

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.

" ____ " _____ 2021 р.

КВАЛІФІКАЦІЙНА РОБОТА

на здобуття ступеня вищої освіти "магістр"

на тему: Розробка інформаційної системи документообігу
закладу позашкільної освіти

Виконав: студент 6141м групи

_____ ***Каратай М.В.***
(підпис)

Керівник роботи:

професор кафедри ІУСТ, д.ф-м.н.,
(посада, науковий ступінь вчене звання)

_____ ***Ушкац М.В.***
(підпис)

м. Миколаїв – 2021 р.

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
Спеціальність 122 "Комп'ютерні науки"
Освітня програма "Комп'ютерні науки"

"ЗАТВЕРДЖУЮ"

Гарант освітньої програми

к.т.н, доц. Ажищев В.Ф.

" ____ " _____ 2021 р.

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

Студенту Каратаю Максиму Віталійовичу

(прізвище, ім'я, по батькові)

1. Тема роботи: Розробка інформаційної системи документообігу закладу позашкільної освіти

Керівник роботи д.ф-м.н., професор Ушкац М.В.

Затверджені наказом ректора № 1232-уч. від "12" жовтня 2021 року.

2. Термін подання роботи: 21 грудня 2021 р.

3. Вихідні дані до проекту (роботи) ДСТУ щодо обробки інформації, літературні джерела, технічна документація ІС палацу творчості учнів, матеріали наукового стажування

4. Перелік питань, що належать до розробки (найменування розділів) _____

1 Аналіз предметної області і постановка задачі. 1.1 Загальний опис проблеми. 1.2 Аналіз підприємства для якого виконується автоматизація. 1.3 Аналіз існуючих рішень. 2 Розробка проектних рішень. 2.1 Інформаційне забезпечення. 2.2 Програмне забезпечення. 3 Реалізація проекту автоматизованої системи. 4 Охорона праці.

5. Перелік презентаційних матеріалів

Структура Палацу творчості учнів, Схема комп'ютерної мережі, Функціональна модель інформаційної системи, Концептуальна модель даних, Фізична структура БД, Схема взаємодії модулів ПЗ, Знімки екранних форм

6. Консультанти розділів роботи

Розділ	Прізвище, ініціали та посада консультанта	Підпис, дата	
		завдання видав	завдання прийняв
1	Д.ф-м.н., проф. Ушкац М.В.		
2	Д.ф-м.н., проф. Ушкац М.В.		
3	Д.ф-м.н., проф. Ушкац М.В.		
4			
5	Ст. викл. Гурець Н.В.		
6	Ст. викл. Гурець Н.В.		

7. Дата видачі завдання 04 вересня 2021 р.

КАЛЕНДАРНИЙ ПЛАН

№ з/п	Назва етапів дипломного проекту (роботи)	Строк виконання етапів роботи	Примітка
1	Вибір теми, визначення мети, завдань, об'єкта та предмета дослідження		
2	Підбір та опрацювання літератури		
3	Дослідження технологій інформаційного забезпечення навчально-наукових лабораторій матеріалознавчого профілю		
4	Розробка концепції інформаційного забезпечення навчально-наукової лабораторії неметалевих матеріалів		
5	Розробка та реалізація проекту		
6	Розробка додаткових розділів, оформлення роботи, підготовка презентаційних матеріалів		
7	Представлення до попереднього захисту	1 грудня 2021 р.	
8	Отримання відгуку керівника	13 грудня 2021 р.	
9	Отримання рецензії	15 грудня 2021 р.	
10	Захист кваліфікаційної роботи	21 грудня 2021 р.	

Студент

_____ (підпис)

Каратай М.В.

(ПІБ)

Керівник роботи

_____ (підпис)

Ушкац М.В.

(ПІБ)

АНОТАЦІЯ

Дана робота присвячена розробці проекту інформаційної системи позашкільного закладу освіти – Палацу творчості учнів міста Миколаєва. У даній роботі проведений аналіз сучасних інформаційних технологій, які використовують для розробки інформаційних систем; сформовані вимоги до інформаційної системи; розроблена її концепція; проведене техніко-економічне обґрунтування розробки системи; розроблене технічне завдання; розроблені загальносистемні рішення, рішення з інформаційного, організаційного, технічного та програмного забезпечення; розроблена відповідна документація. Також у роботі виконані розділи з економіки, охорони праці, охорони оточуючого середовища.

Робота виконана на ____ сторінках машинописного тексту, містить ____ рисунка, ____ таблиць, ____ додатків і список використаних джерел з ____ найменувань.

ABSTRACT

This diploma work is devoted to development of the project of information system of Palace of Pupil's Creativity. The work includes: analysis of modern information technologies, which are used for information system development; formulation of system requirements; system conception development; technical and economical proof of system development benefits; technical task formulation; development of system basic solutions, solutions in informational, organizational, technical and software aspects of system maintenance; development of appropriate documentation. Also, this work contains labor protection section.

The degree work is stated on ____ pages of typescript, contains ____ figures, ____ tables, ____ enclosures, and the list of ____ original sources.

ЗМІСТ

ВСТУП	
РОЗДІЛ 1. ФОРМУВАННЯ ВИМОГ ДО ІНФОРМАЦІЙНОЇ СИСТЕМИ ДОКУМЕНТООБІГУ ПАЛАЦУ ТВОРЧОСТІ УЧНІВ, РОЗРОБКА ЇЇ КОНЦЕПЦІЇ І ПОСТАНОВКА ЗАДАЧІ	
1.1 Опис інформаційної системи документообігу Палацу творчості учнів	
1.2 Формування вимог до інформаційної системи документообігу Палацу творчості учнів	
1.3 Розробка концепції інформаційної системи документообігу Палацу творчості учнів	
1.4 Постановка задачі	
РОЗДІЛ 2. РОЗРОБКА ПРОЕКТНИХ РІШЕНЬ ПО СИСТЕМІ ТА ЇЇ ЧАСТИНАХ	
2.1 Загальносистемні рішення	
2.2 Рішення з організаційного забезпечення	
2.3 Рішення з інформаційного забезпечення	
2.4 Рішення з технічного забезпечення	
2.5 Рішення з програмного забезпечення	
РОЗДІЛ 3. РОЗРОБКА РОБОЧОЇ ДОКУМЕНТАЦІЇ	
РОЗДІЛ 4. РОЗРАХУНОК ЕКОНОМІЧНОГО ЕФЕКТУ ВІД ВПРОВАДЖЕННЯ СИСТЕМИ ПАЛАЦУ ТВОРЧОСТІ УЧНІВ	
РОЗДІЛ 5 ОХОРОНА ПРАЦІ	
РОЗДІЛ 6. ОХОРОНА НАВКОЛИШНЬОГО СЕРЕДОВИЩА	
ВИСНОВКИ	
СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ	
ДОДАТОК А. ЗАГАЛЬНИЙ ОПИС СИСТЕМИ	
ДОДАТОК Б. КЕРІВНИЦТВО КОРИСТУВАЧА	

ДОДАТОК В. ІНСТРУКЦІЯ З ФОРМУВАННЯ І ВЕДЕННЯ БАЗИ ДАНИХ	
ДОДАТОК Г. ПРОГРАМА І МЕТОДИКА ВИПРОБУВАНЬ	
ДОДАТОК Д. ТЕКСТ ПРОГРАМИ	

ВСТУП

Інформаційні технології є найбільш важливою складовою процесу використання інформаційних ресурсів суспільства. До теперішнього часу вони пройшли кілька еволюційних етапів, зміна яких визначалася головним чином рівнем розвитку науково-технічного прогресу, появою більш досконалих засобів обробки інформації. Сучасний науково-технічний прогрес немислимий без широкого впровадження і використання обчислювальної техніки в освіті, виробництві, управлінні та наукових дослідженнях. 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.

Метою даної роботи є розробка проекту інформаційної системи документообігу Палацу творчості учнів міста Миколаєва. Ця система дозволить збільшити продуктивність і поліпшити умови праці вчителів шляхом автоматизації наступних функцій:

- Облік успішності учнів.
- Облік відвідування занять учнями.
- Видача завдань учням вчителями.
- Надання вчителю інформації про розклад його занять (організація робочого часу).
- Створення придатних до друку звітів про успішність, відвідування занять і видані завдання.
- Ведення електронної записної книжки (щоденника) вчителя.

Вигода від створення інформаційної системи Палацу творчості учнів також полягає в тому, що інформація буде зберігатися в єдиній базі даних, вона буде

систематизована і придатна для подальшої обробки. Цю інформацію зможуть надалі використовувати інші інформаційні системи.

Для досягнення поставленої мети необхідно вирішити наступні задачі:

- Провести аналіз сучасних інформаційних технологій, що використовуються для розробки інформаційних систем.
- Сформулювати вимоги до системи.
- Розробити концепцію системи.
- Провести техніко-економічне обґрунтування розробки системи.
- Розробити загальносистемні рішення, рішення з інформаційного, організаційного, технічного та програмного забезпечення.
- Розробити необхідне програмне забезпечення системи.
- Розробити відповідну документацію.

Об'єктом дослідження є засоби автоматизації діяльності закладів освіти. Предметом дослідження є інформаційна система позашкільного закладу освіти – Палацу творчості учнів.

Впровадження інформаційної системи документообігу позашкільного навчального закладу планується здійснити у Палаці творчості учнів міста Миколаєва.

РОЗДІЛ 1 ФОРМУВАННЯ ВИМОГ ДО ІНФОРМАЦІЙНОЇ СИСТЕМИ ДОКУМЕНТООБІГУ ПАЛАЦУ ТВОРЧОСТІ УЧНІВ, РОЗРОБКА ЇЇ КОНЦЕПЦІЇ І ПОСТАНОВКА ЗАДАЧІ

1.1 Опис інформаційної системи документообігу Палацу творчості учнів

Інформаційна система документообігу позашкільного закладу є інформаційною системою, яка призначена для автоматизації роботи педагогів (керівників гуртків) позашкільного закладу – Палацу творчості учнів міста Миколаєва. Функції, що автоматизуються системою, пов'язані з організацією робочого часу педагогів, обліком відвідування й успішності учнів.

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 і програмні засоби зв'язку і т.д. 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 (див. 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 виборки інформації.

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.

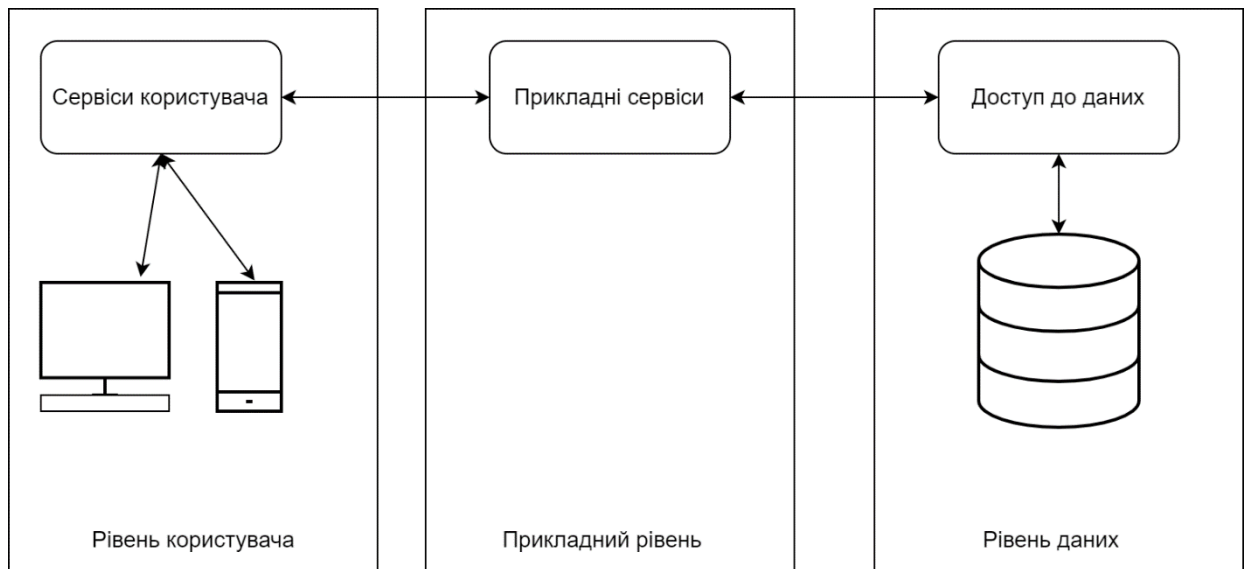


Рис. 1.1 – Логічна архітектура програмного забезпечення інформаційної системи

1.1.1 Технології рівня користувача

Web-оглядач і Web-інтерфейс

Web-оглядач чи Web-браузер – 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 розповсюджен і пропонує готові засоби відображення. Web-оглядачі, що вільно розповсюджуються, мають прості інтерфейси, що збільшує ефективність розгортання програм.

Найпоширеніші Web-оглядачі: MS Internet Explorer, Netscape Navigator, Mozilla, Opera.

HTML і DHTML

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 ґрунтуються на використанні графічного інтерфейсу користувача

Однією з найважливіших властивостей HTML є можливість включення в документ гіпертекстових посилань [1]. Ці посилання дозволяють користувачу завантажити новий документ на свій комп'ютер, просто клацнувши вказівником миші в тій 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 сучасних версіях Web-оглядачів Mozilla Firefox, Google Chrome, Opera, або MS Internet Explorer.

DHTML заснована на технологіях DOM, CSS, JavaScript і HTML.

DOM

DOM (Document Object Model, об'єктна модель документа) – це спеціальна, мовно-незалежна інтерфейсна модель розбору XML і HTML документів. DOM стандартизована W3C.

Ключовим поняттям DOM є об'єкт нового типу – вузол (node). Вузлом у HTML-документі відповідають HTML елементи, їхні атрибути і текстовий зміст. Ієрархія (дерево) вузлів задається вкладеністю тегов друг у друга і тексту усередину елементів. Навігацію по дереву документа можна починати з будь-

якого вузла-елемента, для якого відомий ідентифікатор, що присвоюється йому як значення атрибута ID. DOM, як редактор, дозволяє копіювати, вставляти, замінювати і вилучати як окремі вузли, так і цілі гілки дерева документа.

Вузли мають властивості-характеристики. Властивість вузла `nodeType` (тільки для читання) повертає визначені значення (1, 2 чи 3) для вузлів, що відповідають тегу, атрибуту чи тексту. Властивість `nodeName` (тільки для читання) повертає ім'я HTML тега, якому відповідає даний вузол. Для вузлів-атрибутів `nodeName` повертає назву атрибута, а для текстових вузлів повертає `#text`. Текстові вузли мають ще одну дуже важливу властивість: `nodeValue`. Ця властивість для читання і запису зберігає зміст текстового вузла. Для елементів воно повертає `null`, а для атрибутів – значення атрибута.

DOM можна використовувати для створення і роботи з DHTML-документами (Web-оглядач повинен підтримувати DOM).

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) [2]. 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 [1].

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 [3]. 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, Mozilla Firefox, Google Chrome 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.

Найрозповсюджені 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 [3]. Головна задача, яку він вирішує, – це забезпечення викликів віддалених процедур між клієнтом і сервером з використанням сокетів TCP/IP. Складовими частинами CGI-системи є:

- клієнтська програма – програма на деякій мові програмування, Java-апплет, HTML-документ, що містить форму – яка формує стандартний запит для Web-сервера відповідно до правил протоколу HTTP. Запит містить усю необхідну інформацію – звідки посланий, хто його повинен

обробити, інформацію, еквівалентну імені процедури, список аргументів і їхніх значень; після посилки запиту клієнтська програма очікує результату його виконання й обробляє його тим чи іншим способом.

