, що як короткочасний, так і тривалий вплив усіх видів випромінювання від екрана монітора не небезпечний для здоров'я персоналу, що обслуговує комп'ютери. Однак вичерпних даних щодо безпеки впливу випромінювання від моніторів на працюючих з комп'ютерами не існує і дослідження в цьому напрямку продовжуються.

Припустимі значення параметрів неіонізуючих електромагнітних випромінювань від монітора комп'ютера представлені в таблиці 5.3.

Максимальний рівень рентгенівського випромінювання на робочому місці оператора комп'ютера зазвичай не перевищує 10 мкбер/год., а інтенсивність ультрафіолетового й інфрачервоного випромінювань від екрана монітора

знаходиться в межах 10...100 мВт/м<sup>2</sup>.

Таблиця 5.3 – Припустимі значення параметрів неіонізуючих електромагнітних випромінювань (відповідно до СанПіН 2.2.2.542-96)

Найменування параметра	Припустимі
Напруженість електричної складової електромагнітного поля або на відстані 50 см від поверхні відеомонітора	10 В/м
Напруженість магнітної складової електромагнітного поля або на відстані 50 см від поверхні відеомонітора	0,3 А/м
Напруженість електростатичного поля не повинна перевищувати:	
для дорослих користувачів	20 кв./м.
для дітей дошкільних установ і учнів середніх спеціальних і вищих навчальних закладів	15 кв./м.

### 5.3 Розрахунок системи штучного освітлення приміщення відділу інтерфейсного програмування.

Розрахунок освітленості робочого місця зводиться до вибору системи освітлення, визначення необхідного числа світильників, їхнього типу і розміщення. Виходячи з цього, розрахуємо параметри штучного освітлення.

Зазвичай штучне освітлення виконується за допомогою електричних джерел світла двох видів: ламп накаливання і люмінесцентних ламп. Будемо використовувати люмінесцентні лампи, які у порівнянні з лампами накаливання мають ряд істотних переваг:

- по спектральній сполучі світла вони близькі до денного, природного світла;
- володіють більш високим ККД (у 1,5-2 рази вище, ніж ККД ламп накаливання);
- мають підвищену світловіддачу (у 3-4 рази вище, ніж у ламп

накалювання);

– більш тривалий термін служби.

*Вхідні дані:*

Розрахунок освітлення проводиться для кімнати площею  $20 \text{ м}^2$ , ширина якої  $4 \text{ м}$ , висота –  $3 \text{ м}$ . Висота робочої поверхні  $1 \text{ м}$ . Для освітлення приймається світильник типу ППД2-500. Мінімальна освітленість лампи накалювання по нормах  $E_{\min} = 300 \text{ лк}$ . Коефіцієнт відображення стелі  $S_{\text{п}}=70\%$ , стін  $S_{\text{с}}=50\%$  і робочої поверхні  $S_{\text{р}}=30\%$ . Напруга мережі  $220\text{В}$ .

Скористаємося методом світлового потоку.

Для визначення кількості світильників визначимо світловий потік однієї лампи, який падає на поверхню по формулі:

$$F = \frac{E_{\min} \cdot K_3 \cdot S \cdot z}{N \cdot \eta}, \quad (5.1)$$

де  $F$  – світловий потік, що розраховується, лм;

$E_{\min}$  – нормована мінімальна освітленість, лк;

$S$  – площа освітлюваного приміщення (у нашому випадку  $S = 20 \text{ м}^2$ );

$z$  – відношення середньої освітленості до мінімального (звичайно приймається рівним  $1,1 \dots 1,2$ , нехай  $z = 1,1$ );

$K_3$  – коефіцієнт запасу, що враховує зменшення світлового потоку лампи в результаті забруднення світильників у процесі експлуатації (його значення залежить від типу приміщення і характеру проведених у ньому робіт і в нашому випадку  $K = 1,5$ );

$N$  – необхідна кількість світильників, яка розраховується по формулі 5.2:

$$N = \frac{S}{L^2} = \frac{S}{(1,5 \cdot h)^2} = \frac{20}{(2,25)^2} \approx 4, \quad (5.2)$$

де  $\eta$  – коефіцієнт використання, (виражається відношенням світлового потоку, що падає на розрахункову поверхню, до сумарного потоку всіх ламп і обчислюється в частках одиниці; залежить від характеристик світильника, розмірів приміщення, фарбування стін, стелі і робочої поверхні, які характеризуються відповідними коефіцієнтами відображення). Значення

коефіцієнтів приведені вище. Значення  $\eta$  визначимо по таблиці коефіцієнтів використання різних світильників. Для цього обчислимо індекс приміщення по формулі:

$$i = \frac{S}{h \cdot (A + B)}, \quad (5.3),$$

Де  $S$  – площа приміщення,  $S = 20 \text{ м}^2$ ;

$h$  – розрахункова висота підвісу,  $h = 1,5 \text{ м}$ ;

$A$  – ширина приміщення,  $A = 4 \text{ м}$ ;

$B$  – довжина приміщення,  $B = 5 \text{ м}$ .

Підставивши у формулу 5.3 значення одержимо:  $i = \frac{20}{1,5 \cdot (4 + 5)} \approx 1,48$ .

Знаючи індекс приміщення  $i$ , знаходимо  $\eta = 0,67$ .

Підставимо всі значення у формулу 5.1 для визначення світлового потоку  $F$ :

$$F = \frac{300 \cdot 1,5 \cdot 20 \cdot 1,1}{4 \cdot 0,67} \approx 3347 \text{ (лк)}.$$

Для освітлення вибираємо лампу типу Г125-135-200, потужністю 200 Вт, світловий потік якої  $F = 3200 \text{ лк}$ , найбільш близький до розрахункового.

При цьому фактична освітленість

$$F = E_{\min} \frac{F_n}{F_p} = 300 \frac{3200}{3347} \approx 287 \text{ (лк)}.$$

Загальна потужність  $P_{\text{общ}} = P_l \cdot N = 200 \cdot 4 = 800 \text{ (Вт)}$ .

#### **5.4 Розробка заходів щодо зменшення впливу шкідливих факторів на підприємстві JGSoft.**

Організація і поліпшення умов праці на робочому місці є одним з найважливіших резервів продуктивності праці й економічної ефективності виробництва, а також подальшого розвитку самої працюючої людини. У цьому головний прояв соціального й економічного значення організації і поліпшення умов праці.

Нижче приведені найбільш розповсюджені організаційні заходи і технічні засоби захисту людей, які працюють з комп'ютерами, від шкідливих виробничих факторів.

### ***Колірне оформлення і світлове відображення.***

Фарбування приміщень і меблів повинні сприяти створенню сприятливих умов для здорового сприйняття, гарного настрою.

Джерела світла, такі як світильники і вікна, що дають відображення від поверхні екрана, значно погіршують точність знаків і спричиняють перешкоди фізіологічного характеру, які можуть виразитися в значній напрузі, особливо при тривалій роботі. Відображення, включаючи відображення від вторинних джерел світла, повинне бути зведене до мінімуму. Для захисту від надлишкової яскравості вікон можуть бути застосовані штори й екрани.