- Web-сервер. Під Web-сервером тут розуміється програма, яка приймає запити клієнтів, що приходять на стандартний порт TCP/IP (порт 80), і виконує наступні дії: створення пакета вхідної інформації на основі запиту клієнта, запис цього пакета в стандартний потік виводу для наступної обробки деякою програмою, що відома Web-серверу, ініціалізація цієї програми, отримання (зі стандартного потоку вводу) результатів її роботи в тому чи іншому виді, приєднання стандартного HTTP-заголовка і відправлення всього цього клієнту.
- програма-сервер, тим чи іншим способом зареєстрована на Web-сервері, яка виконує всі необхідні дії у відповідь на запит клієнта. Вхідний пакет інформації вона отримує від Web-сервера, використовуючи стандартний потік вводу, результат поміщає в стандартний потік виводу. Цією програмою може бути командний файл, скрипт на JavaScript чи мові Perl, Java-програма чи звичайна програма на C, C++ чи іншій мові.

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 [4].

1.1.3 Технології рівня даних

OLE DB

OLE DB – основа розробленої компанією Microsoft універсальної архітектури даних (Universal Data Architecture, UDA). UDA призначена для організації високопродуктивного доступу до даних будь-якого типу –

структурованих і неструктурованих, зв'язаних і незв'язаних. UDA – набір COM-інтерфейсів, що реалізують концепцію доступу до 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 [5].

Стандарти SQL затверджені 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) [5]. Сервер передає обрану з бази інформацію по каналу клієнту. Саме сервер працює з даними, піклується про їхнє розміщення на диску.

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

1.1.4 Динаміка роботи інформаційної системи Палацу творчості учнів

Маючи загальне уявлення про технології, які можуть використовуватися для розробки системи позашкільного закладу, можна описати динаміку роботи системи:

- Користувач (педагог, завуч, директор, або інший користувач) запускає Web-оглядач на віддаленому комп'ютері-клієнті і вводить адресу Web-сервера інформаційної системи.
- Web-оглядач надсилає запит Web-серверу.
- Web-сервер робить аутентифікацію користувача по логіну і паролю (у випадку неправильної аутентифікації Web-сервер відмовляє клієнту в доступі до системи).
- Web-сервер отримує запит клієнта і запускає відповідний сервлет, передаючи йому вхідні параметри.
- сервлет обробляє вхідні дані і звертається до EJB-сервера (точніше до Java Enterprise сервера, що запускає EJB-контейнер, що виконує роль EJB-сервера) з «проханням» створити необхідний EJB-об'єкт.
- EJB-сервер (він може працювати як на тому ж комп'ютері, що і Web-сервер, так і на іншій машині) створює потрібний EJB-об'єкт, а сервлет отримує вказівник на його інтерфейс.
- сервлет викликає необхідні методи EJB-об'єкта.
- EJB-об'єкт, за допомогою EJB-інтерфейсів до JDBC, звертається до сервера баз даних для отримання, зміни, додавання чи видалення даних.
- Сервер баз даних робить аутентифікацію клієнта (EJB-об'єкта) по логіну і паролю (у випадку неправильної аутентифікації сервер баз даних відмовляє клієнту в доступі до інформації).
- Сервер баз даних робить необхідні операції з даними і посилає результати роботи EJB-об'єкту.
- EJB-об'єкт обробляє отримані дані і надає їх сервлету.
- сервлет обробляє отримані дані, генерує HTML/DHTML-документ (Web-інтерфейс) на основі JSP і передає його Web-серверу.

- Web-сервер додає HTTP-заголовок до отриманого документа і відправляє його Web-оглядачу.
- Web-оглядач приймає HTML/DHTML-документ і відображає його на екрані користувача.
- Користувач отримує Web-інтерфейс і необхідну інформацію. Він може надіслати новий запит Web-серверу за допомогою клацання вказівником миші по гіпертекстових посиланнях і кнопкам HTML-форм. У такому випадку, послідовність операцій повториться знову (за винятком аутентифікації Web-сервера).

Таким чином, можна визначити схему взаємодії технологій (рисунок 1.2).

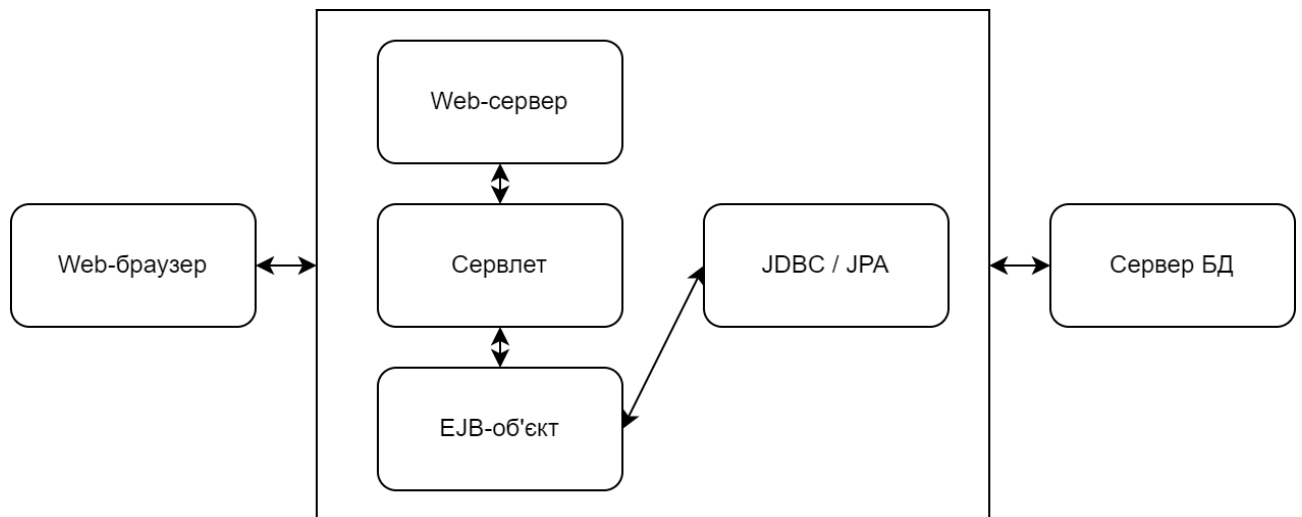


Рис. 1.2 – Схема взаємодії технологій інформаційної системи позашкільного закладу

1.2 Формування вимог до інформаційної системи документообігу Палацу творчості учнів

1.2.1 Характеристика інформаційної системи Палацу творчості учнів

Об'єктом автоматизації є інформаційна система позашкільного закладу, пов'язана з роботою педагог закладу. Існуюча інформаційна система Палацу творчості учнів (її комп'ютеризована версія перебуває в стадії розробки) має потребу в створення єдиної бази даних для централізованого збереження

інформації й автоматизованих робочих місць (АРМ) для співробітників закладу (наприклад, АРМ співробітника медпункту, АРМ директора, АРМ завуча). Об'єкт автоматизації сам по собі є інформаційною системою, для автоматизації роботи якої потрібно створення власної бази даних і програмного забезпечення для педагогів і адміністратора. Об'єкт автоматизації – інформаційна система позашкільного закладу – вирішує задачі пов'язані з обліком успішності і відвідування учнів, видачею завдань учням, організацією робочого часу педагогів і інші.

1.2.2 Опис діючої інформаційної системи Палацу творчості учнів

Діюча інформаційна система не є автоматизованою. Її ключовою фігурою (об'єктом) є педагог (керівник гуртка). В обов'язки педагога входить навчання, перевірка знань, облік відвідування занять, облік успішності учнів і інші види діяльності. Зараз нас цікавить робота педагога по обліку відвідування й успішності учнів, а також організація розпорядку його праці (розклад).

Як правило, педагог повинен сам вести журналу відвідування та успішності. Найчастіше журнал зберігається у кабінеті завучів (завідувачів напрямків) і кожен педагог повинен брати його на свої заняття. Оскільки журнал є документом суворої звітності, виправлення в ньому мало бажані, педагог найчастіше відмічає відвідування його занять учнями, здачу робіт (домашніх, лабораторних, контрольних і інших) учнями у своєму власному журналі, а лише потім переносить до класних журналів. При цьому у педагога накопичується досить велика кількість документів, які він повинен носити із собою на кожне заняття. Доступ до цих документів має тільки педагог, тобто інформацію про успішність і відвідування учнів не може використовувати інший підрозділ закладу освіти.

Учні та батьки також не мають доступу до інформації про свої успіхи в опануванні матеріалів, що вивчаються.

Розклад керівника гуртка також ведеться ним на папері та із використанням електронних таблиць, що не є дуже зручним.

1.2.3 Опис недоліків діючої інформаційної системи Палацу творчості учнів

Діюча інформаційна система має багато недоліків, які в основному пов'язані з паперовою технологією збереження даних і збереженням даних у неструктурованому та децентралізованому сховищі. Основні недоліки існуючої інформаційної підсистеми такі:

- Незручність використання інформації, яка зберігається на папері. Відсутність можливості пошуку й обробки інформації. Педагог повинен завжди мати із собою відомості.
- Складність відновлення інформації у випадку її втрати. Якщо педагог загубить відомість, то йому потрібно буде «підняти» усі здані учнями роботи; інформація про відвідування може бути отримана з цих відомостей.
- Складність обробки й аналізу інформації на глобальному рівні. Наприклад, важко проаналізувати поточну успішність учнів якогось класу по всіх досліджуваних предметах, оскільки потрібно мати відомості усіх педагогів, що працюють з учнями цього класу.
- Відсутність єдиної бази даних, єдиного формату збереження інформації. Кожен педагог веде свої відомості так, як вважає за необхідне.
- Недостатній захист інформації. Можливість несанкціонованого доступу, зміни і знищення інформації у відомостях педагогів учнями й іншими особами, які не мають на це прав.

1.2.4 Обґрунтування необхідності вдосконалення діючої інформаційної системи документообігу Палацу творчості учнів

Діюча інформаційна система використовує застарілі методи обробки і збереження інформації. Паперова технологія і відсутність єдиного сховища (бази даних) інформації породжують багато недоліків системи, що описані вище.

Використання сучасних інформаційних технологій для автоматизації роботи інформаційної системи може вирішити всі ці недоліки, підвищити ефективність і якість роботи системи, а також збільшити її функціональність.

Удосконалення діючої інформаційної системи, зокрема створення єдиної бази даних, поліпшить умови праці педагогів (наприклад, не потрібно буде зберігати і носити із собою паперові відомості) і завучів (наприклад, одержання коректної і своєчасної інформації про відвідування занять учнями). Крім того, єдина база даних послужить основою для розробки програмного забезпечення з новими можливостями для учнів, педагогів, завучів і деяких інших підрозділів позашкільного закладу по обробці й аналізу інформації.

1.2.5 Цілі, критерії й обмеження створення інформаційної системи Палацу творчості учнів

Метою розробки (вдосконалення) інформаційної системи є збільшення продуктивності і поліпшення умов праці педагогів 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 закладу освіти 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. Утрата даних може привести до серйозних збоїв у роботі системи. Необхідний захист системи від несанкціонованого втручання.

Обмеженням інформаційної системи, яка розробляється, може вважатися те, що вона є тільки лише підсистемою більш глобальної інформаційної системи позашкільних закладів міста. Це означає, що використання ресурсів розроблювальної системи (наприклад, бази даних) можливо не тільки педагогами, але і деякими іншими підрозділами управління освіти, що не може бути враховане при створенні системи Палацу творчості учнів.

1.2.6 Функції та задачі інформаційної системи Палацу творчості учнів

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

- Облік успішності учнів.
- Облік відвідування занять учнями.
- Видача завдань учням.
- Формування звітів (відомостей) для друку.
- Допомога в організації робочого часу педагога.

Для створення інформаційної системи необхідно розробити єдину базу даних для збереження інформації про педагогів і учнів, про відвідування й успішність учнів, про розклад педагогів і іншу. Для роботи з системою педагоги повинні мати відповідне програмне забезпечення. Також повинно бути передбачене програмне забезпечення для адміністрування системи.

1.2.7 Висновки і пропозиції

Таким чином, необхідно розробити автоматизовану інформаційну систему позашкільного закладу освіти, що дозволить підвищити ефективність та якість роботи педагогів, та адміністрації закладу освіти, поліпшити умови їхньої праці. Система є частиною глобальної інформаційної системи закладів освіти міста та вирішує задачі, що пов'язані з трудовою діяльністю педагогів та адміністрації позашкільного закладу, а також обліком успішності учнів. Інформаційна система повинна включати базу даних, програмне забезпечення педагога та керівника (завуча, директора) та програмне забезпечення адміністратора системи, а також відповідне організаційне, технічне та інші види забезпечення.

У перспективі, база даних інформаційної системи Палацу творчості учнів може використовуватися різними підрозділами. Завуч може одержувати коректну і своєчасну інформацію про відвідування занять учнями. Використовуючи технології OLAP можна одержувати статистику відвідування занять учнями в різні пори року.

1.3 Розробка концепції інформаційної системи документообігу Палацу творчості учнів

1.3.1 Опис результатів вивчення системи Палацу творчості учнів

Інформаційна система Палацу творчості учнів розробляється як компонент глобальної інформаційної системи загальноосвітніх шкіл та позашкільних закладів міста і вирішує задачі, що пов'язані з автоматизацією праці педагогів та керівництва позашкільного закладу. Для створення системи потрібна розробка бази даних, програмного забезпечення для педагогів, керівників і адміністраторів, а також прийняття рішень по інших видах забезпечення. Основні задачі, що вирішує інформаційна система Палацу творчості учнів пов'язані з обліком успішності та відвідування учнів, видачею завдань учням, організацією робочого часу педагогів.

Розробка автоматизованої інформаційної системи Палацу творчості учнів дозволить підвищити ефективність і якість праці педагогів, поліпшити умови їхньої праці.

1.3.2 Аналіз вимог до інформаційної системи Палацу творчості учнів

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

- Простий і зручний інтерфейс користувача.
- Доступ до системи через Internet з будь-якого комп'ютера глобальної мережі.
- Багатокористувачева 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 (CGI, ISAPI, DCOM/COM+, CORBA, RMI і інші).
- Ефективні методи доступу до даних.
- Забезпечити простоту модернізації, можливість повторного використання програмного забезпечення (коду).

1.3.3 Вибір логічної архітектури програмного забезпечення інформаційної системи Палацу творчості учнів

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» [5]. 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, масштабованість 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. Масштабованість 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.

Масштабованість 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.

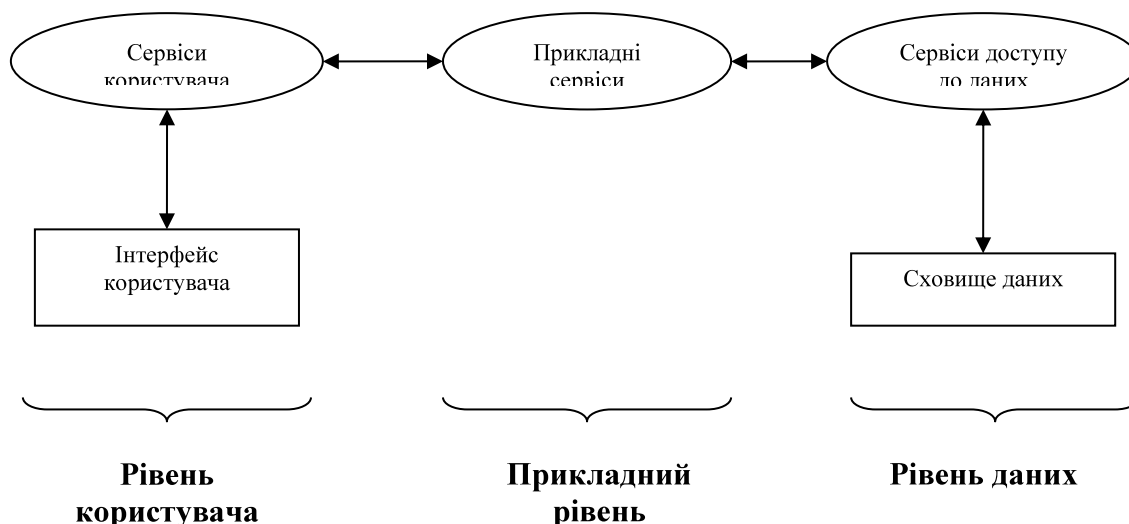


Рис.1.3 - Логічна архітектура програмного забезпечення інформаційної системи

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 виборки інформації.

1.3.4 Вибір фізичної архітектури програмного забезпечення інформаційної системи Палацу творчості учнів

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).



Рис.1.4 – Триланкова фізична архітектура програмного забезпечення інформаційної системи

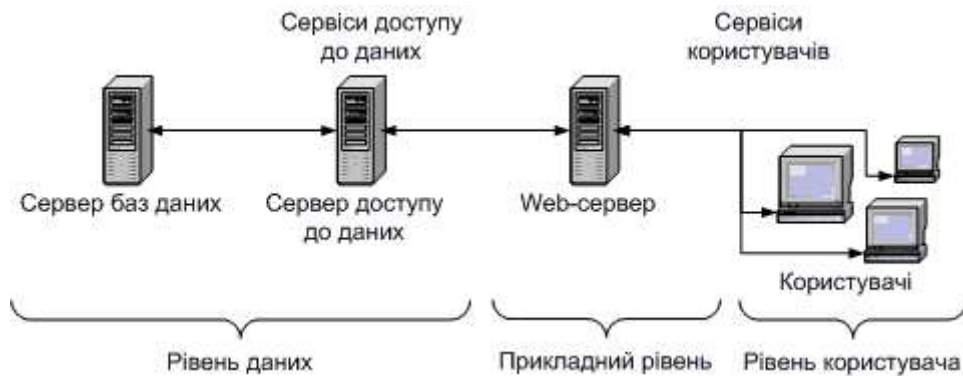


Рис. 1.5 – Чотириланкова фізична архітектура програмного забезпечення інформаційної системи

Порівняємо отримані фізичні архітектури для інформаційної системи.

Переваги триланкової архітектури над чотириланковою:

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

Переваги чотири ланкової 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 Палацу творчості учнів

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

Існують наступні технології для створення розподілених систем:

- CGI, ISAPI/NSAPI

- Servlets/JSP 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. Усі зазначені технології (CGI, ISAPI/NSAPI, Servlets) вирішують однакові задачі і мають приблизно однакові характеристики по продуктивності. Серед них CGI є універсальною і найпоширенішою (у той час як ISAPI обмежена використанням Windows-платформи, Netscape відмовилася від підтримки NSAPI), Servlets/JSP обмежена використанням мови Java, але через поширеність Java-технології, та її кросплатформовість її вибір є найкращим.

Для взаємодії серверів потрібно застосовувати більш розвинуті технології, наприклад, новішу технологію EJB.

RMI орієнтована на Java-технології і вимагає використання мови програмування Java. Але технологія EJB, яка побудована на основі різних технологій, у тому числі RMI, є більш універсальною.

Це безкоштовна технологія і на ній засновано багато інших технологій (наприклад, JDBC – ефективніший/продуктивний 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.

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

Доступ до даних можна здійснювати за допомогою технологій ODBC, JDBC, OLE DB і/або ADO. ADO заснована на OLE DB, тому безпосередня робота з OLE DB дасть кращу продуктивність програмного забезпечення. OLE DB також ефективніше за ODBC і є універсальною (дозволяє працювати з даними будь-якого типу). Проте найбільшій ефективності при роботі з БД у програмах,

розроблених мовою Java, можна досягнути використовуючи протокол та засоби JDBC та у поєднанні з EJB, засоби ORM (об'єктно-реляційного відображення) – JPA.

Обрані технології для розробки інформаційної системи Палацу творчості учнів представлені на рисунку 1.6.



Рис. 1.6 – Технології для розробки інформаційної системи Палацу творчості учнів

1.3.6 Розробка варіантів концепції інформаційної системи Палацу творчості учнів

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

Оскільки для розробки системи використовується технологія Java EE с EJB3, то на всіх трьох серверах може використовуватися будь-яка платформа, яка підтримує роботу віртуальних машин Java. Відзначимо, що на сервері баз даних можливо використання платформи, що не підтримує сучасних версій Java. Але при цьому адміністрування системи сильно ускладниться, а також може погіршитися продуктивність (якщо немає відповідного JDBC драйвера).

Необхідну функціональність, надійність, безпеку і продуктивність інформаційної системи можуть забезпечити наступні операційні системи (у порядку збільшення продуктивності і функціональності):

- Ubuntu 20.04 LTS.
- Oracle Solaris.
- Windows 10 Professional.

Ці мережні операційні системи включають підтримку віртуальної машин Java - HotSpot.

Випуск Oracle Solaris у вигляді вільної версії - OpenSolaris уже припинений, підтримка обмежена, тому використовувати цю операційну систему в нових розробках нераціонально.

Ubuntu 20.04 LTS хоча і слабше попередньої альтернативи, проте має тривалу підтримку від компанії Canonical і задовольняє усім вимогам інформаційної системи.

Windows 10 Professional має всі можливості попередніх версій Windows, а також додаткові можливості (підтримка кластерів, підтримка більшої кількості оперативної пам'яті і мікропроцесорів). Windows 10 Professional має кращу продуктивність і надійність, але можливості цієї операційної системи набагато перевищують максимальні вимоги до інформаційної системи, що розробляється,

тому немає необхідності купувати більш дорогу інформаційну систему, якщо її можливості не будуть використані повною мірою.

Таким чином, Ubuntu 20.04 – це найбільш оптимальний вибір для інформаційної системи, що розробляється. Ubuntu 20.04 може підтримувати оперативну пам'ять до 64Гб, до чотирьох мікропроцесорів (мінімальний CPU – 1Ггц) і жорсткий диск від 200Гб і більше. Усе це додасть інформаційній системі широкі можливості по нарощуванню продуктивності і забезпечить високу масштабуємість.

На Web-сервері повинен бути встановлений HTTP-сервер з підтримкою CGI, HTTP 1.1 і можливістю аутентифікації користувача. Найрозповсюдженими HTTP-сервера, що задовольняють зазначеним вимогам і мають високу продуктивність, є:

- Apache Server 2.1 і вище.
- MS IIS (Internet Information Services).
- Payara 5 Community edition і вище

На Web-сервері також повинно бути встановлене програмне забезпечення (сервлети та JSP), що реалізує бізнес-правила інформаційної системи; його необхідно розробити.

На сервері доступу до даних необхідно установити програмне забезпечення у виді EJB-компонентів (об'єктів), яке необхідно розробити.

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-92. Крім того, СУБД повинна працювати під управлінням Windows. Серверними СУБД, що задовольняють зазначеним вимогам, на сьогодні є:

- MS SQL Server 2008.

- MySQL 5.5 (або вище).
- PostgreSQL 9.
- Sybase (SQL Anyware Studio 8).
- Oracle 10i (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 10, Ubuntu Linux.

Програмне забезпечення, з яким буде працювати користувач, являє собою Web-оглядач з підтримкою HTML 5.0, JavaScript, CSS 3 (наприклад, MS Edge, Google Chrome та ін.).

Засоби розробки

Для розробки програмного забезпечення сервера доступу до даних (Сервлетів, JSP і EJB-об'єктів) необхідно використовувати мову програмування Java і засоби розробки, що підтримують Java-технологію. Такими засобами мовами є Eclipse, NetBeans, IntelliJ IDEA. Насправді, для компіляції модулів, що підтримуються віртуальною машиною Java, можна використовувати різноманітні мови програмування та засоби розробки. Так, існує компілятор мови Pascal, що трансліює програму у код віртуальної машини Java. Більш сучасні мови, наприклад Scala, Ruby, Python, мають компілятори, що трансліюють код з цих мов у код, придатний для виконання у віртуальній машині Java. Проте ці мови не мають загальноприйнятих стандартів розробки. Це не говорить про те, що ці мови використовувати не можна – це говорить про слабкість підтримки даних технологій на рівні відповідності затвердженим

промисловим стандартам. Крім того, Java є найрозповсюдженішою мовою програмування, що спростить розробку і супровід програмного забезпечення; програмне забезпечення, розроблене на Java має достатньо високу продуктивність [2]. Отже, у якості мови програмування для розробки сервлетів і EJB-об'єктів необхідно використовувати Java, а засіб розробки – IntelliJ IDEA. Також можливе використання інших засобів розробки, проте IntelliJ IDEA має найбільш потужні можливості[6].

Для розробки програмного забезпечення Web-сервера (сервлетів) можна використовувати різні мови, що мають можливості компіляції у код віртуальної машини Java: Pascal, Python, Ruby і інші. Кожна мова має свої переваги і недоліки. Python уступає мові Java у швидкості виконання програм, перевершує його в наявності могутніх вбудованих засобів обробки текстової інформації хоча програє у зручності розробки NetBeans підтримує цю мову лише в обмеженому обсязі. Ruby має аналогічні можливості що і Python, але Ruby не є універсальною мовою програмування і має незрівнянно більш слабку підтримку, чим Python чи Java. Pascal і інші мови програмування або менш придатні для рішення наших задач, або уступають по своїх характеристиках мовам Java, Python і Ruby. Не можна також забувати, що сервлети повинні взаємодіяти з EJB-об'єктами, а значить підтримувати EJB-технологію. Таким чином, для розробки сервлетів необхідно використовувати мову Java (Jakarta EE 9) завдяки наявності підтримки EJB 3.0, могутніх вбудованих засобів JPA, обробки інформації і зручності розробки і супроводу.

Мережне середовище

Позашкільний заклад освіти (Палац творчості учнів) 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, бажаною є швидкість 100 Мбіт.

Технічні засоби

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

- Мікропроцесор класу Intel i5-2500 і вище.
- Оперативна пам'ять RAM 8 Гб і більше.
- Жорсткий диск з вільним місцем 20 Гб і більше (сервер баз даних повинен мати жорсткий диск з більшим обсягом вільного місця – 80 Гб 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 Intel i3-550 і вище.
- Оперативна пам'ять RAM 4 Гб і більше.
- Жорсткий диск з вільним місцем 20 Гб і більше.
- Мережна карта Ethernet (не менше 10 Мбіт/с).

Варіанти концепції інформаційної системи

Таким чином, розглянувши компоненти системи і вибравши найбільш оптимальні, можна визначити кілька варіантів концепції інформаційної системи позашкільного закладу освіти (див. таблицю 1.1). У таблиці використані наступні скорочення:

- ПЗ – програмне забезпечення.
- ОС – операційна система.
- СУБД – система управління базами даних.

Таблиця 1.1 – Варіанти концепції інформаційної системи

№	ПЗ Web-сервера (ОС, HTTP-сервер)	ПЗ сервера баз даних (ОС, СУБД)	Засоби розробки (серверне ПЗ)
1	Ubuntu 20.04, Payara 5.2021	Ubuntu 20.04, MySQL 5.5 (MariaDB)	JDK 11 + Jakarta EE 9, IntelliJ IDEA 2021
2	MS Windows 10 Professional, Apache Server 2	Ubuntu 20.04, PostgreSQL 12	JDK 11 + Jakarta EE 9, IntelliJ IDEA 2021
3	MS Windows 2008 Server, MS IIS	MS Windows 2008 Server, Sybase (SQL AS 8)	JDK 11 + Jakarta EE 9, IntelliJ IDEA 2021
4	MS Windows 2008 Advanced Server, MS IIS	MS Windows 2008 Advanced Server, MS SQL Server 2000	JDK 11 + Jakarta EE 9, IntelliJ IDEA 2021

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 віртуальної машини Java та технології EJB. Сервер баз даних може працювати під управлінням будь-якої ОС. Якщо сервер баз даних не буде виділеним (наприклад, сполучений із сервером доступу до даних), то необхідне використання тієї ж системи, що і на веб-сервері.

- 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).
- Не існує реалізації СУБД PostgreSQL для Windows.
- СУБД MySQL не підтримує вкладених запитів (стандартний дистрибутив) і трохи втрачає в продуктивності при роботі під управлінням Windows-систем.
- Використовувати СУБД Oracle нераціонально – вона має високі вимоги до системних ресурсів, а її можливості не будуть використані повною мірою.
- Використання ОС Windows 2008 Server нераціонально, оскільки її могутні можливості не будуть використані повною мірою, а витрати на її придбання досить великі.
- Випуск ОС OpenSolaris уже припинений, підтримка обмежена, тому використовувати цю операційну систему в нових розробках нераціонально.
- Використання ОС Windows 7 Professional Server може підвищити тільки продуктивність системи в порівнянні з використанням Ubuntu, оскільки додаткові можливості цієї операційної системи не будуть затребувані.

1.3.7 Вимоги, що впливають на якість інформаційної системи Палацу творчості учнів

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 роботи закладу освіти.
- 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 Постановка задачі

1.4.1 Загальні відомості

Потрібно розробити автоматизовану інформаційну систему для автоматизації функцій, які виконуються педагогами та адміністрацією позашкільного закладу освіти. Замовником виступає Палац творчості учнів міста Миколаєва. Назва розробки: інформаційна система Палацу творчості учнів.

1.4.2 Призначення і мета створення інформаційної системи Палацу творчості учнів

Призначенням системи є автоматизація функцій, які виконуються педагогами та адміністративними робітниками (директор та його заступники) палацу, а також створення єдиної бази даних, у якій буде зберігатися вся інформація про поточну успішність і відвідування занять учнями закладу, а також інша інформація. У майбутньому, ця база даних може служити для адміністративних цілей міського управління освіти.

Метою створення системи є підвищення продуктивності і поліпшення умов праці педагогів та адміністрації позашкільного навчального закладу.

1.4.3 Вимоги до інформаційної системи Палацу творчості учнів в цілому

Режим роботи системи повинен відповідати режиму роботи співробітників (педагогів) Палацу творчості учнів; інструментарій віддаленого доступу повинен функціонувати цілодобово. У режимі роботи системи повинні бути передбачені інтервали для проведення профілактичних і технічних робіт.

Експлуатація інформаційної системи не вимагає створення додаткових робочих місць. Кваліфікація користувачів системи повинна відповідати вимогам до займаних ними посади. Рівень комп'ютерної підготовки користувачів повинен відповідати рівню «користувач ПЕОМ». Супроводження інформаційної системи повинно здійснюватися штатними фахівцями Палацу творчості учнів.

Компоненти системи повинні мати можливості для модернізації і розвитку в межах обраних технологій, що лежать у її основі, або навіть до їх часткової заміни.

Цільове призначення системи повинне зберігатися протягом часу остаточного застарівання технічного, програмного, інформаційного й організаційного забезпечення.

1.4.4 Вимоги до функцій, які виконує інформаційна система Палацу творчості учнів

Інформаційна система повинна мати наступні функціональні характеристики:

- 1) Аутентифікація користувача.
- 2) Ідентифікація користувача (надання користувачу тільки тих даних, які необхідні йому для виконання своєї роботи).
- 3) Конфіденційність передачі даних по комп'ютерній мережі.
- 4) Облік успішності учнів:
 - додавання проведеного заняття;
 - виставлення оцінок учням, які виконали завдання;

- зміна оцінок учням, які виконали завдання;
- вилучення оцінок.

5) Облік відвідування занять учнями:

- додавання проведеного заняття (уроку);
- додавання відмітки про відсутність учня на занятті;
- вилучення відмітки про відсутність учням на занятті.

6) Видача завдань учням:

- додавання нових завдань (лабораторних, домашніх і інших робіт);
- вилучення існуючих завдань;
- зміна існуючих завдань.

7) Надання користувачу (педагогу) інформації про його розклад занять.

8) Надання зручного Web-інтерфейсу:

- однакове представлення інформації незалежно від платформи і програмного забезпечення машини користувача;
- сортування даних;
- проста навігація;
- можливості настройки шрифтів інтерфейсу й інших параметрів інтерфейсу;
- можливість збереження параметрів інтерфейсу між сеансами роботи;
- можливість автоматичного заповнення полів форм.

9) Надання можливості створення паперових звітів про видані завдання, успішність і відвідування учнів; друк розкладу занять, які проводить педагог.

10) Ведення записної книжки (щоденника) педагога.

- додавання запису;
- зміна запису;
- вилучення запису.

11) Можливість віддаленої роботи з інформаційною системою як у локальній мережі, так і в Internet.

1.4.5 Вимоги до інформаційного забезпечення інформаційної системи Палацу творчості учнів

Система повинна бути відкритою (використовувати існуючі стандарти), тобто мати властивості мобільності й інтероперабельності.