У залежності від орієнтації вікон рекомендується наступна фарбування стін і підлоги:

- *вікна орієнтовані на південь*: стіни – зеленувато-блакитний або світло-блакитний кольори; підлога – зелений;
- *вікна орієнтовані на північ*: стіни – світло-жовтогарячий або оранжево-жовтий кольори; підлога – червонясто-жовтогарячий;
- *вікна орієнтовані на схід*: стіни – жовто-зеленого кольору; підлога – зелений або червонясто-жовтогарячий;
- *вікна орієнтовані на захід*: стіни – жовто-зелений або блакитнувато-зелений кольори; підлога – зелений чи червонясто-жовтогарячий.

У приміщеннях, де знаходиться комп'ютер, необхідно забезпечити наступні величини коефіцієнта відображення: для стелі – 60...70%, для стін – 40...50%, для підлоги – близько 30%. Для інших поверхонь і робочих меблів – 30...40%.

### ***Шум і вібрація.***

Зниження шуму, створюваного на робочих місцях внутрішніми джерелами, а також шуму проникаючого ззовні, є дуже важливою задачею. Зниження шуму в джерелі надходження можна забезпечити застосуванням пружних прокладок між підставою машини, приладу й опорною поверхнею. В якості прокладки використовуються гума, повсть, пробка, різної конструкції амортизатори. Під

настільні шумливі апарати можна підкладати м'які коврики із синтетичних матеріалів, а під ніжки столів, на яких вони встановлені, – прокладки з м'якої гуми, повсті, товщиною 6-8 мм. Кріплення прокладок можливе шляхом приклейки їх до опорних частин.

Можливе також застосування звукоізолюючих кожухів, що не заважають технологічному процесу. Не менш важливим для зниження шуму в процесі експлуатації є питання правильного і своєчасного регулювання, змазування і заміни механічних вузлів шумливого устаткування.

Рациональне планування приміщення і розміщення устаткування є важливим чинником, що дозволяє знизити шум при існуючому устаткуванні ЕОМ. Приміщення необхідно розташовувати в далечині від шумливого і вібруючого устаткування.

Зниження рівня шуму, що проникає у виробниче приміщення ззовні, може бути досягнуте збільшенням звукоізоляції обгороджуючих конструкцій, ущільненням по периметру притворів вікон, дверей.

Таким чином для зниження шуму, створюваного на робочих місцях внутрішніми джерелами, а також шуму, що проникає з поза, слід:

- послабити шум самих джерел (застосування екранів, звукоізолюючих кожухів);
- знизити ефект сумарного впливу відбитих звукових хвиль (звуковбирні поверхні конструкцій);
- застосовувати раціональне розташування устаткування;
- використовувати архітектурно-планувальні і технологічні рішення ізоляції джерел шуму.

### ***Мікроклімат***

Нормалізація мікроклімату приміщення може бути досягнута поліпшенням вентиляції приміщення за рахунок витяжних вентиляторів і нормального функціонування вбудованої вентиляції будинку.

### ***Освітлення***

У приміщеннях з обчислювальною технікою, як правило, застосовується бічне природне освітлення. Робочі кімнати і кабінети повинні мати природне

освітлення. В інших приміщеннях допускається штучне освітлення.

Освітленість на поверхні столу в зоні розміщення робочого документа повинна бути 300-500 лк, також допускається установка світильників місцевого освітлення для підсвічування документів, але з такою умовою, щоб воно не створювало відблисків на поверхні екрана і не збільшувало освітленість екрана більш ніж на 300 лк.

В якості джерела світла при штучному висвітленні повинні застосовуватися переважно люмінесцентні лампи типу ЛБ. При порядкуванні відбитого освітлення в адміністративно-суспільних приміщеннях допускається застосування металогалогених ламп потужністю до 250 Вт. Допускається застосування ламп накаливання у світильниках місцевого освітлення.

Загальне освітлення варто виконувати у вигляді суцільних або переривчастих ліній світильників, розташованих збоку від робочих місць, паралельно лінії зору користувача при рядному розташуванні ПЕОМ. При периметральному розташуванні комп'ютерів лінії світильників повинні розташовуватися локалізовано над робочим столом, ближче до його переднього краю, зверненого до оператора.

Для забезпечення нормованих значень освітленості в приміщеннях використання ПЕОМ варто проводити чищення стікол віконних рам і світильників не рідше двох разів у рік і проводити своєчасну заміну перегорілих ламп.

### ***Ергономіка робочого місця.***

Проектування робочих місць, оснащених відеотерміналами, відноситься до числа важливих проблем ергономічного проектування в області обчислювальної техніки.

Робоче місце і взаємне розташування всіх його елементів повинне відповідати антропометричним, фізичним і психологічним вимогам. Велике значення має також характер роботи. Зокрема, при організації робочого місця програміста повинні бути дотримані наступні основні умови: оптимальне розміщення устаткування, що входить до складу робочого місця і достатній робочий простір, що дозволяє здійснювати всі необхідні рухи і переміщення.

Ергономічними аспектами проектування відеотермінальних робочих місць, зокрема, є: висота робочої поверхні, розміри простору для ніг, вимоги до розташування документів на робочому місці (наявність і розміри підставки для документів, можливість різного розміщення документів, відстань від очей користувача до екрана, документа, клавіатури і т.д.), характеристики робочого крісла, вимоги до поверхні робочого столу, регульованість елементів робочого місця.

Головними елементами робочого місця програміста є стіл і крісло. Основним робочим положенням є положення сидячи.

Робоча поза сидячи викликає мінімальне стомлення програміста. Рациональне планування робочого місця передбачає чіткий порядок і сталість розміщення предметів, засобів праці і документації. Те, що потрібно для виконання робіт частіше, розташовується в зоні легкої досяжності робочого простору.

Для комфортної роботи стіл повинний задовольняти наступним умовам:

- висота столу повинна бути обрана з урахуванням можливості сидіти вільно, у зручній позі, при необхідності спираючись на підлокітники;
- нижня частина столу повинна бути сконструйована так, щоб програміст міг зручно сидіти, не був змушений підгортати ноги;
- поверхня столу повинна мати властивості, що виключають появу відблисків у полі зору програміста;
- конструкція столу повинна передбачати наявність висувних шухляд (не менше ніж 3 для збереження документації, лістингів, канцелярських предметів);
- висота робочої поверхні рекомендується в межах 680-760 мм. Висота поверхні, на яку встановлюється клавіатура, повинна бути близько 650 мм.

Велике значення надається характеристикам робочого крісла. Так, рекомендована висота сидіння над рівнем підлоги знаходиться в межах 420-550 мм. Поверхня сидіння м'яка, передній край закруглений, а кут нахилу спинки – регульований.

Необхідно передбачати при проектуванні можливість різного розміщення

документів: збоку від відеотерміналу, між монітором і клавіатурою і т.п. Крім того, у випадках, коли відеотермінал має низьку якість зображення, наприклад помітні мелькання, відстань від очей до екрана роблять більше (близько 700 мм), чим відстань від ока до документа (300-450 мм). Узагалі при високій якості зображення на відеотерміналі відстань від очей користувача до екрана, документа і клавіатури може бути рівною.

Велике значення також надається правильній робочій позі користувача. При незручній робочій позі можуть з'явитися болі в м'язах, суглобах і сухожиллях. Вимоги до робочої пози користувача відеотерміналу наступні:

- голова не повинна бути нахилена більш ніж на  $20^{\circ}$  см.;
- плечі повинні бути розслаблені;
- лікті – під кутом  $80^{\circ}\dots 100^{\circ}$ ;
- передпліччя і кисті рук – у горизонтальному положенні.

Причина неправильної пози користувачів обумовлена наступними факторами: немає гарної підставки для документів, клавіатура знаходиться занадто високо, а документи – низько, нікуди покласти руки і кисті, недостатній простір для ніг.

З метою подолання зазначених недоліків даються загальні рекомендації: краще пересувна клавіатура; повинні бути передбачені спеціальні пристосування для регулювання висоти столу, клавіатури й екрана, а також підставка для рук.

Істотне значення для продуктивної і якісної роботи на комп'ютері мають розміри знаків, щільність їхнього розміщення, контраст і співвідношення яркостей символів і фону екрана. Якщо відстань від очей оператора до екрана дисплея складає 60...80 см, то висота знака повинна бути не менше ніж 3 мм, оптимальне співвідношення ширини і висоти знака складає 3:4, а відстань між знаками – 15...20% їхньої висоти. Співвідношення яскравості фону екрана і символів – від 1:2 до 1:15.

Під час користування комп'ютером медики радять установлювати монітор на відстані 50-60 см від очей. Фахівці також вважають, що верхня частина

відеодисплея повинна бути на рівні очей або трохи нижче. Коли людина дивиться прямо перед собою, її очі відкриваються ширше, ніж коли вона дивиться вниз. За рахунок цього, площа огляду значно збільшується, викликаючи зневоднювання очей. До того ж якщо екран установлений високо, а очі широко відкриті, порушується функція моргання. Це значить, що очі не закриваються цілком, не омиваються слізною рідиною, не одержують достатнього зволоження, що приводить до їх швидкої стомлюваності.

### ***Режим праці.***

Для підтримки тривалої працездатності людини велике значення має режим праці і відпочинку. Під раціональним фізіологічно обґрунтованим режимом праці і відпочинком мається на увазі таке чергування періодів роботи з періодом відпочинку, при якому досягається висока ефективність суспільно-корисної діяльності людини, гарний стан здоров'я, високий рівень працездатності і продуктивності праці.

У таблиці 5.4 представлені зведення про регламентовані перерви, які необхідно робити при роботі на комп'ютері, в залежності від тривалості робочої зміни, видів і категорій трудової діяльності з ВДТ (відеодисплейний термінал) і ПЕОМ (відповідно до СанПіН 2.2.2 542-96 "Гігієнічні вимоги до відеодисплейних терміналів, персональних електронно-обчислювальних машин і організації робіт").

Таблиця 5.4 – Час регламентованих перерв при роботі на комп'ютері

<b>Категорія роботи з ВДТ чи ПЕОМ</b>	<b>Рівень навантаження за робочу зміну при видах роботи з ВДТ</b>			<b>Сумарний час регламентованих перерв, хв</b>	
	<b>Група А, кількість знаків</b>	<b>Група Б, кількість знаків</b>	<b>Група В, годин</b>	<b>При 8-годинній зміні</b>	<b>При 12-годинній зміні</b>
I	до 20 000	до 15 000	до 2,0	30	70
II	до 40 000	до 30 000	до 4,0	50	90
III	до 60 000	до 40 000	до 6,0	70	120

*Примітка.* Час перерв даний при дотриманні зазначених Санітарних

правил і норм. При невідповідності фактичних умов праці вимогам Санітарних правил і норм час регламентованих перерв варто збільшити на 30%.

У відповідності із СанПіН 2.2.2 546-96 усі види трудової діяльності, пов'язані з використанням комп'ютера, розділяються на три групи:

*група А:* робота зі зчитування інформації з екрана ВДТ чи ПОЕМ із попереднім запитом;

*група Б:* робота з введення інформації;

*група В:* творча робота в режимі діалогу з ЕОМ.

Ефективність перерв підвищується при сполученні з виробничою гімнастикою або організацією спеціального приміщення для відпочинку персоналу зі зручними м'якими меблями, акваріумом, зеленою зоною і т.п.

Створення сприятливих умов праці і правильне естетичне оформлення робочих місць на виробництві має велике значення як для полегшення праці, так і для підвищення його привабливості, що позитивно впливає на продуктивність праці.

## **6. ОХОРОНА НАВКОЛИШНЬОГО СЕРЕДОВИЩА.**

### **6.1 Вступ.**

У наш час захист навколишнього середовища висувається на перший план. Наслідки недостатньої уваги до проблеми можуть бути катастрофічними. Мова йде не тільки про добробут людства, а про його виживання. Особливо тривожно те, що деградація природного середовища може виявитися необоротною.

Під охороною навколишнього середовища розуміють сукупність міжнародних, державних і регіональних правових актів, інструкцій і стандартів, що доводять загальні юридичні вимоги до кожного конкретного забруднювача та, які забезпечують його зацікавленість у виконанні цих вимог, конкретних природоохоронних заходів щодо перетворення в життя цих вимог.

Тільки якщо всі ці складові частини відповідають один одному за змістом і темпами розвитку, тобто складаються в єдину систему охорони навколишнього природного середовища, можна розраховувати на успіх.

Оскільки не була вирішена вчасно задача охорони природи від негативного впливу людини, тепер все частіше постає задача захисту людини від впливу змінившогося природного середовища. Обидва ці поняття інтегруються в термін "охорона навколишнього (людини) природного середовища"

Охорона навколишнього природного середовища складається з:

- правової охорони, що формулює наукові екологічні принципи у вигляді юридичних законів, обов'язкових для виконання;
- матеріального стимулювання природоохоронної діяльності, що прагне зробити її економічно вигідною для підприємств;
- інженерної охорони, що розробляє природоохоронну і ресурсозбережну технологію та техніку.

### **6.2 Забруднення атмосферного повітря автотранспортом.**

У результаті господарської діяльності людини в атмосфері з'являється велика кількість забруднюючих речовин. Взаємодія атмосферного повітря з водою і ґрунтом приводить до якісних і кількісних змін усієї біосфери в цілому,

підсилюючи і прискорюючи небажані зміни складу і структури атмосферного повітря, клімату Землі. Найбільш сильні зміни клімату і якості атмосферного повітря спостерігаються у великих містах. Якщо кисень в атмосферному повітрі буде дуже сильно забруднений усілякими речовинами, то поступово у всього живого на Землі буде скорочуватися термін життя, доки не зменшиться до винищення всіх і всього.