Система повинна підтримувати розподілену багаторівневу архітектуру для забезпечення масштабуємості.

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

Web-сервер системи і сервлети повинні реалізовувати бізнес логіку системи і генерувати Web-інтерфейс, який забезпечує доступ до інформації через мережу Intranet/Internet. Серверні програми повинні генерувати документи і форми мовою HTML/DHTML. Генерація документів повинна здійснюватися на основі інформації, що зберігається в базі даних інформаційної системи, а також на основі файлів записної книжки. Вхідними даними будуть також поля форм, заповнені користувачем.

Сервіси доступу до бази даних повинні бути виділеними (працювати на окремій машині) з метою розвантаження Web-сервера і для поліпшення масштабуємості системи.

Сервер баз даних повинен бути виділений з метою розвантаження інших серверів і забезпечення більшої безпеки збереження інформації. У якості програмного забезпечення сервера бази даних необхідно використовувати серверну СУБД, що підтримує стандарт SQL-92 і універсальні засоби доступу до даних. Необхідно використовувати стандартні можливості СУБД із метою підтримки переносимості.

1.4.6 Вимоги до лінгвістичного забезпечення інформаційної системи Палацу творчості учнів

Web-інтерфейс повинен бути реалізований з використанням HTML/DHTML, CSS і JavaScript.

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

Бізнес логіка повинна розроблятися мовою Java, яка має могутні засоби для обробки будь-якої текстової інформації та мережових з'єднань.

Сервіси доступу до бази даних повинні розроблятися мовою високого рівня Java у виді EJB-об'єктів з використанням відповідних інтерфейсів JDBC. Запити до бази даних повинні здійснюватися з використанням мови SQL.

1.4.7 Вимоги до програмного забезпечення інформаційної системи Палацу творчості учнів

Інформаційна система включає кілька серверів (Web-сервер з бізнес логікою; сервер із сервісами доступу до даних; сервер баз даних) і клієнтські машини.

Вимоги до програмного забезпечення Web-сервера:

- Операційна система: Ubuntu 20.04 LTS, Windows 10 Professional.
Операційна система, що рекомендується – Ubuntu 20.04.
- Підтримка EJB 3.0.
- Web-сервер (HTTP-сервер): будь-який із підтримкою HTTP 1.1, CGI, сервлетів і можливістю аутентифікації користувача. Web-сервер, що рекомендується – Payara 5.2021 і вище.
- Компілятор та засоби розробки мови Java: JDK 11 + Jakarta EE 9.

Вимоги до програмного забезпечення сервера доступу до даних:

- Операційна система: Ubuntu 20.04 LTS, Windows 10 Professional.
Операційна система, що рекомендується – Ubuntu 20.04.

- Підтримка EJB 3.0.

Вимоги до програмного забезпечення сервера баз даних:

- Операційна система: Ubuntu 20.04 LTS, Windows 10 Professional. Операційна система, що рекомендується – Ubuntu 20.04. Можливо використання СУБД на платформі Windows, якщо вона цілком підтримує стандарт SQL 92 і має JDBC-драйвер.
- Сервер баз даних: рекомендується використання MySQL 5.5 або MariaDB 10 (що повністю сумісний з ним). Інші можливі сервера: PostgreSQL, Sybase, Oracle, Informix.
- JDBC драйвера.

Вимоги до програмного забезпечення клієнтської машини:

- Операційна система: будь-яка операційна система з можливістю установки віддаленого мережного з'єднання і графічним інтерфейсом. Операційні системи, що рекомендуються – MS Windows 10, Linux.
- Web-оглядач: будь-який Web-оглядач із підтримкою HTML 5, JavaScript, CSS. Web-оглядачі, що рекомендуються: MS Edge, Google Chrome, Mozilla Firefox, Opera.

1.4.8 Вимоги до технічного забезпечення інформаційної системи Палацу творчості учнів

Інформаційна система включає кілька серверів (Web-сервер з бізнес логікою; сервер із сервісами доступу до даних; сервер баз даних) і клієнтські машини.

Вимоги до технічного забезпечення Web-сервера:

- Мікропроцесор класу Intel i5-2500 і вище.
- Оперативна пам'ять RAM 8 Гб і більше.
- Жорсткий диск з вільним місцем 20 Гб і більше.
- Мережна карта Ethernet (не менше 10 Мбіт/с).

Вимоги до технічного забезпечення сервера доступу до даних:

- Мікропроцесор класу Intel i5-2500 і вище.
- Оперативна пам'ять RAM 4 Гб і більше.
- Жорсткий диск з вільним місцем 15 Гб і більше.
- Мережна карта Ethernet (не менше 10 Мбіт/с).

Вимоги до технічного забезпечення сервера баз даних:

- Мікропроцесор класу Intel i5-2500 і вище.
- Оперативна пам'ять RAM 8 Гб і більше.
- Жорсткий диск з вільним місцем 40 Гб і більше.
- Мережна карта Ethernet (не менше 10 Мбіт/с).

Вимоги до технічного забезпечення клієнтської машини:

- Мікропроцесор класу Intel i3-550 і вище.
- Оперативна пам'ять RAM 4 Гб і більше.
- Жорсткий диск з вільним місцем 15 Гб і більше.
- Мережна карта Ethernet (не менше 10 Мбіт/с).

У якості мережного середовища можна використовувати кабелі типу вита пара категорії 5, 5е і вище. Мережне обладнання повинне підтримувати швидкість не менше 10 Мбіт/сек, бажано – 100 Мбіт/сек.

1.4.9 Вимоги до документації інформаційної системи Палацу творчості учнів

Необхідна наявність наступних документів:

- Загальний опис системи.
- Керівництво користувача.
- Інструкція з формування і ведення бази даних.
- Програма і методика випробувань.
- Текст програми.

1.4.10 Порядок контролю і приймання інформаційної системи Палацу творчості учнів

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

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

Найкраще для тестування розподілених систем застосовувати метод «знизу-нагору». Спочатку розробник повинен перевірити кожен компонент (елемент системи) окремо; потім – протестувати взаємодію компонента з іншими компонентами системи на автономному комп'ютері; і тільки потім – перевірити роботу всієї системи в розподіленому середовищі. Усі ці види тестування виконуються розробником системи без участі замовника.

Приймання системи виконується замовником. Система повинна вважатися придатною, якщо вона задовольняє всім пунктам даного технічного завдання (постановки задачі). Замовник і/або його представники самостійно проводять тестування усіх функцій системи методом «чорного ящика» у присутності розробника. При виявленні помилок у роботі системи складається їхній перелік і обговорюється термін їхнього виправлення розробником. Після цього замовник проводить повторне тестування.

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

РОЗДІЛ 2. РОЗРОБКА ПРОЕКТНИХ РІШЕНЬ ПО СИСТЕМІ ТА ЇЇ ЧАСТИНАХ

2.1 Загальносистемні рішення

2.1.1 Схема організаційної структури

Інформаційна система Палацу творчості учнів розробляється для використання співробітниками Палацу творчості учнів. Схема організаційної структури Палацу творчості учнів представлена на рисунку 2.1.

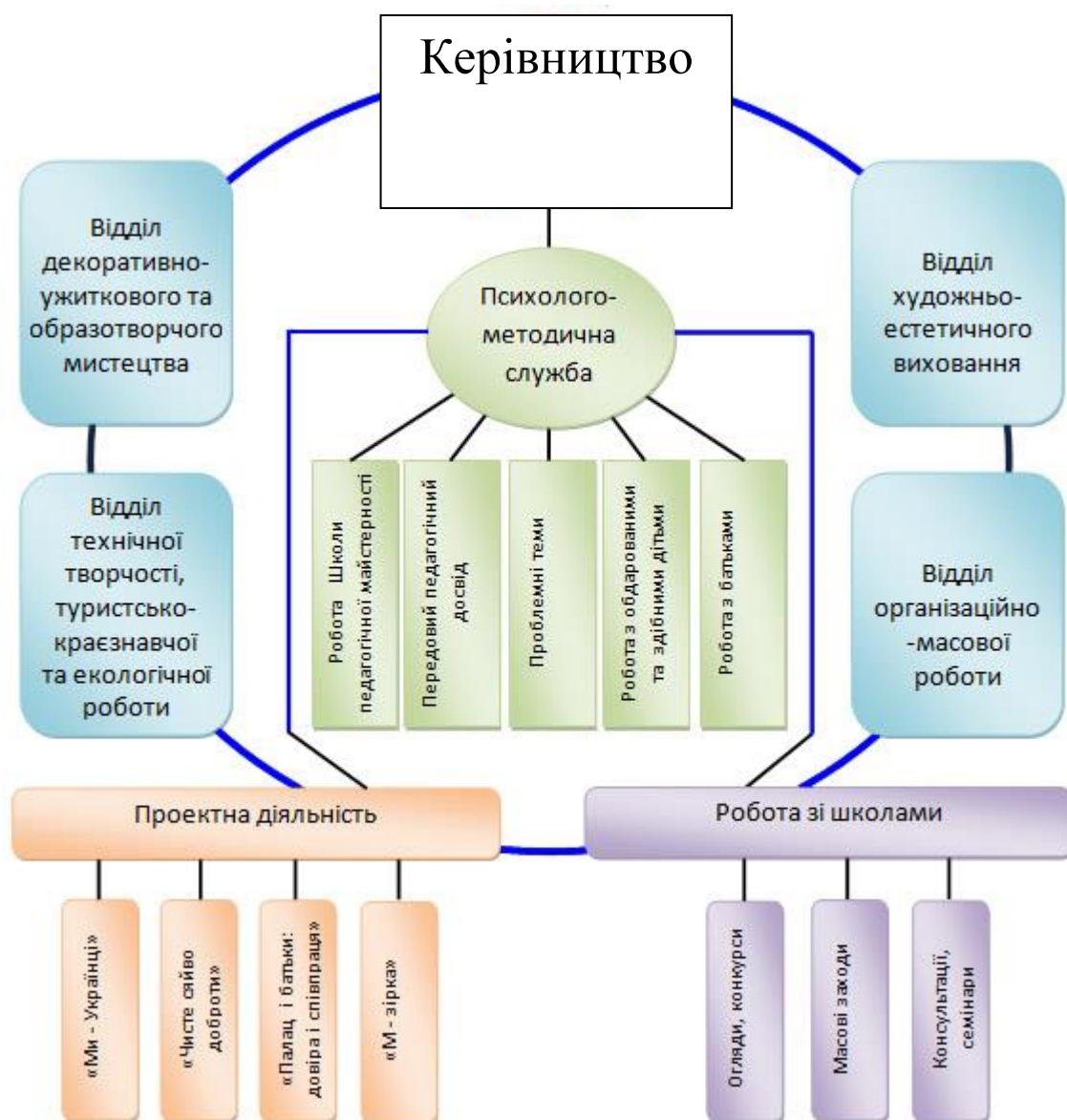


Рис. 2.1 – Схема організаційної структури Палацу творчості учнів

Користувачами системи, що розробляється, будуть співробітники Палацу творчості учнів. Адмініструвати систему може один з адміністраторів серверів закладу.

2.1.2 Схема функціональної структури

Для уточнення функціональності інформаційної системи, виявлення схеми функціональної структури, визначення підсистем (модулів), які входять до її складу, і вивчення динаміки роботи системи в цілому необхідно побудувати функціональну модель.

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

Діаграми потоків даних (Data Flow Diagrams – DFD) є основним засобом моделювання функціональних вимог системи при проектуванні. З їхньою допомогою ці вимоги розбиваються на функціональні компоненти (процеси) і представляються у виді мережі, зв'язаної потоками даних. Головна мета таких засобів – продемонструвати, як кожен процес перетворить свої вхідні дані у вихідні, а також, виявити відносини між цими процесами. Далі при побудові DFD буде використовуватися нотація Гейна-Сарсона (Gane-Sarson).

Джерела інформації (зовнішні сутності) породжують інформаційні потоки (потоки даних), що переносять інформацію до підсистем чи до процесів. Ті у свою чергу перетворюють інформацію і породжують нові потоки, що переносять інформацію до інших процесів чи підсистемам, накопичувачів даних чи зовнішніх сутностей – споживачів інформації. Таким чином, основними компонентами діаграм потоків даних є:

- зовнішні сутності;

- процеси;
- накопичувачі даних;
- потоки даних.

Зовнішня сутність являє собою матеріальний предмет чи фізичну особу, що представляє собою джерело чи приймач інформації [7]. Для системи, що розробляється, зовнішніми сутностями є:

- Педагог (користувач).
- Адміністратор (адміністратор системи).

Процес являє собою перетворення вхідних потоків даних у вихідні відповідно до певного алгоритму [7]. Фізично процес може бути реалізований різними способами.

Для системи, що розробляється, основні процеси на DFD пов'язані з аутентифікацією педагога, обліком успішності і відвідування, видачею завдань учням, наданням інформації про розклад, формуванням звітів для друку, записною книжкою педагога, а також із задачами адміністратора системи: ведення бази даних і налагодження системи для коректної аутентифікації.

Накопичувач даних являє собою абстрактний пристрій для збереження інформації, яку можна в будь-який момент помістити в накопичувач і через якийсь час знову взяти, причому способи запису і зчитування можуть бути будь-якими [7]. Накопичувачами даних у даній системі є:

- База даних системи (зберігає всю інформацію про успішність, відвідування, завдання, розклад і іншу).
- Файли записної книжки (містять записи педагогів, співробітників адміністрації та іншу інформацію).
- Файли аутентифікації (зберігають логіни та паролі користувачів системи).

Потік даних визначає інформацію, яка передається через деяке з'єднання від джерела до приймача. Реальний потік даних може бути інформацією, яка передається по кабелю між двома пристроями, листами, що пересилаються

поштою, магнітними стрічками чи дискетами, які переносять з одного комп'ютера на інший і таке інше [7].

Важливу специфічну роль у моделі грає спеціальний вид діаграми потоків даних – контекстна діаграма, що моделює систему найбільш загальним чином. Контекстна діаграма відображає інтерфейс системи з зовнішнім світом, а саме, інформаційні потоки між системою і зовнішніми сутностями, з якими вона повинна бути зв'язана. Контекстна діаграма однієї з основних підсистем «Робоче місце педагога» інформаційної системи позашкільного закладу представлена на рисунку 2.2.

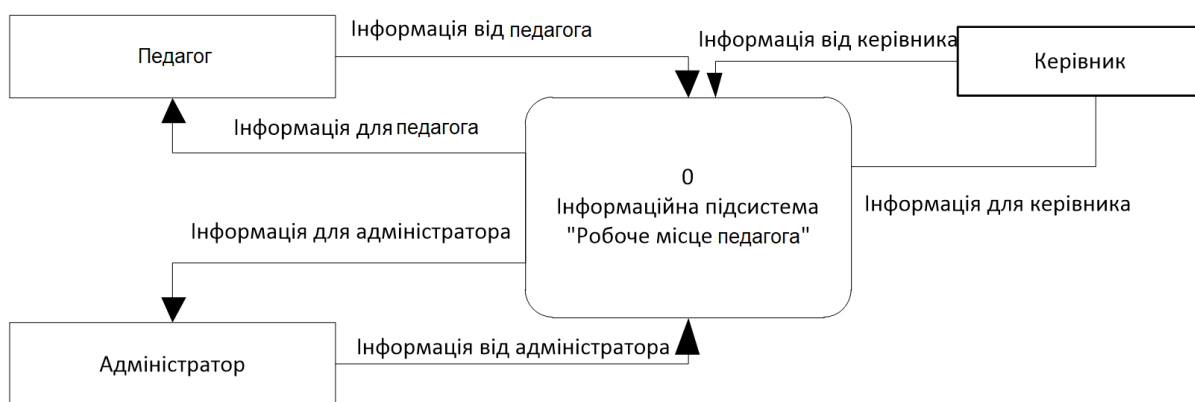


Рисунок 2.2 – Контекстна діаграма потоків даних
інформаційної підсистеми «Робоче місце педагога»

Проведемо декомпозицію потоків даних, показаних на рисунку 2.2 (див. таблицю 2.1).