Різноманітне втручання людини в природні процеси в біосфері можна згрупувати за наступними видами забруднень, розуміючи під ними будь-які небажані для екосистем антропогенні зміни:

- інгредієнтне (інгредієнт – складова частина складного з'єднання або суміші) забруднення як сукупність речовин, кількісно або якісно далеких природнім біогеоценозам;

- параметричне забруднення (параметр навколишнього середовища – одне з його властивостей, наприклад рівень шуму, освітленості, радіації і т.д.), пов'язане зі зміною якісних параметрів навколишнього середовища;

- біоценотичне забруднення, що полягає у впливі на склад і структуру популяції живих організмів;

- стаціонально-деструкційне забруднення (стація – місце проживання популяції, деструкція – руйнування), яке являє собою зміну ландшафтів і екологічних систем у процесі природокористування.

Стрімкий розвиток усіх галузей промисловості, енергетики, транспорту, збільшення чисельності населення й урбанізація, хімізація всіх сфер діяльності людини привели до визначених змін навколишнього природного середовища, у тому числі несприятливих, що полягають головним чином у забрудненні біосфери. Вплив шкідливих речовин антропогенного походження на природне середовище, а так само відгук середовища на ці впливи стають глобальними, всеосяжними. Тому питання спостережень, охорони і контролю природного середовища в умовах науково-технічної революції є складеною і невід'ємною частиною соціального розвитку суспільства.

В нашій країні добре усвідомлюється небезпека можливих негативних наслідків впливу людини на природу (хоча ми і не розділяємо думки про

неминучість настання екологічної кризи) і приймаються дійові заходи по регулюванню взаємодії людини з навколишнім середовищем. Звичайно, рішення цієї проблеми полягає не в обмеженні розвитку людського суспільства, а в оптимізації його відносин із природою, у розумному перетворенні природи, раціональному використанні її ресурсів в інтересах нинішнього і майбутнього поколінь.

До мобільних джерел забруднення відносяться автомобілі і транспортні механізми, що пересуваються по землі, по воді і по повітрю. У великих містах до числа основних джерел забруднення атмосферного повітря відноситься *автотранспорт*. Гази двигунів, що відходять, містять складну суміш, їх більш ніж двісті компонентів, серед яких чимало канцерогенів. Наземні транспортні засоби – це механізми, що пересуваються по шосейним і залізничним дорогам, а також будівельне, сільськогосподарське і військове устаткування.

Шкідливі речовини при експлуатації рухливих транспортних засобів надходять у повітря з відпрацьованими газами, випарами з паливних систем і при заправленні, а також з картерними газами. На викиди оксиду вуглецю значний вплив має рельєф дороги і режим руху автомашини. Так, наприклад, при прискоренні і гальмуванні у відпрацьованих газах збільшується зміст оксиду вуглецю майже в 8 разів. Мінімальна кількість оксиду вуглецю виділяється при рівномірній швидкості автомобіля 60 км/г.

У таблиці 6.1 наведені значення концентрації основних домішок карбюраторного двигуна при різних режимах його роботи.

Таблиця 6.1 – Концентрація речовин у залежності від режиму роботи карбюраторного двигуна

Режим роботи двигуна	Оксид вуглецю, % по об'єму	Вуглеводні, мг/л	Оксиди азоту, мг/л
Холостий хід	4-12	2-6	—
Примусовий холостий хід	2-4	8-12	—
Середні навантаження	0-1	0,8-1,5	2,5-4,0
Повні навантаження	2	0,7-0,8	4-8

Викиди оксидів азоту максимальні при відношенні повітря – паливо 16:1. Таким чином, значення викидів шкідливих речовин у відпрацьованих газах

автотранспорту залежать від цілого ряду факторів: відношення в суміші повітря і палива, режимів руху автотранспорту, рельєфу і якості доріг, технічного стану автотранспорту й ін. Склад і обсяги викидів залежать також від типу двигуна. У таблиці 6.2 показані викиди ряду шкідливих речовин карбюраторного і дизельного двигунів.

Таблиця 6.2 – Викиди (% по об'єму) речовин при роботі дизельних і карбюраторних двигунів

РЕЧОВИНА	ДВИГУН	
	Карбюраторний	Дизельний
Оксид вуглецю	0,5-12,0	0,01-0,5
Оксид азоту	0,005-0,8	0,002-0,5
Вуглеводні	0,2-0,3	0,009-0,5
Бенз(а)пирен	До 20 мкг/м <sup>3</sup>	До 10 мкг/м <sup>3</sup>

Як видно з даних таблиці 6.2, викиди основних забруднюючих речовин значно нижче в дизельних двигунах. Тому прийнято вважати їх більш екологічно чистими. Однак дизельні двигуни відрізняються підвищеними викидами сажі, яка утворюється внаслідок перевантаження палива. Сажа насичена канцерогенними вуглеводнями і мікроелементами; їх викиди в атмосферу неприпустимі.

У зв'язку з тим, що відпрацьовані гази автомобілів надходять у нижній шар атмосфери, а процес їхнього розсіювання значно відрізняється від процесу розсіювання високих стаціонарних джерел, шкідливі речовини знаходяться практично в зоні подиху людини. Тому автомобільний транспорт варто віднести до категорії найбільш небезпечних джерел забруднення атмосферного повітря поблизу автомагістралей.

Відповідно до формули для середнього питомого викиду (коефіцієнта викиду)

$$E_m = \frac{\text{сумарний річний викид забруднюючих речовин}}{\text{сума річних транспортних показників}}$$

У таблиці 6.3 приведені ці величини для автомобільних викидів.

Таблиця 6.3 – Середні питомі викиди (коефіцієнти викидів) автотранспорту

Вид забруднюючої речовини	Середній питомий викид (при середній швидкості 31,7	
	За годину	На кілометр
Оксид вуглецю	752 г/год.	23,7 г/км.
Незгорілі вуглеводні	29,4 г/год.	0,93 г/км.
Оксиди азоту	33,2 г/год.	1,05 г/км.
Свинець	1,11 г/год.	0,035 г/км.
Сумарна кількість вихлопних газів (при 0° С)	28,95 м <sup>3</sup> /год.	0,914 м <sup>3</sup> /км.
Середня витрата палива	2,75 кг/год.	0,087 кг/км.

### 6.3 Заходи запобігання забруднення навколишнього середовища.

Природоохоронною є будь-яка діяльність, спрямована на збереження якості навколишнього середовища на рівні, що забезпечує стійкість біосфери. До неї відноситься як великомасштабна, здійснювана на загальнодержавному рівні діяльність по збереженню еталонних зразків недоторканої природи і збереженню розмаїтості видів на Землі, організації наукових досліджень, підготовці фахівців-екологів і вихованню населення, так і діяльність окремих підприємств по очищенню від шкідливих речовин стічних вод і відхідних газів, зниженню норм використання природних ресурсів і т.д. Така діяльність здійснюється в основному інженерними методами.

Існують два основних напрямки природоохоронної діяльності підприємств. Перший – очищення шкідливих викидів. Цей шлях «у чистому вигляді» малоефективний, тому що з його допомогою далеко не завжди вдається цілком припинити надходження шкідливих речовин у біосферу. До того ж скорочення рівня забруднення одного компонента навколишнього середовища веде до посилення рівня забруднення іншого.

І, наприклад, установка вологих фільтрів при газоочищенні дозволяє скоротити забруднення повітря, але веде до ще більшого забруднення води. Уловлені з відхідних газів і зливальних вод речовини часто отруюють значні земельні площі.

Використання очисних споруд, навіть найефективніших, різко скорочує рівень забруднення навколишнього середовища, однак не вирішує цієї проблеми цілком, оскільки в процесі функціонування цих установок теж виробляються

відходи, хоча й у меншому обсязі, але, як правило, з підвищеною концентрацією шкідливих речовин. Нарешті, робота більшої частини очисних споруджень вимагає значних енергетичних витрат, що, у свою чергу, теж небезпечно для навколишнього середовища.

Крім того, забруднювачі, на знешкодження яких йдуть величезні засоби, являють собою речовини, на які уже витрачена праця і які за рідкісним винятком можна було б використовувати в народному господарстві.

Для досягнення високих еколого-економічних результатів необхідно процес очищення шкідливих викидів сполучити з процесом утилізації уловлених речовин, що уможливить об'єднання першого напрямку з другим.

Другий напрямок – усунення самих причин забруднення, що вимагає розробки маловідходних, а в перспективі і безвідхідних технологій виробництва, які дозволяли б комплексно використовувати вихідну сировину й утилізувати максимум шкідливих для біосфери речовин.

Однак далеко не для усіх виробництв знайдені прийнятні техніко-економічні рішення по різкому скороченню кількості відходів, що утворюються, і їх утилізації, тому на даний час доводиться працювати в обох зазначених напрямках.

Піклуючись про удосконалювання інженерної охорони навколишньої природного середовища, треба пам'ятати, що ніякі очисні спорудження і безвідхідні технології не зможуть відновити стійкість біосфери, якщо будуть перевищені припустимі (граничні) значення скорочення природних, не перетворених людиною природних систем, у чому виявляється чинність закону незамінності біосфери.

Таким порогом може виявитися використання більш 1% енергетики біосфери і глибоке перетворення більш 10% природних територій (правила одного і десяти відсотків). Тому технічні досягнення не знімають необхідності рішення проблем зміни пріоритетів суспільного розвитку, стабілізації народонаселення, створення достатнього числа заповідних територій і інших, розглянутих раніше.

*Основні заходи щодо зниження забруднення атмосферного повітря викидами автомобільного транспорту.*

Основними напрямками робіт в галузі захисту атмосфери від забруднення викидами автотранспорту є:

- а) створення і розширення виробництва автомобілів з високоекономічними і малотоксичними двигунами, у тому числі подальша дизелізація автомобілів;
- б) розвиток робіт зі створення і впровадження ефективних систем нейтралізації відпрацьованих газів;
- в) зниження токсичності моторних палив;
- г) розвиток робіт з раціональної організації руху автотранспорту в містах, удосконалюванню дорожнього будівництва з метою забезпечення невинного руху на автомагістралях.

До труднощів очищення газів від забруднювачів відноситься в першу чергу те, що обсяги промислових газів, що викидаються в атмосферу, величезні.

Наприклад, велика теплоелектроцентральною здатна в одну годину викинути в атмосферу до 1 млрд. куб. метрів газів. Тому навіть при дуже високому ступені очищення відхідних газів, кількість забруднюючої речовини, що надходить у повітряний басейн, буде оцінюватися значною величиною.

Крім того, немає єдиного універсального методу очищення для всіх забруднювачів. Ефективний метод очищення відхідних газів від однієї забруднюючої речовини може виявитися марним стосовно інших забруднювачів. Чи метод, що добре виправдав себе в конкретних умовах (наприклад, у строго обмежених межах зміни концентрації або температури), в інших умовах виявляється малоефективним. З цієї причини доводиться використовувати комбіновані методи, сполучати кілька способів одночасно. Усе це визначає високу вартість очисних споруд, знижує їхню надійність при експлуатації.

Шкідливі домішки в відхідних газах можуть бути представлені або у вигляді аерозолів, або в газоподібному чи пароподібному стані. У першому випадку завдання очищення полягає у вилученні зважених твердих і рідких домішок, що містяться в промислових газах, – пилу, диму, крапельок туману і

бризів. У другому випадку – нейтралізація газо- і пароподібних домішок.

Очищення від аерозолів здійснюється застосуванням електрофільтрів, методів фільтрації через різні пористі матеріали, гравітаційної чи інерційної сепарації, способами мокрого очищення.

Очищення викидів від газо- і пароподібних домішок здійснюються методами адсорбції, абсорбції і хімічних методів.

Адсорбція – це процес поглинання газу чи пари поверхнею твердих тіл (адсорбентів) – силікагелю, активованого вугілля й інших. У випадку низької концентрації і правильного підбору адсорбенту цей метод дозволяє витягати будь-яку домішку з високим ступенем очищення, що досягає 99 %. Адсорбенти використовуються у вигляді зерен розміром 2-8 мм або в пилоподібному стані. Забруднений газ пропускається через шар адсорбенту.

Абсорбційний спосіб очищення заснований на різній розчинності компонентів газової суміші в рідині – абсорбенті. В якості абсорбентів, що використовуються для очищення газових викидів, застосовуються вода, розчини лугів, етаноламіни й інші рідини. Достоїнством абсорбційного очищення є, насамперед, високий рівень очищення, безперервність процесу, можливість витягу великої кількості домішок і можливість регенерації абсорбенту, а до недоліків – громіздкість устаткування, складність технологічних схем очищення.

Хімічні методи очищення газоподібних відходів полягають в тому, що до промислових відхідних газів додають різні реагенти, що вступають у хімічні реакції з домішками. Іноді цими реагентами можуть служити компоненти самих забруднювачів, а реакції підтримуються застосуванням каталізаторів. В результаті взаємодії утворюються нові з'єднання, які не роблять негативного впливу на природу.

Основне достоїнство хімічних методів очищення – високий рівень очищення.

Одним із видів хімічних методів може служити термічне очищення – допалювання відроблених газів. При високих температурах відбувається окислювання токсичних органічних забруднень киснем повітря до нетоксичних

з'єднань. Допалювання органічних домішок у газах промислових викидів і транспорту застосовують в основному в тих випадках, коли їхня утилізація недоцільна або неможлива.

#### **6.4. Висновки.**

Охорона природи – задача нашого століття, проблема, що стала соціальною. Знову і знову ми чуємо про небезпеку, що загрожує навколишньому середовищу, але і дотепер багато хто з нас вважає її неприємним, але неминучим породженням цивілізації, і думає, що ми ще встигнемо зправитися з всіма виникаючими утрудненнями.