Таблиця 2.1 – Декомпозиція потоків даних для контекстної DFD

Потоки даних на контекстній діаграмі	Потоки на діаграмі першого рівня
Інформація від педагога.	Інформація для аутентифікації. Інформація про успішність. Інформація про відвідування. Інформація про завдання. Інструкції. Інформація про записи.

Інформація для педагога.	Результат операції. Інформація про успішність. Інформація про відвідування. Інформація про завдання. Інформація про розклад. Звіти. Інформація про записи.
Інформація від керівника	Накази, розпорядження
Інформація для керівника	Звіти, відомості
Інформація від адміністратора.	Інформація. Інформація про аутентифікацію.
Інформація для адміністратора.	Інформація. Інформація про аутентифікацію. Результат операції.

Для продовження аналізу функціонального моделі системи будуються діаграми наступних рівнів. При цьому процеси піддаються декомпозиції, існуючі "абстрактні" потоки даних трансформуються в потоки, що представляють обмін даними на більш конкретному рівні.

DFD першого рівня будується як декомпозиція процесів, що присутні на контекстній діаграмі. DFD першого рівня представлена на рисунку 2.3.

Можна провести подальшу декомпозицію процесів на діаграмі потоків даних першого рівня, однак зараз опишемо тільки декомпозицію деяких процесів.

Процес «Вести облік успішності»:

- Вивести на екран успішність групи по конкретному предмету.
- Додати запис про успішність у базу даних.
- Змінити запис про успішність у базі даних.
- Вилучити запис про успішність з бази даних.

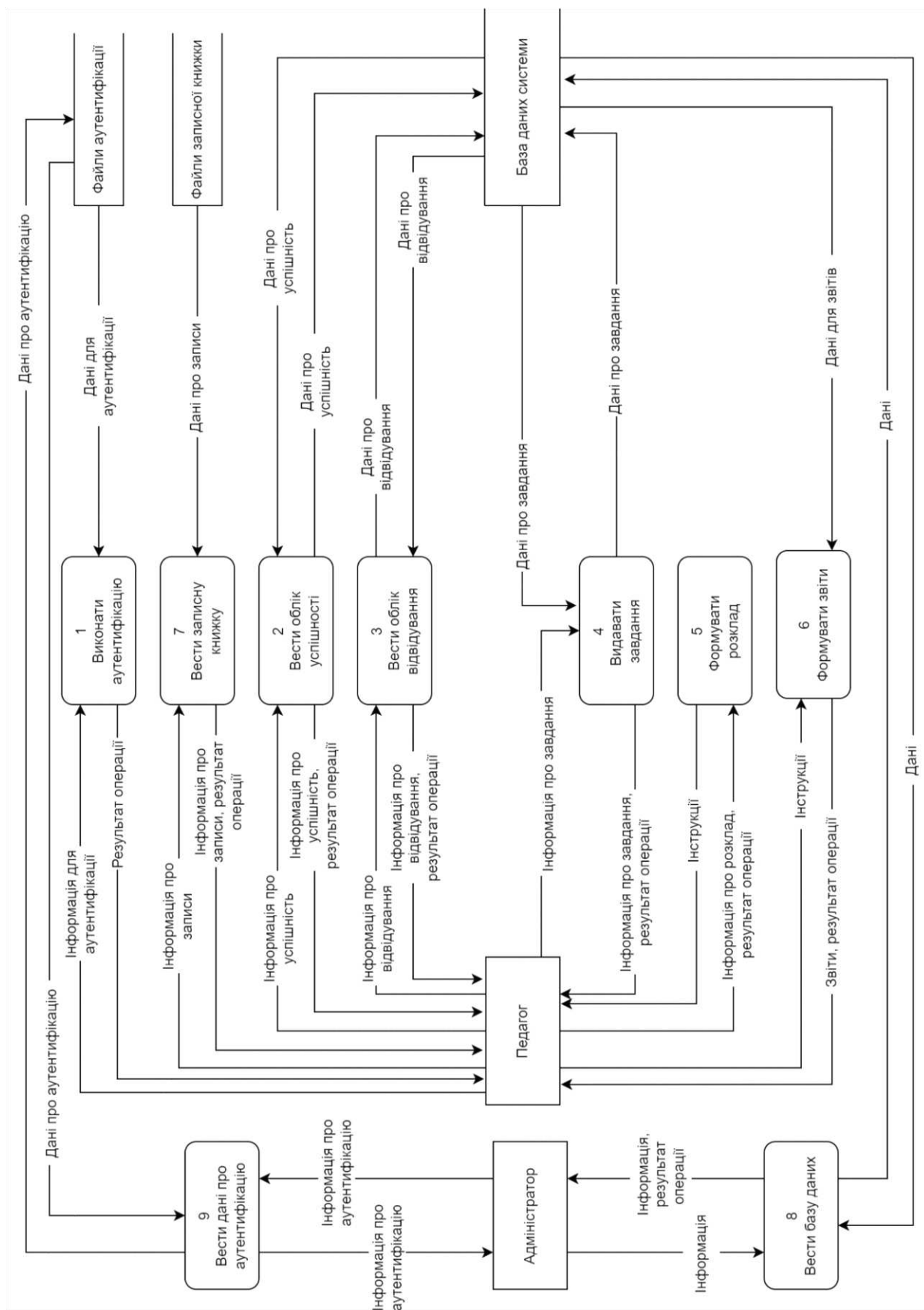


Рисунок 2.3 – Діаграма потоків даних першого рівня інформаційної системи Палацу творчості учнів

Процес «Вести облік відвідування»:

- Вивести на екран відвідування класу по конкретному предмету.
- Додати запис про відвідування у базу даних.
- Змінити запис про відвідування у базі даних.
- Вилучити запис про відвідування з бази даних.

Процес «Видавати завдання»:

- Вивести на екран завдання для класу по конкретному предмету.
- Додати запис про завдання в базу даних.
- Змінити запис про завдання в базі даних.
- Вилучити запис про завдання з бази даних.

Процес «Формувати звіти»:

- Вивести на екран звіт про успішність класу по конкретному предмету.
- Вивести на екран звіт про відвідування групи по конкретному предмету.
- Вивести на екран звіт про завдання для групи по конкретному предмету.

Процес «Вести записну книжку»:

- Додати запис у файл записної книжки.
- Змінити запис у файлі записної книжки.
- Вилучити запис із файлу записної книжки.

Таким чином, функціональність системи цілком визначена. Подальша декомпозиція процесів приведе до опису алгоритмів роботи окремих програмних модулів, що входять до складу програмного забезпечення системи.

2.1.3 Опис функцій, що автоматизуються

Первісні дані

При розробці функціональної частини проекту використовувалися наступні документи:

- Відомість з оцінками за виконані завдання.
- Відомість про відвідування занять учнями.
- Відомість про видані завдання.

Усі ці документи ведуться педагогами. Для кожного класу учнів і предмета заводяться окремі відомості. Складність полягає в тому, що стандартів на ці документи немає. Кожний педагог вирішує сам для себе як повинен виглядати кожен документ, оскільки вся інформація в цих документах використовується тільки самим педагогом для прийняття рішень. Опишемо кожен документ по окремої, включивши в нього необхідні елементи (поля, записи), що зустрічаються найбільш часто.

Документ «Відомість з оцінками за виконані завдання» повинна містити інформацію про педагога (прізвище, ім'я, по батькові), номер класу учнів, назву предмета, таблицю з оцінками. Таблиця з оцінками містить список учнів класу в першому стовпчику й інформацію про оцінки, завдання і дату здачі роботи в комірках таблиці. Інформацію про завдання зручно розмістити а верхньому рядку таблиці, а в комірках ставити тільки оцінку і дату здачі роботи.

Документ «Відомість про відвідування занять учнями» повинен містити інформацію про педагога (прізвище, ім'я, по батькові), номер класу учнів, назву предмета, таблицю з відмітками про відвідування. Таблиця з відмітками про відвідування містить список учнів класу в першому стовпчику, дати проведення занять у верхньому рядку та відмітки про відсутніх учнів в комірках таблиці.

Документ «Відомість про видані завдання» повинен містити інформацію про педагога (прізвище, ім'я, по батькові), номер класу учнів, назву предмета, таблицю з завданнями. Таблиця з завданнями містить для кожного завдання: номер завдання, вид завдання, назва, опис і термін здачі.

Цілі розробки системи і автоматизовані функції

Ціль розробки системи – збільшення продуктивності і поліпшення умов праці співробітників закладу освіти, а саме – адміністрації та педагогів шляхом

автоматизації функцій, які вони виконують. Оскільки головною діючою особою системи є педагог, розглянемо його функції, що можуть бути автоматизовані.

Автоматизовані функції, що виконує педагог:

- Облік успішності учнів. (Педагог ставить оцінки учням, що виконали (чи не виконали) завдання.)
- Облік відвідування занять учнями. (Педагог відмічає відсутніх на занятті учнів.)
- Видача завдань учням педагогами. (Педагог видає завдання учням.)
- Надання педагогу інформації про розклад його занять (організація робочого часу). (Педагог отримує інформацію про свій розклад.)
- Створення звітів про успішність, відвідування та видані завдання, які придатні для друку. (Педагог робить паперові копії інформації, що необхідна йому для прийняття рішень.)
- Ведення електронної записної книжки (щоденника) педагога. (Педагог записує необхідну йому інформацію на кожний день.)

Вигода від створення інформаційної системи позашкільного закладу освіти також полягає в тому, що інформація буде зберігатися в єдиній базі даних, вона буде систематизована і придатна для подальшої обробки. Цю інформацію зможуть надалі використовувати інші інформаційні системи.

2.1.4 Загальний опис системи

Призначення системи

Інформаційна система позашкільного закладу освіти – Палацу творчості учнів призначена для автоматизації задач, що виконують педагоги та керівництво закладу.

Система реалізує наступні функції:

- Аутентифікація користувачів.
- Облік успішності учнів.
- Облік відвідування занять учнями.
- Видача завдань учням педагогами.
- Надання педагогу інформації про розклад його занять (організація робочого часу).
- Створення звітів про успішність, відвідування та видані завдання, які придатні для друку.
- Ведення електронної записної книжки (щоденника) педагога.
- Функції, які виконує адміністратор системи: ведення бази даних і настройка аутентифікації системи.

Опис системи

Користувачами системи є педагоги та адміністрація Палацу творчості учнів. Також необхідний адміністратор системи, який повинен підтримувати роботу системи.

Інформаційне забезпечення системи складається з:

- Бази даних, у якій зберігається вся інформація про успішність, відвідування, завдання, розклад та інша необхідна інформація.
- Файлів електронної записної книжки, у яких зберігаються записи, зроблені педагогом на конкретні дні, а також інформація про вільні (немає записів) та зайняті дні (є запис).
- Файлів аутентифікації, у яких зберігаються логіни та паролі (у зашифрованому виді) користувачів системи. З цими файлами безпосередньо працює тільки адміністратор системи.

Система не вносить змін в організаційну структуру Палацу творчості учнів. Адміністратором системи може бути один з існуючих адміністраторів.

Технічне забезпечення системи складається з одного чи декількох серверів і робочих станцій, встановлених у навчальних класах чи лабораторіях. Для

максимальної продуктивності та масштабуємості системи бажано використовувати 3 сервери (хоча для коректної роботи системи досить і одного сервера, на який потрібно установити все програмне забезпечення):

- Web-сервер.
- Сервер застосувань (EJB-сервер).
- Сервер баз даних (БД).

Програмне забезпечення системи складається з декількох пакетів програм:

- Пакет програм для забезпечення функціональності, необхідної педагогам (описана вище).
- Пакет програм для адміністратора системи.

Кожен пакет програм складається з певної кількості програм (модулів), що реалізують окремі функції системи.

Бізнес-процеси реалізуються сервлетами та EJB-сторінками, що встановлюються на Web-сервері.

Доступ до бази даних здійснюється за допомогою EJB-об'єктів, наданих EJB серверами, і Enterprise Java Beans 3. EJB-сервер та EJB-об'єкти, які входять до них, встановлюються на сервер доступу до даних.

База даних є реляційною і може працювати під управлінням реляційної СУБД, що підтримує стандарт SQL-92 та має JDBC драйвера.

Операційна система, встановлена на Web-сервері і сервері доступу до даних, повинна мати реалізацію EJB серверів для цієї платформи (операційна система, що рекомендується – Ubuntu 20.04 LTS, у майбутньому можливий перехід на новішу ОС тієї ж лінійки). Сервер баз даних не має ніяких обмежень, крім тих, котрі накладає СУБД, яка використовується.

На Web-сервері системи повинен бути встановлений HTTP-сервер з підтримкою HTTP 1.1, JSP і можливістю аутентифікації користувача.

Опис взаємозв'язків системи з іншими системами

Система, що розробляється, зв'язана з інформаційно-управляючою системою загального освітнього простору міста, яка ще знаходиться у стадії тестової експлуатації. Частина таблиць бази дані системи аналогічна таблицям існуючої системи "Освітній простір", тому їх немає необхідності заповнювати. Можна реалізувати реплікацію чи звичайне копіювання цих таблиць (таблиці «Учень», «Педагог», «Дисципліна»).

Крім того, база даних системи позашкільного закладу освіти може використовуватися іншими інформаційними системами (наприклад, за допомогою системи «Освітній простір» можна отримувати інформацію про відвідування занять учнями певного гуртка і так далі).

2.1.5 Захист системи від несанкціонованого доступу

Аутентифікація

Аутентифікація – це процес, у ході якого на підставі пароля, ключа чи якої-небудь іншої інформації користувач підтверджує, що є саме тим, за кого себе видає.

При вході користувача (педагога) у систему першою справою відбувається його аутентифікація. Якщо введені логін і пароль збігаються з тими, що зберігаються в системі на сервері, то користувач успішно входить у систему, інакше йому відмовляється в доступі. Тут доречно контролювати кількість спроб, щоб уникнути підбора паролів.

Логін і пароль необхідно передавати винятково методом POST. При методі GET дані кешуються проксі-серверами і Web-оглядачами, тобто імовірність вкрати секретний пароль значно вище.

Наступним важливим питанням Web-аутентифікації є проблема аутентифікації при виклику кожної нової JSP-сторінки. Проблема полягає в тім, що пароль

користувач вводить тільки один раз при першому вході в систему, а аутентифікацію треба робити щораз, коли він звертається до будь-якої JSP-сторінки.

Найбільше часто використовуються наступні способи Web-аутентифікації:

- 1) Передавати логін і пароль кожній JSP («носити за собою»).
- 2) Записати логін і пароль у файл cookie.
- 3) Згенерувати ключ з певним терміном дії і передавати його кожній сторінці.
- 4) Аутентифікація засобами HTTP-сервера.

Перший спосіб незручний тим, що в системі можуть бути HTML-сторінки, яким можна передати параметри тільки методом GET. А тоді логін і пароль будуть закешовані. Перетворити всі HTML-сторінки у JSP і працювати тільки методом POST нерозумно, тому що штучно загальмовується робота сервера. Цей спосіб є ненадійним.

В другому і третьому способі логін і пароль передаються тільки одного разу, а cookie і ключ генеруються на певний час. Cookie – це параметри з певним терміном життя, що зберігаються на стороні клієнта. Ключ – це теж саме, тільки на стороні сервера. Cookie мають два принципових недоліки. По-перше, вони можуть бути або відключені, або не підтримуватися Web-оглядачем. По-друге, користувач може працювати на загальнодоступному комп'ютері і є ризик, що після нього хто-небудь увійде в систему з цього комп'ютера. Cookie можна встановлювати тільки на персональний комп'ютер користувача, за умови, що він точно упевнений, що до комп'ютера не мають доступу злоумисники. Цей спосіб є також ненадійним, оскільки педагог, як правило, буде працювати із системою, використовуючи загальнодоступний комп'ютер.