Однак вплив людини на навколишнє середовище прийняло загрозливі масштаби. Щоб у корені поліпшити стан, знадобляться цілеспрямовані і продумані дії. Відповідальна і діюча політика стосовно навколишнього середовища буде можлива лише в тому випадку, якщо ми нагромадимо надійні дані про сучасний стан середовища, обґрунтовані знання про взаємодію важливих екологічних факторів, якщо розробимо нові методи зменшення і запобігання шкоди, що наносить Природі Людина.

## ВИСНОВКИ

У результаті виконання даної дипломної роботи було розроблено проект системи управління завданнями на підприємстві JGSoft.

Для досягнення цієї мети були вирішені наступні задачі:

- Досліджено технології проектування розподілених систем.
- Сформульовані вимоги до автоматизованої системи управління.
- Розроблені концепції системи, та проведено вибір оптимального варіанту методом функціонально-вартісного аналізу.
- Розроблені проектні рішення по системі з різних видів забезпечення, передбачених етапами ескізного та технічного проектів, зокрема загальносистемні рішення, рішення з організаційного забезпечення, рішення з інформаційного забезпечення, рішення з технічного забезпечення, рішення з програмного забезпечення.
- Розроблена робоча документація по системі.
- Розрахована економічна ефективність від впровадження автоматизованої системи.

Отримані результати дозволяють зробити висновок, що у разі успішної реалізації та впровадження системи управління, підвищиться ефективність управління виробництвом, що призведе до загального підвищення ефективності роботи та конкурентоспроможності підприємства. Також це дозволить підвищити рівень організації і досвіду робітників у використанні інформаційних технологій, що підготує умови для наступного етапу інформатизації підприємства.

## СПИСОК ВИКОРИСТАНОЇ ЛІТЕРАТУРИ

1. Тубальцев А.М., Виробниче освітлення та його розрахунок., Миколаїв:УДМТУ, 2001.
2. Шевченко, В. М. (2018). Проектування інформаційних систем – Київ: Вид-во "Наукова думка".
3. Руденко, В. М. (2016). Основи проектування інформаційних систем – Харків: Видавництво "Фоліо".
4. Бенкс, К. Л. (2005). System Analysis and Design – Pearson Prentice Hall.
5. Bourgeois, D., & Wasserman, A. I. (2010). Software Engineering: A Practitioner's Approach – McGraw-Hill.
5. Sommerville, I. (2011). Software Engineering (9th ed.). Boston: Addison-Wesley. This book covers a broad range of topics on software development methodologies, which are crucial for information systems development.
6. O'Brien, J.A., & Marakas, G.M. (2011). Management Information Systems (10th ed.). New York: McGraw-Hill. This textbook gives insights into the role of information systems in organizations.
7. Bill Burke, Richard Monson-Haefel. Enterprise JavaBeans, 3.0, O'Reilly, 2006.
8. Bruce W. Perry, Java Servlet & JSP Cookbook, O'Reilly, 2004.
9. Jennifer Ball, Debbie Carson, Ian Evans, Scott Fordin, Kim Haase, Eric Jendrock, The Java EE 5 Tutorial, Sun Microsystems, 2006.
10. Scott Davis, Tom Marrs, JBoss at Work: A Practical Guide, O'Reilly, 2005
11. William Crawford, Jim Farley, William Crawford, Jim Farley, O'Reilly, 2005.

## **Додаток А. – Загальний опис системи управління завданнями на підприємстві JGSoft.**

### **1. Загальні відомості.**

"Системи управління завданнями на підприємстві JGSoft", надалі Система. Для функціонування Системи потрібно наступне програмне забезпечення:

- SUSE Linux.
- Apache.
- MySQL.
- JBoss.

Програмне забезпечення Системи, що представляє дистрибутив зі .ear файлу, бібліотек, шаблонів і т.п., а також схеми таблиць бази даних.

Програмне забезпечення Системи реалізоване мовою програмування Java з використанням середовища розробки Eclipse SDK, програмне забезпечення серверної частини реалізовано у виді об'єктів бази даних (збережених процедур, тригерів і т.п.) з використанням SQL-діалекту MySQL.

### **2. Функціональне призначення.**

Система являє собою інструмент управління документообігу на підприємстві. Основними функціями системи є:

- Облік відділків та їх працівників.
- Облік завдань та інформацію про їхній статус.
- Формування звітів.
- Можливість обміну повідомленнями між працівниками.

### **3. Опис логічної структури.**

При розробці програмного забезпечення Системи використані методи об'єктно-орієнтованого проектування і розробки. Основною структурною одиницею ПО є клас, діаграма класів Системи приведена в проектній документації. Для побудови Системи використана архітектура "клієнт-сервер",

основними частинами програмного забезпечення є клієнтська і серверна частина, як правило, розташовані на різних комп'ютерах у мережі підприємства. Клієнтською частиною є веб-оглядач який має бути встановлений на комп'ютерах користувачів системи. Система виконує функції збереження, підтримки цілісності даних їхньої основної обробки шляхом використання об'єктів БД при доступі до даних.

#### **4. Використовувані технічні засоби.**

Для функціонування програмного забезпечення Системи використовуються:

- робочі місця користувачів, що представляють собою персональні комп'ютери (ПК) з мінімальною конфігурацією із встановленим веб оглядачем; мінімальний обсяг вільного місця на твердому диску 20 Гб;
- сервер SmartServer SE100R-2U;
- локальна мережа для надання з робочих місць користувачів доступу до даних, що знаходиться в БД сервера, із пропускною здатністю 10/100Мбіт/с з використанням комутатора Allied Telesyn AT-8024 (24 порти).

#### **5. Виклик і завантаження.**

Виклик і завантаження Системи виробляється з клієнтського робочого місця запуском веб оглядача та введенням адреси системи в мережі, після чого відбувається безпосередньо вхід у головне вікно системи з запитом імені користувача і пароля.

Завантаження Системи проходить як копіювання ear-архіву до Java серверу.

## **Додаток Б. – Інструкція користувача автоматизованої системи управління завданнями.**

### **1. Призначення програми.**

Програмне забезпечення інформаційної системи управління завданнями, яка призначена для автоматизації роботи працівників підприємства на будь-якому рівні, від начальника відділу до працівника і не тільки. Функції, що автоматизуються системою, пов'язані з організацією робочого часу працівників отримання завдань та іншого документообігу, як то звіт та статистичні показники працівників.

Задачею інформаційної системи, є автоматизація наступних функцій користувача:

- Облік відділків та їх працівників.
- Облік завдань та інформацію про їхній статус.
- Формування звітів.

Для створення інформаційної системи необхідно розробити єдину базу даних для збереження інформації про відділи та їх співробітників, про отриманні ними завдання і інше. Також повинно бути передбачене програмне забезпечення для адміністрування системи. ПЗ функціонує в мережі підприємства, обмін даними будується через єдину інформаційну базу.

### **2. Умови виконання програми.**

Для виконання програми необхідна наявність на підприємстві наступних апаратних і програмних засобів:

- робочі місця користувачів, що представляють собою персональні комп'ютери з мінімальною конфігурацією із встановленим веб оглядачем; мінімальний обсяг вільного місця на твердому диску 20Гб;
- сервера, що може бути реалізований на базі ПК із мінімальною конфігурацією Pentium 3000 MHz /1024/200, із встановленої ОС SUSE Linux і СКБД MySQL;

– локальної мережі для надання з робочих місць користувачів доступу до даних, що знаходиться в БД сервера, із пропускнуою здатністю 10/100Мбіт/с.

### **3. Виконання програми.**

Виконання програми на кожному з робочих місць починається запуском веб-оглядача та вводом адреси системи у мережі. Безпосередньо після цього Користувач системи повинний пройти авторизацію – ввести в діалоговому вікні, що з'явилося, користувальницьке ім'я і пароль. Пароль є унікальною комбінацією буквено-цифрових символів, відомих тільки користувачу. Якщо ви забули пароль – звернетесь до системного адміністратора.

Після успішного входу в систему користувач бачить екран з двома списками завдань, перший завдання які він отримав, другий які він створив.

При переході за посиланням на перегляд отриманого завдання, користувач може відмовитися від нього, завершити або створити нове підзавдання. Якщо ж відбувається перегляд створеного користувачем завдання тоді він може його видалити (якщо стан завдання не відмінний від початкового та завдання не має "потомків"), створити підзавдання або якщо завдання було відхилене або завершене він може його схвалити чи навпаки (з указівками причин). Усі дії можуть бути виконані при натисканні на відповідні кнопки. Усі перелічені дії повертають на попередній екран.

З цього ж екрану користувач може перейти:

– на екран де може змінити свої налаштування (кількість секунд для оновлення екрану завдань, отримувати листа при отриманні завдання, звіту або повідомлення), після заповнення форми користувач повинен натиснути на кнопку "Save" щоб дані були збереженні у базі;

– на екран статистики де відображується статистика за обраний період.

Після вибору початкового періоду та кінцевого користувач бачить кількість виконаних та отриманих завдань, а також їх перелік, якщо у користувача є підлеглі то він може переглянути їхню статистику теж, вибором працівника у випадаючому списку:

– на екран звітів, де бачить свої звіти у вигляді посилань на сторінку з

текстом звіту, або якщо у користувача є підлеглі то відіслані ними звіти теж. Користувач може створити новий звіт натиском на кнопку "New Report"

– на екран повідомлень, де бачить повідомлення адресовані йому, або надіслані ним, він може їх видалити, створити нове натиском відповідної кнопки.

Користувач може вийти з системи натиском на відповідне посилання з екрану списку звітів, він може завершити дію веб-оглядача, але у цілях безпеки рекомендується виходити з системи.

#### **4. Повідомлення користувачу.**

При введенні некоректних даних над полями уведення виводиться діалогове вікно з описом типу і місця виникнення помилки.

При неправильному введенні чи пароля імені користувача виводиться повідомлення "Wrong login or password"

## Додаток В. - Тексти програмних модулів системи.

### Веб дескриптор (web.xml)

```
<?xml version="1.0" encoding="UTF-8"?>
<web-app version="2.4"
  xmlns="http://java.sun.com/xml/ns/j2ee"
  xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
  xsi:schemaLocation="http://java.sun.com/xml/ns/j2ee
http://java.sun.com/xml/ns/j2ee/web-app_2_4.xsd" >
  <context-param>
    <param-name>javax.servlet.jsp.jstl.fmt.localizationContext</param-name>
    <param-value>MessageResources</param-value>
  </context-param>
  <servlet>
    <servlet-name>action</servlet-name>
    <servlet-class>org.apache.struts.action.ActionServlet</servlet-class>
    <init-param>
      <param-name>config</param-name>
      <param-value>/WEB-INF/struts-config.xml</param-value>
    </init-param>
  </servlet>
  <servlet-mapping>
    <servlet-name>action</servlet-name>
    <url-pattern>*.html</url-pattern>
  </servlet-mapping>

  <filter>
    <filter-name>UrlRewriteFilter</filter-name>
    <filter-class>org.tuckey.web.filters.urlrewrite.UrlRewriteFilter</filter-
class>
    <init-param>
      <param-name>logLevel</param-name>
      <param-value>WARNING</param-value>
    </init-param>
  </filter>
  <filter-mapping>
    <filter-name>UrlRewriteFilter</filter-name>
    <url-pattern>/*</url-pattern>
  </filter-mapping>

  <filter>
```

```
<filter-name>CharEncodingFilter</filter-name>
  <filter-class>org.dionis.todo.web.utils.CharEncodingFilter</filter-class>
</filter>
<filter-mapping>
  <filter-name>CharEncodingFilter</filter-name>
  <url-pattern>/*</url-pattern>
</filter-mapping>

<jsp-config>
  <jsp-property-group>
    <url-pattern>*.jspx</url-pattern>
    <page-encoding>utf-8</page-encoding>
    <is-xml>>true</is-xml>
  </jsp-property-group>
</jsp-config>

<session-config>
  <session-timeout>30</session-timeout>
</session-config>

<welcome-file-list>
  <welcome-file>index.jsp</welcome-file>
  <welcome-file>admin/index.jsp</welcome-file>
</welcome-file-list>

<error-page>
  <error-code>404</error-code>
  <location>/nopage.html</location>
</error-page>

<security-role>
  <description>Authorized to access to admin zone</description>
  <role-name>todo</role-name>
</security-role>
<security-constraint>
  <display-name>Security</display-name>
  <web-resource-collection>
    <web-resource-name>Admin Zone</web-resource-name>
    <url-pattern>/admin/*</url-pattern>
  </web-resource-collection>
  <auth-constraint>
    <role-name>todo</role-name>
  </auth-constraint>
</security-constraint>
```

```
<login-config>
    <auth-method>BASIC</auth-method>
</login-config>
</web-app>
```