Третій спосіб більш надійний. Після введення логіна і пароля на сервері генерується ключ з певним терміном дії. Далі від сценарія до сценарія чи HTML-документа передається тільки ключ. Кожна JSP-сторінка, отримуючи ключ, по ключу отримує логін і пароль користувача на сервері. Цей спосіб надійний, але

складний у реалізації, оскільки необхідно розробити базу даних для збереження ключів, логіна і пароля, функції для генерації випадкових і унікальних ключів. JSP-сторінки будуть працювати повільніше через обов'язкове звертання до бази даних. Крім того, ключ генерується на певний час. Якщо час його дії малий, то користувачу прийдеться вводити логін і пароль повторно; якщо час дії великий, то існує імовірність того, що, коли користувач закінчить роботу із системою, злоумисник зможе увійти в систему.

Четвертий спосіб використовує вбудовані засоби HTTP-сервера. Сервер видає вікно з полями для введення логіна і пароля, що передаються на сервер у зашифрованому виді. Сервер порівнює отримані дані з тими, що зберігаються в його файлах. Після цього користувач отримує доступ до всіх файлів, що знаходяться у визначеній папці (JSP-сторінки, HTML-документи і так далі). Для виходу із системи потрібно буде просто закрити вікно Web-оглядача. Цей спосіб надійний, а його реалізація вимагає мінімальних витрат часу. Робота JSP-сторінок не буде сповільнюватися звертаннями до бази даних (як у попередньому способі), а логін для виконання наступної ідентифікації користувача можна буде отримати зі стандартної змінної оточення HTTP-сервера.

Таким чином, найбільш ефективним з погляду захищеності, ефективності і простоти реалізації є четвертий спосіб – аутентифікація за допомогою вбудованих засобів HTTP-сервера. Саме його використовує інформаційна система Палацу творчості учнів.

Крім аутентифікації на HTTP-сервері, необхідно робити аутентифікацію на сервері доступу до даних і сервері баз даних. На сервері доступу до даних (EJB-компоненти) аутентифікація налагоджується за допомогою утиліти конфігурування JNDI (Java Naming and Directory Interface). Ця утиліта записує до реєстру EJB-сервера параметри, що визначають, де дозволено запускати компонент і хто отримає до нього доступ. Аутентифікація на сервері баз даних реалізується засобами СУБД.

Ідентифікація

Ідентифікація – це процес, у ході якого з'ясовуються права доступу, привілеї, властивості і характеристики користувача на підставі його імені, логіна чи якої-небудь іншої інформації про нього.

Після проходження аутентифікації, яка здійснюється вбудованими засобами HTTP-сервера, будь-яка JSP-сторінка може отримати логін користувача зі змінних оточення сервера і зробити ідентифікацію.

Доступ до будь-яких даних у системі Палацу творчості учнів (крім загальнодоступних) відбувається вибіркою даних із джерел даних з використанням логіна педагога чи керівника. Таким чином, при відсутності логіна неможливо отримати необхідні дані. При наявності логіна можна отримати тільки необхідні дані.

Захист каналів зв'язку

Логін і пароль педагога повинні передаватися по захищеному з'єднанню (у зашифрованому виді), що створює HTTP-сервер.

Інша інформація (інформація про успішність, відвідування, завдання та інша) є такою, що складно інтерпретується, тому питання з її захистом не так актуально. Більш того, інші системи можуть надавати цю інформацію для ознайомлення учням і іншим зацікавленим особам. Важливо, щоб ніхто не міг змінювати, додавати і вилучати цю інформацію в базі даних, крім відповідних педагогів, що забезпечується процесами аутентифікації й ідентифікації.

Однак у майбутньому в систему можуть бути додані нові дані (і функції), які потрібно захищати (зараз вже є інформація в записній книжці педагога, що є особистою і повинна бути захищена від перегляду).

Умовно канали зв'язку в системі можна розділити на кілька видів:

- Між користувачами і Web-сервером. Захист, що рекомендується: захищене з'єднання з використанням SSL (Secure Sockets Layer). При використанні SSL сервер і Web-оглядач погодять між собою застосований метод шифрування, обмінюючись сертифікатами і ключами і встановлюючи доступне тільки їм захищене з'єднання.
- Між Web-сервером і сервером доступу до даних. Захист, що рекомендується: захист EJB. При цьому захист даних можна також здійснити на програмному рівні.
- Між сервером доступу до даних і сервером баз даних. Захист, що рекомендується: засобами СУБД, яка використовується, SSL.

Необхідно розуміти, що захист каналів зв'язку пов'язаний з додатковими операціями по шифруванню даних, обміну ключами і сертифікатами, тому продуктивність роботи системи може погіршитися.

2.2 Рішення з організаційного забезпечення

Інформаційна система позашкільного закладу освіти розробляється для Палацу творчості учнів. Схема організаційної структури Палацу творчості учнів була представлена на рисунку 2.1.

Користувачами підсистеми, що розробляється, буде керівництво Палацу творчості учнів. Адмініструвати систему повинен один з адміністраторів.

Таким чином, зміни в організаційній структурі управління та в організації існуючих підрозділів не передбачено.

2.3 Рішення з інформаційного забезпечення

2.3.1 Опис інформаційного забезпечення системи

Склад інформаційного забезпечення

Інформаційне забезпечення системи складається з:

- Бази даних, у якій зберігається вся інформація про успішність, відвідування, завдання, розклад та інша необхідна інформація.
- Файлів електронної записної книжки, у яких зберігаються записи, зроблені педагогом на конкретні дні, а також інформація про вільні (немає записи) та зайняті дні (є запис).
- Файлів аутентифікації, у яких зберігаються логіни та паролі (у зашифрованому виді) користувачів системи. З цими файлами безпосередньо працює тільки адміністратор системи.

Організація інформаційного забезпечення

Файли записної книжки і файли аутентифікації є звичайними текстовими файлами, що зберігаються у файловій системі на жорсткому диску Web-сервера. Розмір вільного місця жорсткого диска повинен бути не менше 5 Гб. Структура файлів буде описана в наступних підрозділах.

База даних повинна зберігатися в реляційній СУБД, що підтримує стандарт SQL-92. Рекомендується використання СУБД MySQL 5.5 або MariaDB 10. База даних розташовується на жорсткому диску сервера баз даних. Розмір вільного місця жорсткого диска повинен бути не менше 8 Гб (оцінка отримана шляхом проведення розрахунку).

Організація збору і передачі інформації

Введення інформації в базу даних системи, файли записної книжки і файли аутентифікації повинно здійснюватися за допомогою програмного забезпечення системи.

Наступна інформації повинна бути введена в систему до початку її експлуатації:

- Інформація про учнів, педагогів, розклад, предмети. За збір цієї інформації відповідає адміністратор системи, якому можуть допомагати оператори.
- Інформація про аутентифікацію. Адміністратор повинен ввести інформацію про логіни та паролі вчителів. За збір і збереження цієї інформації відповідає адміністратор системи.

Інша інформація (про успішність, відвідування, завдання, інформація в записній книжці) вводиться педагогами протягом всього навчального процесу.

Адміністратор системи, як правило, вводить інформацію локально (безпосередньо використовуючи Web-сервер або сервер баз даних), а педагог – віддалено, використовуючи робочі станції і Web-оглядач.

2.3.2 Концептуальна модель даних

З аналізу функціональної моделі системи (дивись пункт 2.1.2) можна побудувати концептуальну модель даних системи.

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

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

Перший крок побудови концептуальної моделі даних – виділення сутностей.

Можна виділити наступні сутності:

- Педагог.

- Група педагогів (напря́м, підрозді́л).
- Розклад.
- Тиждень.
- День.
- Заняття (урок/пара).
- Учень.
- Клас.
- Дисциплі́на.
- Відвідування.
- Завдання.
- Оці́нка.
- Успі́шність.

Наступний крок побудови моделі – ідентифікація атрибутів.

Визначимо атрибути сутностей:

Сутність «Педагог».

- Код педагога.
- Прізвище.
- Ім'я.
- По батькові.

Сутність «Група педагогів».

- Код групи.
- Код предмета.
- Код групи.

Сутність «Розклад».

- Код педагога.
- Код тижня.
- Код дня.
- Номер уроку.

- Код гуртка.
- Код предмета.
- Вид заняття.
- Аудиторія.

Сутність «Тиждень».

- Код тижня.
- Опис.

Сутність «День».

- Код дня.
- Назва.

Сутність «Урок».

- Номер уроку.
- Час початку.
- Час закінчення.

Сутність «Учень».

- Код учня.
- Код класу.
- Прізвище.
- Ім'я.
- По батькові.

Сутність «Клас».

- Код класу.
- Номер класу.
- Кількість учнів.

Сутність «Дисципліна».

- Код дисципліни.
- Назва.

Сутність «Заняття».

- Код заняття.
- Код групи педагогів.

- Дата.
- Номер уроку.
- Вид заняття.

Сутність «Відвідування».

- Код заняття.
- Код учня.

Сутність «Завдання».

- Код завдання.
- Код групи педагогів.
- Вид завдання.
- Номер.
- Опис.
- Термін виконання.

Сутність «Оцінка».

- Код оцінки.
- Оцінка.

Сутність «Успішність».

- Код учня.
- Код завдання.
- Код педагога.
- Код оцінки.
- Дата.

З урахуванням наявної інформації побудуємо ER-діаграму (рисунок 2.4) у нотації IE (Information Engineering).

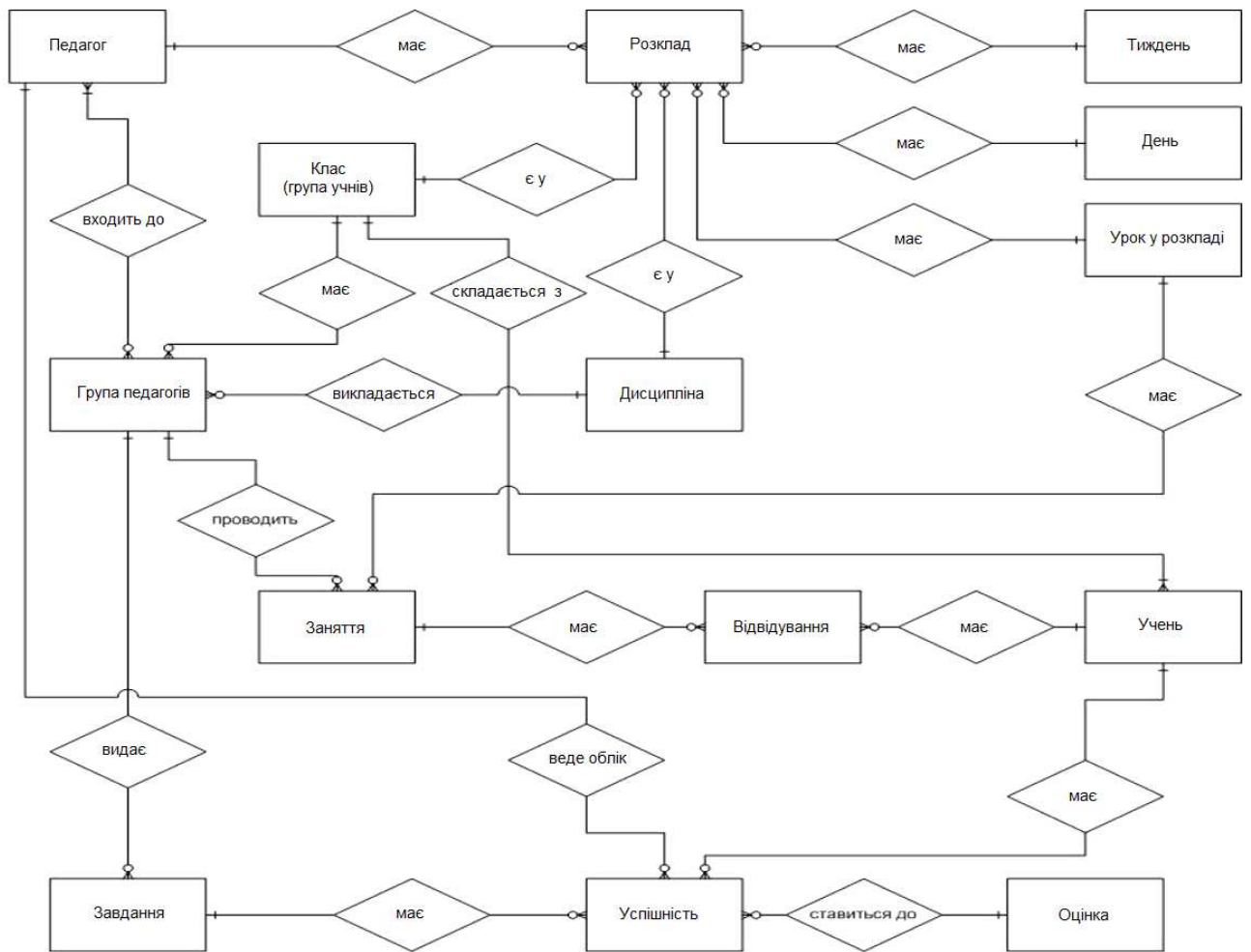


Рисунок 2.4 – Концептуальна модель даних

Опис зв'язків між сутностями (див. рисунок 2.4) представлений нижче.

Зв'язок «Педагоги входять до групи педагогів».

Опис: кожен педагог може входити до складу однієї чи декількох груп. Він також може не належати жодній групі.

Сутність 1: «Педагог».

Сутність 2: «Група педагогів».

Відношення: багато до багатьох.

Сутність 1 → Сутність 2; роль: може входити до; мін., макс.: 0, n.

Сутність 2 → Сутність 1; роль: складається з; мін., макс.: 1, n.

Зв'язок «Педагог має розклад(и)».

Опис: у кожного педагога може бути розклад.

Сутність 1: «Педагог».

Сутність 2: «Розклад».

Відношення: один до багатьох.

Сутність 1 $\rightarrow$ Сутність 2; роль: може мати; мін., макс.: 0, п.

Сутність 2 $\rightarrow$ Сутність 1; роль: відноситься до; мін., макс.: 1, 1.

Зв'язок «Тиждень має розклад(и)».

Опис: на кожний тиждень може бути складений розклад.

Сутність 1: «Тиждень».

Сутність 2: «Розклад».

Відношення: один до багатьох.

Сутність 1 $\rightarrow$ Сутність 2; роль: може мати; мін., макс.: 0, п.

Сутність 2 $\rightarrow$ Сутність 1; роль: складений на; мін., макс.: 1, 1.

Зв'язок «День має розклад(и)».

Опис: на кожний день може бути складений розклад.

Сутність 1: «День».

Сутність 2: «Розклад».

Відношення: один до багатьох.

Сутність 1 $\rightarrow$ Сутність 2; роль: може мати; мін., макс.: 0, п.

Сутність 2 $\rightarrow$ Сутність 1; роль: складений на; мін., макс.: 1, 1.

Зв'язок «Урок має розклад(и)».

Опис: на кожен урок може бути складений розклад.

Сутність 1: «Урок».

Сутність 2: «Розклад».

Відношення: один до багатьох.

Сутність 1 $\rightarrow$ Сутність 2; роль: може мати; мін., макс.: 0, п.

Сутність 2 $\rightarrow$ Сутність 1; роль: складений на; мін., макс.: 1, 1.

Зв'язок «Клас є у розкладі(ах)».

Опис: кожен клас може бути включений у розклад вчителя.

Сутність 1: «Клас».

Сутність 2: «Розклад».

Відношення: один до багатьох.

Сутність 1 → Сутність 2; роль: може бути у; мін., макс.: 0, п.

Сутність 2 → Сутність 1; роль: відноситься до; мін., макс.: 1, 1.

Зв'язок «Дисципліна є у розкладі(ах)».

Опис: кожна дисципліна може бути включена у розклад вчителя.

Сутність 1: «Дисципліна».

Сутність 2: «Розклад».

Відношення: один до багатьох.

Сутність 1 → Сутність 2; роль: може бути у; мін., макс.: 0, п.

Сутність 2 → Сутність 1; роль: включає; мін., макс.: 1, 1.

Зв'язок «Клас має групи педагогів».

Опис: кожен клас (учнів) може мати кілька груп вчителів, які викладають предмети.

Сутність 1: «Клас».

Сутність 2: «Група педагогів».

Відношення: один до багатьох.

Сутність 1 → Сутність 2; роль: має; мін., макс.: 0, п.

Сутність 2 → Сутність 1; роль: викладає в; мін., макс.: 1, 1.

Зв'язок «Клас складається з учнів».

Опис: кожен клас складається з одного чи декількох учнів.

Сутність 1: «Клас».

Сутність 2: «Учень».

Відношення: один до багатьох.

Сутність 1 → Сутність 2; роль: складається зі; мін., макс.: 1, п.

Сутність 2 → Сутність 1; роль: входить до; мін., макс.: 1, 1.

Зв'язок «Дисципліна викладається групами педагогів».

Опис: кожна дисципліна може викладатися декількома групами педагогів.

Сутність 1: «Дисципліна».

Сутність 2: «Група вчителів».

Відношення: один до багатьох.

Сутність 1 → Сутність 2; роль: викладається; мін., макс.: 0, п.

Сутність 2 → Сутність 1; роль: викладає; мін., макс.: 1, 1.

Зв'язок «Група педагогів проводить заняття».

Опис: кожна група педагогів може проводити заняття.

Сутність 1: «Група вчителів».

Сутність 2: «Заняття».

Відношення: один до багатьох.

Сутність 1 → Сутність 2; роль: проводить; мін., макс.: 0, п.

Сутність 2 → Сутність 1; роль: проводиться; мін., макс.: 1, 1.

Зв'язок «Урок має заняття».

Опис: на кожному уроці можуть проводитися заняття.

Сутність 1: «Урок».

Сутність 2: «Заняття».

Відношення: один до багатьох.

Сутність 1 → Сутність 2; роль: має; мін., макс.: 0, п.

Сутність 2 → Сутність 1; роль: проводиться на; мін., макс.: 1, 1.

Зв'язок «Заняття має відвідування».

Опис: на кожному занятті може проставлятися відвідування (відмічатися відсутні учні).

Сутність 1: «Заняття».

Сутність 2: «Відвідування».

Відношення: один до багатьох.

Сутність 1 $\rightarrow$ Сутність 2; роль: має; мін., макс.: 0, п.

Сутність 2 $\rightarrow$ Сутність 1; роль: відноситься до; мін., макс.: 1, 1.

Зв'язок «Учень має відвідування».

Опис: кожен учень може мати відвідування (не бути або бути присутнім на занятті).

Сутність 1: «Учень».

Сутність 2: «Відвідування».

Відношення: один до багатьох.

Сутність 1 $\rightarrow$ Сутність 2; роль: має; мін., макс.: 0, п.

Сутність 2 $\rightarrow$ Сутність 1; роль: відноситься до; мін., макс.: 1, 1.

Зв'язок «Група педагогів видає завдання».

Опис: кожна група педагогів може видавати завдання учням.

Сутність 1: «Група педагогів».

Сутність 2: «Завдання».

Відношення: один до багатьох.

Сутність 1 $\rightarrow$ Сутність 2; роль: видає; мін., макс.: 0, п.

Сутність 2 $\rightarrow$ Сутність 1; роль: видається; мін., макс.: 1, 1.

Зв'язок «Педагог веде облік успішності».

Опис: кожен педагог може вести облік успішності учнів.

Сутність 1: «Педагог».

Сутність 2: «Успішність».

Відношення: один до багатьох.

Сутність 1 $\rightarrow$ Сутність 2; роль: веде облік; мін., макс.: 0, п.

Сутність 2 $\rightarrow$ Сутність 1; роль: проставляється; мін., макс.: 1, 1.

Зв'язок «Учень має успішність».

Опис: кожен учень може мати успішність (оцінки за виконанні завдання).

Сутність 1: «Учень».

Сутність 2: «Успішність».

Відношення: один до багатьох.

Сутність 1 $\rightarrow$ Сутність 2; роль: має; мін., макс.: 0, п.

Сутність 2 $\rightarrow$ Сутність 1; роль: відноситься до; мін., макс.: 1, 1.

Зв'язок «Завдання має успішність».

Опис: кожне завдання може мати успішність (за його виконання можуть виставлятися оцінки).

Сутність 1: «Завдання».

Сутність 2: «Успішність».

Відношення: один до багатьох.

Сутність 1 $\rightarrow$ Сутність 2; роль: має; мін., макс.: 0, п.

Сутність 2 $\rightarrow$ Сутність 1; роль: відноситься до; мін., макс.: 1, 1.

Зв'язок «Оцінка ставиться до успішності».

Опис: кожна оцінка може бути поставлена до успішності (за виконання завдання).

Сутність 1: «Оцінка».

Сутність 2: «Успішність».

Відношення: один до багатьох.

Сутність 1 $\rightarrow$ Сутність 2; роль: ставиться до; мін., макс.: 0, п.

Сутність 2 $\rightarrow$ Сутність 1; роль: має; мін., макс.: 1, 1.

2.3.3 Логічна та фізична структура бази даних

Виходячи з аналізу концептуальної моделі даних системи, розробимо логічну структуру бази даних. Оскільки використовується СУБД MySQL, будемо одразу визначати типи даних полів таблиць відповідно до правил цієї СУБД.

Розробимо фізичну структуру кожної таблиці, що входить у базу даних. Типи даних будемо вказувати для СУБД MySQL 5.5.

При описі фізичних таблиць бази даних використані стандартні позначення ключів:

- PK (первинний ключ).
- ALT (альтернативний ключ).
- FK (зовнішній ключ).

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

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

Структура фізичної таблиці «Вчитель» представлена в таблиці 2.2. Таблиця «Вчитель» знаходиться в 3 НФ, оскільки дані полів, що не входять у первинний ключ, не залежать один від одного.

Структура фізичної таблиці «Тиждень» представлена в таблиці 2.3. Таблиця «Тиждень» знаходиться в 3 НФ, оскільки дані полів, що не входять у первинний ключ, не залежать один від одного.

Структура фізичної таблиці «День» представлена в таблиці 2.4. Таблиця «День» знаходиться в 3 НФ, оскільки дані полів, що не входять у первинний ключ, не залежать один від одного.

Структура фізичної таблиці «Урок» представлена в таблиці 2.5. Таблиця «Уроки» не знаходиться в 3 НФ, оскільки значення поля «Час закінчення» може

бути обчислене за значенням поля «Час початку» при відомому постійному значенні тривалості уроку.

Структура фізичної таблиці «Розклад» представлена в таблиці 2.6. Таблиця «Розклад» знаходиться в 3 НФ, оскільки дані полів, що не входять у первинний ключ, не залежать один від одного.

Структура фізичної таблиці «Клас» представлена в таблиці 2.7. Таблиця «Клас» знаходиться в 3 НФ, оскільки дані полів, що не входять у первинний ключ, не залежать один від одного.

Структура фізичної таблиці «Група вчителів» представлена в таблиці 2.8. Таблиця «Група вчителів» знаходиться в 3 НФ, оскільки дані полів, що не входять у первинний ключ, не залежать один від одного.

Структура фізичної таблиці «Вчитель у класі» представлена в таблиці 2.9. Таблиця « Вчитель у класі » знаходиться в 3 НФ, оскільки дані полів, що не входять у первинний ключ, не залежать один від одного.

Структура фізичної таблиці «Дисципліна» представлена в таблиці 2.10. Таблиця «Дисципліна» знаходиться в 3 НФ, оскільки дані полів, що не входять у первинний ключ, не залежать один від одного.

Структура фізичної таблиці «Заняття» представлена в таблиці 2.11. Таблиця «Заняття» знаходиться в 3 НФ, оскільки дані полів, що не входять у первинний ключ, не залежать один від одного.

Структура фізичної таблиці «Учень» представлена в таблиці 2.12. Таблиця « Учень » знаходиться в 3 НФ, оскільки дані полів, що не входять у первинний ключ, не залежать один від одного.

Структура фізичної таблиці «Відвідування» представлена в таблиці 2.13. Таблиця «Відвідування» знаходиться в 3 НФ, оскільки дані полів, що не входять у первинний ключ, не залежать один від одного.

Структура фізичної таблиці «Завдання» представлена в таблиці 2.14. Таблиця «Завдання» знаходиться в 3 НФ, оскільки дані полів, що не входять у первинний ключ, не залежать один від одного.

Структура фізичної таблиці «Вид завдання» представлена в таблиці 2.15. Таблиця «Вид завдання» не знаходиться в 3 НФ, оскільки значення поля «Скорочена назва» визначає значення поля «Назва» і навпаки.

Структура фізичної таблиці «Оцінка» представлена в таблиці 2.16. Таблиця «Оцінка» знаходиться в 3 НФ, оскільки дані полів, що не входять у первинний ключ, не залежать один від одного.

Структура фізичної таблиці «Успішність» представлена в таблиці 2.17. Таблиця «Успішність» знаходиться в 3 НФ, оскільки дані полів, що не входять у первинний ключ, не залежать один від одного.

Структура фізичної таблиці «Вид заняття» представлена в таблиці 2.18. Таблиця «Вид заняття» не знаходиться в 3 НФ, оскільки значення поля «Скорочена назва» визначає значення поля «Назва» і навпаки.

Таблиця 2.2 – Структура таблиці «Педагог» / «Teacher»

Назва поля (українською)	Назва поля (англійською)	Тип даних	Дов- жина, байти	Ключ	Примітка
Код педагога	code_teacher	int	4	PK	
Прізвище	last_name	varchar(30)	30	ALT	
Ім'я	first_name	varchar(30)	30	ALT	
По батькові	middle_name	varchar(30)	30	ALT	

Таблиця 2.3 – Структура таблиці «Тиждень» / «Week»

Назва поля (українською)	Назва поля (англійською)	Тип даних	Дов- жина, байти	Ключ	Примітка
Код тижня	code_week	tinyint	1	PK	
Опис	description	varchar(50)	50	ALT	

Таблиця 2.4 – Структура таблиці «День» / «Day»

Назва поля (українською)	Назва поля (англійською)	Тип даних	Дов- жина, байти	Ключ	Примітка
Код дня	code_day	tinyint	1	PK	
Назва	name	varchar(15)	15	ALT	

Таблиця 2.5 – Структура таблиці «Урок» / «Class»

Назва поля (українською)	Назва поля (англійською)	Тип даних	Дов- жина, байти	Ключ	Примітка
Номер уроку	code_class	tinyint	1	PK	
Час початку	begin_time	smalldatetime	4	ALT	
Час закінчення	end_time	smalldatetime	4	ALT	

Таблиця 2.6 – Структура таблиці «Розклад» / «Schedule»

Назва поля (українською)	Назва поля (англійською)	Тип даних	Дов- жина, байти	Ключ	Примітка
Код педагога	code_teacher	int	4	PK, FK	
Код тижня	code_week	tinyint	1	PK, FK	
Код дня	code_day	tinyint	1	PK, FK	
Номер уроку	class	tinyint	1	PK, FK	
Код класу	code_group	smallint	2	FK	
Код дисципліни	code_subject	smallint	2	FK	
Код виду заняття	code_class_type	tinyint	1	FK	
Аудиторія	room	varchar(10)	10		

Таблиця 2.7 – Структура таблиці «Клас» / «Group»

Назва поля (українською)	Назва поля (англійською)	Тип даних	Дов- жина, байти	Ключ	Примітка
Код класу	code_group	smallint	2	PK	
Номер класу	num_group	varchar(10)	10	ALT	
Кількість учнів	quantity	tinyint	1		

Таблиця 2.8 – Структура таблиці «Група вчителів» / «CoTeachersGroup»

Назва поля (українською)	Назва поля (англійською)	Тип даних	Дов- жина, байти	Ключ	Примітка
Код групи педагогів	code_coteachers	smallint	2	PK	
Код дисципліни	code_subject	smallint	2	FK	
Код класу	code_group	smallint	2	FK	

Таблиця 2.9 – Структура таблиці «Вчитель у групі» / «CoTeachers»

Назва поля (українською)	Назва поля (англійською)	Тип даних	Дов- жина, байти	Ключ	Примітка
Код групи вчителів	code_coteachers	smallint	2	PK, FK	
Код вчителя	code_teacher	int	4	PK, FK	

Таблиця 2.10 – Структура таблиці «Дисципліна» / «Subject»

Назва поля (українською)	Назва поля (англійською)	Тип даних	Дов- жина, байти	Ключ	Примітка
Код дисципліни	code_subject	smallint	2	PK	
Назва	name	varchar(100)	100	ALT	

Таблиця 2.11 – Структура таблиці «Заняття» / «AttendanceTime»

Назва поля (українською)	Назва поля (англійською)	Тип даних	Дов- жина, байти	Ключ	Примітка
Код заняття	code_attendance	int	4	PK	
Код групи вчителів	code_coteachers	smallint	2	ALT, FK	
Дата	date	smalldatetime	4	ALT	
Код виду заняття	code_class_type	tinyint	1	ALT	
Номер уроку	code_class	tinyint	1	ALT, FK	

Таблиця 2.12 – Структура таблиці «Учень» / «Student»

Назва поля (українською)	Назва поля (англійською)	Тип даних	Дов- жина, байти	Ключ	Примітка
Код учня	code_student	int	4	PK	
Прізвище	last_name	varchar(30)	30	ALT	
Ім'я	first_name	varchar(30)	30	ALT	
По батькові	middle_name	varchar(30)	30	ALT	
Код групи	code_group	smallint	2	FK	

Таблиця 2.13 – Структура таблиці «Відвідування» / «Attendance»

Назва поля (українською)	Назва поля (англійською)	Тип даних	Довжина, байти	Ключ	Примітка
Код заняття	code_attendance	int	4	PK, FK	
Код учня	code_student	int	4	PK, FK	

Таблиця 2.14 – Структура таблиці «Завдання» / «Task»

Назва поля (українською)	Назва поля (англійською)	Тип даних	Довжина, байти	Ключ	Примітка
Код завдання	code_task	smallint	2	PK	
Код групи вчителів	code_coteachers	smallint	2	ALT,FK	
Код виду завдання	code_task_type	tinyint	1	ALT,FK	
Номер	number	tinyint	1	ALT	
Опис	description	varchar(80)	80	ALT	
Термін виконання	deadline	smalldatetime	4		

Таблиця 2.15 – Структура таблиці «Вид завдання» / «TaskType»

Назва поля (українською)	Назва поля (англійською)	Тип даних	Довжина, байти	Ключ	Примітка
Код виду завдання	code_task_type	tinyint	1	PK	
Скорочена назва	abbreviation	varchar(10)	10	ALT 1	
Назва	name	varchar(50)	50	ALT 2	

Таблиця 2.16 – Структура таблиці «Оцінка» / «Grade»

Назва поля (українською)	Назва поля (англійською)	Тип даних	Дов- жина, байти	Ключ	Примітка
Код оцінки	code_grade	tinyint	1	PK	
Оцінка	grade	varchar(10)	10	ALT	

Таблиця 2.17 – Структура таблиці «Успішність» / «Progress»

Назва поля (українською)	Назва поля (англійською)	Тип даних	Дов- жина, байти	Ключ	Примітка
Код учня	code_student	int	4	PK, FK	
Код завдання	code_task	smallint	2	PK, FK	
Код вчителя	code_teacher	int	4	FK	
Код оцінки	code_grade	tinyint	1	FK	
Дата	date	smalldatetime	4		

Таблиця 2.18 – Структура таблиці «Вид заняття» / «ClassType»

Назва поля (українською)	Назва поля (англійською)	Тип даних	Дов- жина, байти	Ключ	Примітка
Код виду заняття	code_class_type	tinyint	1	PK	
Скорочена назва	abbreviation	varchar(10)	10	ALT 1	
Назва	name	varchar(20)	20	ALT 2	