## Клас для роботи з електронною поштою (Mailer.java)

```
package org.jssoft.todo.ejb.utils;

import javax.mail.Transport;
import javax.mail.Session;
import javax.mail.Message;
import javax.mail.Address;
import javax.mail.internet.MimeMessage;
import javax.mail.internet.InternetAddress;
import java.util.Properties;
import java.util.Date;

public class Mailer {
    private Transport transport = null;
    private static Session mailConnection;

    public Mailer() {
        try {
            Properties props = new Properties();
            props.put("mail.smtp.auth", true);
            mailConnection = Session.getInstance(props, null);
            transport.connect("localhost", "", "");
        } catch (Exception e) {
            transport = null;
            e.printStackTrace();
        }
    }

    public boolean mail(String emailTo, String nameTo, String emailFrom, String
nameFrom, String content, String subject, boolean plain) {
        if (null == transport && !initTransport()) return false;
        try {
            Message msg = new MimeMessage(mailConnection);
            Address addressTo = new InternetAddress(emailTo, nameTo);
            Address addressFrom = new InternetAddress(emailFrom, nameFrom);
            msg.setContent(content, plain ? "text/plain" : "text/html");
            msg.setFrom(addressFrom);
            msg.setRecipient(Message.RecipientType.TO, addressTo);
        }
    }
}
```

```

        msg.setSubject(subject);
        msg.setSentDate(new Date());
        msg.saveChanges();
        transport.sendMessage(msg, msg.getAllRecipients());
        return true;
    } catch(Exception e) {
        e.printStackTrace();
        return false;
    }
}

private boolean initTransport() {
    try {
        Properties props = new Properties();
        props.put("mail.smtp.auth", "true");
        mailConnection = Session.getInstance(props, null);
        transport = mailConnection.getTransport("smtp");
        transport.connect("localhost", "", "");
        return true;
    } catch(Exception e) {
        e.printStackTrace();
        transport = null;
        return false;
    }
}

@Override
protected void finalize() throws Throwable {
    transport.close();
    super.finalize();
}
}

```

## **SessionBean клас, що реалізує методи для роботи з завданнями (TaskSessionBean.java)**

```

package org.jgsoft.todo.ejb.session;

import org.jgsoft.todo.ejb.entity.Task;
import org.jgsoft.todo.ejb.entity.User;
import org.jgsoft.todo.ejb.entity.Action;
import org.jgsoft.todo.ejb.exception.NotFoundException;
import org.jgsoft.todo.ejb.exception.TODOException;
import javax.ejb.Stateless;

```

```

import javax.persistence.PersistenceContext;
import javax.persistence.EntityManager;
import java.util.List;

@Stateless
public class TaskSessionBean implements TaskSession {
    @PersistenceContext(unitName = "todo")
    EntityManager em;

    public Task createTask(Task t) {
        em.persist(t);
        return t;
    }

    public Task editTask(Task t) throws NotFoundException, TodoException {
        Task tt = em.find(Task.class, t.getId());
        if (tt == null) throw new NotFoundException("Task with this id not
found");
        if (searchTasks(tt).isEmpty()) throw new TodoException("Task have
childrens");
        tt = t;
        return tt;
    }

    public void removeTask(long id) throws NotFoundException, TodoException {
        Task tt = em.find(Task.class, id);
        if (tt == null) throw new NotFoundException("Task with this id not
found");
        if (searchTasks(tt).isEmpty()) throw new TodoException("Task have
childrens");
        em.remove(tt);
    }

    public Task getTaskById(long id) {
        return em.find(Task.class, id);
    }

    protected List<Task> searchRecTasks(Task parent) {
        List<Task> l = em.createQuery("select t from Task t where t.parent =
:parent").setParameter("parent", parent).getResultList();
        if (!l.isEmpty)
        for (Task task : l) {
            l.addAll(searchRecTasks(task));
        }
    }
}

```

```

        return l;
    }

    public List<Task> searchTasks(Task parent) {
        return searchRecTasks(parent);
    }

    public List<Task> getToUserTasks(User u) {
        return em.createQuery("select t from Task t where t.reciever = :u and
t.state <> :state").setParameter("state",
Task.State.DONE_APPROVED).setParameter("u", u).getResultList();
    }

    public List<Task> getFromUserTasks(User u) {
        if (u.getSettings().isShowDoneNoteApproved()) {
            return em.createQuery("select t from Task t where t.creator = :u and
t.state <> :state").setParameter("state",
Task.State.DONE_APPROVED).setParameter("u", u).getResultList();
        } else {
            return em.createQuery("select t from Task t where t.creator = :u and
t.state <> :state1 and t.state <> :state2").setParameter("state1",
Task.State.DONE_APPROVED).setParameter("state2",
Task.State.DONE).setParameter("u", u).getResultList();
        }
    }

    public List<User> getPermittedUsers(User u) {
        return em.createQuery("select u from User u where u.leader = :l or
u.department = :d").setParameter("l", u.getLeader()).setParameter("d",
u.getDepartment()).getResultList();
    }

    public List<User> getSubUsers(User u) {
        return em.createQuery("select u from User u where u.leader =
:l").setParameter("l", u).getResultList();
    }

    public Action addAction(Action a) {
        em.persist(a);
        return a;
    }
}

```



```

        <td> </td>
        <td><html:submit/></td>
    </tr>
</table>
</html:form>
</body>
</html:html>
</jsp:root>

```

## Action клас, що реалізує авторизацію користувача (LoginAction.java)

```

package org.jgsoft.todo.web.action;

import org.apache.struts.action.*;
import org.apache.struts.validator.DynaValidatorForm;
import org.jgsoft.todo.ejb.session.UserSession;
import org.jgsoft.todo.ejb.session.UserSessionBean;
import org.jgsoft.todo.ejb.entity.User;
import org.jgsoft.todo.web.utils.LookupUtil;
import javax.servlet.http.HttpServletRequest;
import javax.servlet.http.HttpServletResponse;

public class LoginAction extends Action {
    public ActionForward execute(ActionMapping mapping, ActionForm aForm,
        HttpServletRequest request, HttpServletResponse response) throws Exception {
        ActionMessages validatorErrors = new ActionMessages();
        request.setAttribute("validatorErrors", validatorErrors);
        DynaValidatorForm form = (DynaValidatorForm) aForm;
        String errors = "";
        try {
            if ("login".equals((String) form.get("act"))) {
                validatorErrors.add(form.validate(mapping, request));
                if (validatorErrors.isEmpty()) {
                    UserSession uSession =
LookupUtil.getSessionBean(UserSessionBean.class);
                    User user =
uSession.getUserByLoginAndPassword((String) form.get("login"),
                        (String) form.get("password"));
                    request.getSession().setAttribute("loggedUser", user);
                    return mapping.findForward("tasksListAction");
                }
            }
        } catch (Exception te) {
            errors = te.getMessage();
        }
    }
}

```

```
    }
    request.setAttribute("errors", errors);
    return mapping.findForward("login");
}
}
```

## **Фільтр переводу символів у UTF-8 (CharEncodingFilter.java)**

```
package org.jgsoft.todo.web.utils;

import javax.servlet.*;
import java.io.IOException;

public class CharEncodingFilter implements Filter {
    private FilterConfig filterConfig = null;
    private String encoding = null;

    public void init(FilterConfig filterConfig) throws ServletException {
        this.filterConfig = filterConfig;
        this.encoding = this.filterConfig.getInitParameter("encoding");
    }

    public void destroy() {
        this.filterConfig = null;
    }

    public void doFilter(ServletRequest request, ServletResponse response,
                        FilterChain chain) throws IOException, ServletException
    {
        if (encoding != null) {
            request.setCharacterEncoding(encoding);
            response.setCharacterEncoding(encoding);
        } else {
            request.setCharacterEncoding("UTF-8");
            response.setCharacterEncoding("UTF-8");
        }
        chain.doFilter(request, response);
    }
}
```