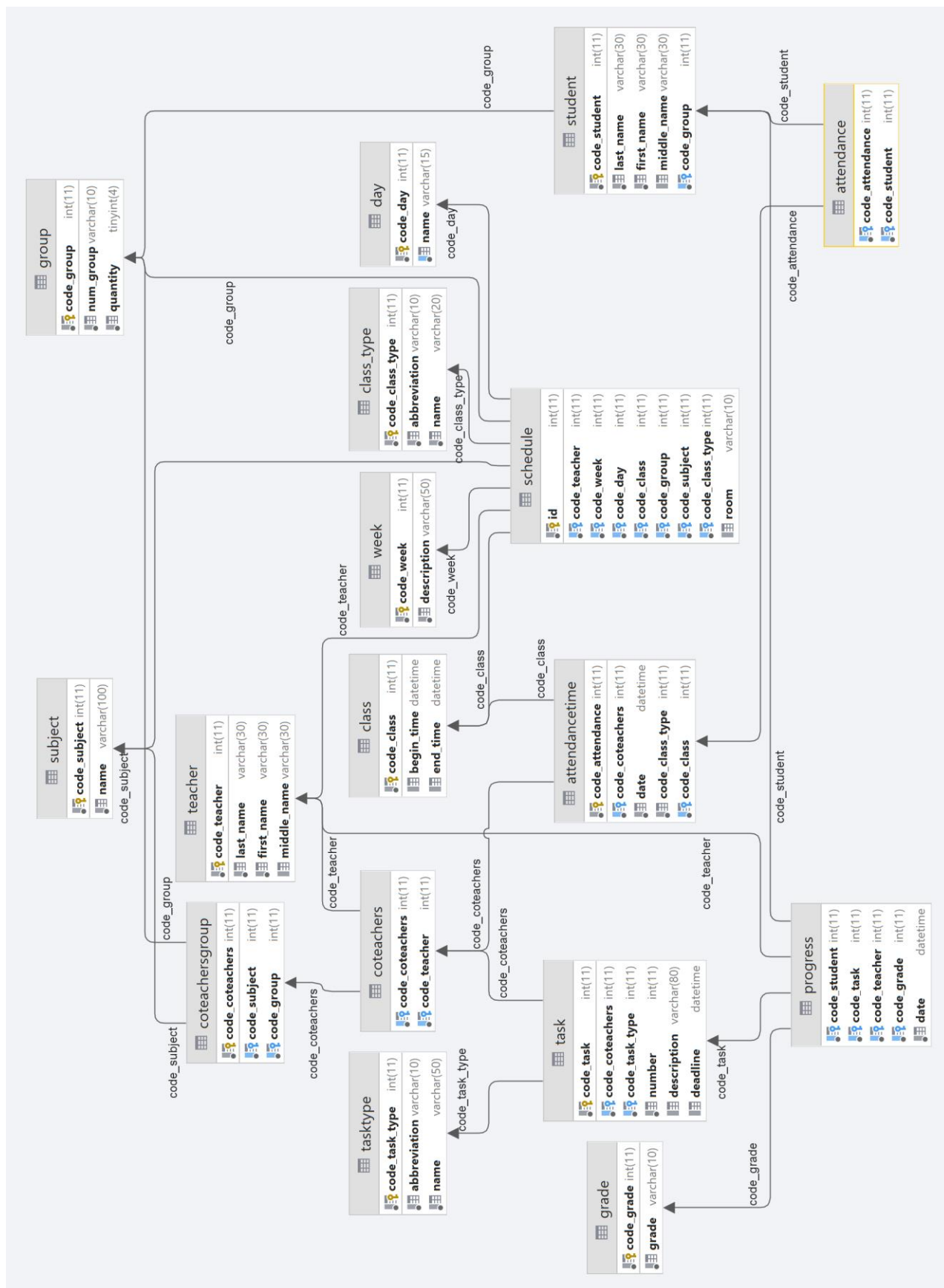


Рисунок 2.5 – Фізична модель БД

2.3.4 Оцінка розміру бази даних

Оцінка розміру бази даних дозволить прийняти рішення про вибір технічного забезпечення для сервера баз даних. Оцінка буде проводитися шляхом розрахунку.

Первісні дані

У таблиці 2.19 приведені первісні дані для розрахунку. Дані взяті з невеликим запасом для Палацу творчості учнів на півроку використання системи.

Таблиця 2.19 – Первісні дані для розрахунку розміру бази даних

Найменування	Значення
Кількість вчителів	50
Кількість видів тижнів (парний, непарний, перший тиждень місяця, другий тиждень місяця і т.д.)	7
Кількість місяців	4
Кількість тижнів у місяці	4
Кількість днів у тижні	7
Кількість навчальних днів у тижні	6
Кількість уроків у навчальному дні	8
Кількість класів учнів	30
Кількість дисциплін, які викладаються	100
Кількість занять в одному класі в день	6
Кількість учнів у класі	30
Кількість видів завдань	20
Кількість видів занять	10
Кількість оцінок (1..12, «здано» і т.д.), які використовуються	10
Кількість дисциплін в одному класі за півроку	15
Кількість вчителів у групі вчителів	2
Кількість завдань по предмету	10

Методика розрахунку

Розмір запису (кортежу) таблиці обчислюється додаванням розміру (довжини) усіх полів таблиці.

Розмір будь-якої таблиці бази даних обчислюється шляхом множення розміру запису (кортежу) таблиці на кількість записів.

Загальна кількість учнів обчислюється множенням кількості учнів у класі на кількість класів.

Кількість груп вчителів обчислюється множенням кількості дисциплін в одному класі на кількість класів.

Кількість вчителів, що входять у групи вчителів, обчислюється множенням кількості вчителів у групі вчителів на кількість груп вчителів.

Кількість завдань обчислюється перемножуванням кількості груп вчителів і кількості завдань по предмету.

Максимальна кількість занять обчислюється перемножуванням кількості класів учнів, кількості місяців, кількості тижнів у місяці, кількості навчальних днів у тижні і кількості уроків у дні.

Максимальна кількість записів у таблиці «Розклад» обчислюється перемножуванням кількості вчителів, кількості тижнів у місяці, кількості навчальних днів у тижні і кількості уроків у дні.

Максимальна кількість записів у таблиці «Відвідування» обчислюється перемножуванням максимального числа занять і кількості учнів у класі.

Кількість записів у таблиці «Успішність» обчислюється перемножуванням кількості завдань і кількості учнів у класі.

Розмір бази даних обчислюється додаванням розмірів усіх таблиць.

Розрахунок

Розрахунок розміру бази даних системи представлений у таблиці 2.20.

Таблиця 2.20 – Розрахунок розміру бази даних

Таблиця бази даних	Розмір запису (кортежу), байти	Кількість записів	Розмір таблиці, байти
«Вчитель» / «Teacher»	94	50	4700
«Тиждень» / «Week»	51	7	357
«День» / «Day»	16	7	112
«Урок» / «Class»	9	8	72
«Розклад» / «Schedule»	21	$50 \cdot 4 \cdot 6 \cdot 8 = 9600$	211200
«Клас» / «Group»	13	30	390
«Група вчителів» / «CoTeachersGroup»	6	$15 \cdot 30 = 450$	2700
«Вчитель у групі» / «CoTeachers»	6	$2 \cdot 450 = 900$	5400
«Дисципліна» / «Subject»	102	100	10200
«Заняття» / «AttendanceTime»	12	$30 \cdot 4 \cdot 4 \cdot 6 \cdot 8 =$ 23040	276480
«Учень» / «Student»	96	$30 \cdot 30 = 900$	86400
«Відвідування» / «Attendance»	8	$23040 \cdot 30 =$ 691200	5529600
«Завдання» / «Task»	90	$450 \cdot 10 = 4500$	405000
«Вид завдання» / «TaskType»	61	20	1220
«Оцінка» / «Grade»	11	10	110
«Успішність» / «Progress»	15	$4500 \cdot 30 = 13500$ 0	2025000
«Вид заняття» / «ClassType»	31	10	310
Розмір бази даних, байти			8569451

Аналіз результатів розрахунку

Таким чином, розмір бази даних для Палацу творчості учнів на півроку буде складати приблизно 9Мб; на 2 півріччя (навчальний рік) – 18Мб. Це порівняно невеликий розмір для бази даних, у якій міститься майже 1 мільйон записів.

Однак потрібно розуміти, що реальний розмір бази даних на жорсткому диску буде в кілька разів більший. Це справедливо, оскільки СУБД зберігає індекси, тимчасові таблиці, файли журналів і іншу службову інформацію.

2.4 Рішення з технічного забезпечення

Загальні положення

Програмне забезпечення інформаційної системи Палацу творчості учнів має розподілену архітектуру, тобто передбачає використання декількох серверів і клієнтських машин.

Для максимальної продуктивності і масштабуємості системи необхідно використовувати 3 сервери (хоча для коректної роботи системи досить і одного сервера, на який потрібно установити все програмне забезпечення):

- Web-сервер.
- Сервер доступу до даних або сервер EJB-об'єктів.
- Сервер баз даних (БД).

Засоби обчислювальної техніки

Вимоги до технічного забезпечення Web-сервера:

- Мікропроцесор класу Intel i5-2500 і вище.
- Оперативна пам'ять RAM 8 Гб і більше.
- Жорсткий диск з вільним місцем 20 Гб і більше (сервер баз даних повинен мати жорсткий диск з більшим обсягом вільного місця – 80 Гб і більше).
- Мережна карта Ethernet (не менше 10 Мбіт/с).

Вимоги до технічного забезпечення сервера доступу до даних:

- Мікропроцесор класу Intel i5-2500 і вище.
- Оперативна пам'ять RAM 8 Гб і більше.
- Жорсткий диск з вільним місцем 20 Гб і більше (сервер баз даних повинен мати жорсткий диск з більшим обсягом вільного місця – 80 Гб і більше).
- Мережна карта Ethernet (не менше 10 Мбіт/с).

Слід зазначити, що наведені конфігурації серверів (мікропроцесор, оперативна пам'ять, жорсткий диск) були узяті з рекомендацій компанії Oracle для серверів, що працюють під управлінням будь-якої операційної системи із встановленим Middleware компанії Oracle (GlassFish 4.1, Java JRE8 – серверна конфігурація). Крім того, сервер баз даних містить програмне забезпечення СУБД і базу даних системи, тому вимоги до розміру жорсткого диска збільшені.

Клієнтські машини або робочі станції можуть мати різні технічні характеристики, які впливають на комфортність роботи користувача. У якості робочих станцій рекомендується використовувати персональні комп'ютери з наступними характеристиками:

- Мікропроцесор класу Intel i3-550 і вище.
- Оперативна пам'ять RAM 4Гб і більше.
- Жорсткий диск з вільним місцем 20 Гб і більше.
- Мережна карта Ethernet (не менше 10 Мбіт/с).

Апаратура передачі даних

Палацу творчості учнів має наступні класи, які обладнані персональними комп'ютерами, що можуть використовуватися як робочі станції для розроблювальної системи:

- Комп'ютерний клас 1 (для проведення занять з програмування та інформаційних технологій).

- Комп'ютерний клас 2 (для занять з дизайну та технологій використання комп'ютера).
- Комп'ютерний клас 3 (для проведення занять за індивідуальними планами)
- Вчительська.
- Директорська.

Сервера і робочі станції системи повинні бути об'єднані в єдину комп'ютерну мережу. Швидкість передачі даних по мережі повинна бути не менше 10 Мбіт/с. На сегменті мережі із серверами для кращої продуктивності потрібно використовувати мережне обладнання для роботи на швидкості 100 Мбіт/с (мережні карти, концентратори (hub), комутатори (switch), маршрутизатори (router), кабелі). Як мережне середовище можна використовувати неекрановану виту пару 5 категорії (100Base-T) і 5е категорії (1000Base-TX).

На рисунку 2.6 представлена схема комп'ютерної мережі Палацу творчості учнів. Діаметр мережі не перевищує 120 метрів. Максимальна довжина сегмента не перевищує 50 метрів. Комутатори (switch) поділяють мережу на окремі домени колізій. З цього випливає, що мережа буде працювати коректно.

На даний момент Палацу творчості учнів має майже все необхідне обладнання для монтажу вказаної комп'ютерної мережі. Крім того, можлива інша конфігурація мережі з більшою пропускнуою здатністю.

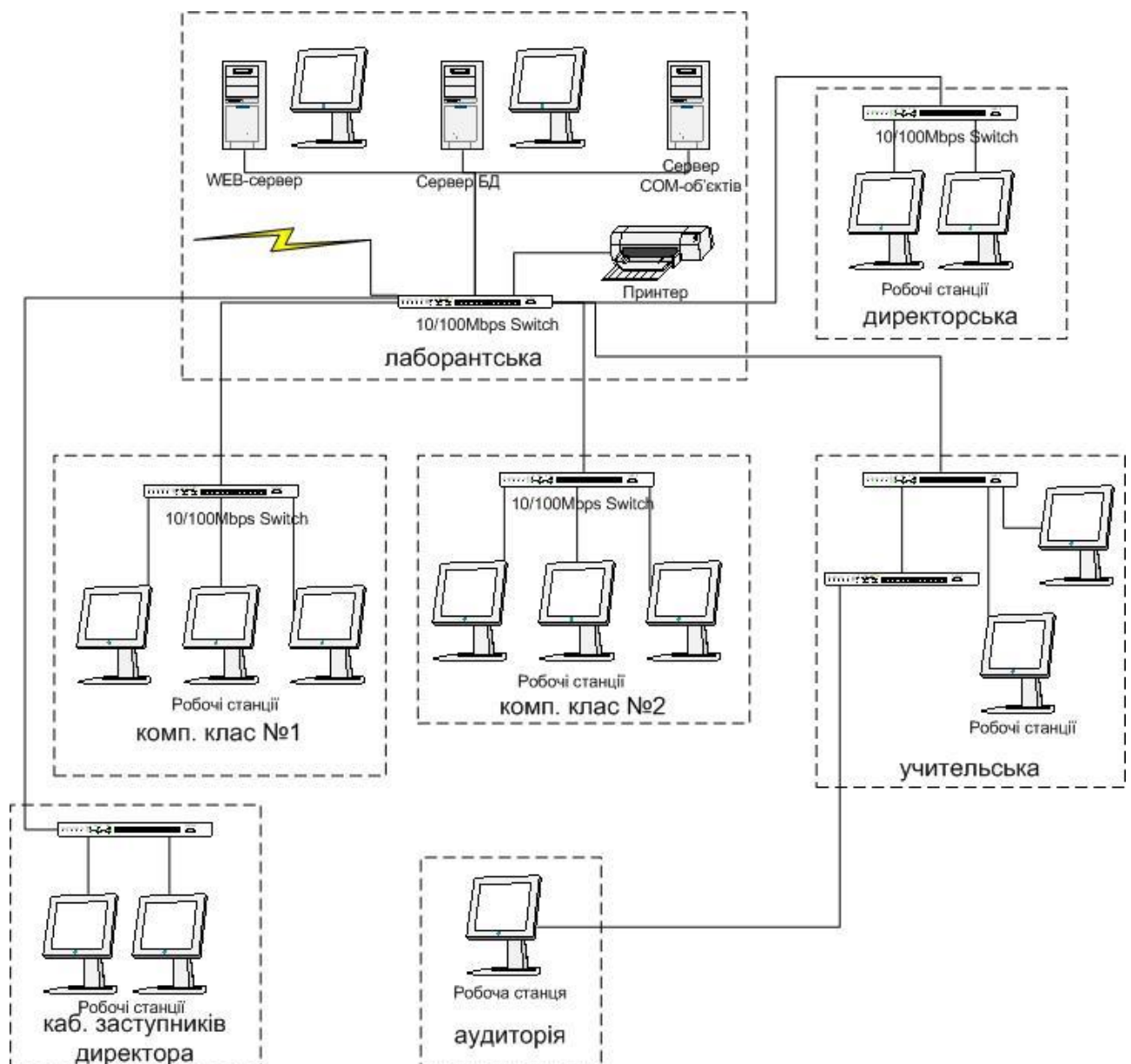


Рисунок 2.6 – Схема комп'ютерної мережі Палацу творчості учнів

Вартість обладнання, необхідного для монтажу вказаної мережі, представлена у таблиці 2.21. Щомісячна оплата інтернету складає 90 грн. при швидкості 30 Мбіт/с, 150 грн. при швидкості 80 Мбіт/с, 200 грн. при швидкості 100 Мбіт/с. Для роботи системи буде достатньо швидкості 4 Мбіт/с, але взагалі заклад повинен мати більшу швидкість доступу до Internet.

Таблиця 2.21 – Вартість мережного обладнання

Найменування	Вартість, грн.	Кіль- кість	Разом, грн.
Кабель кручена пара неекранована 5 категорії (5 UTP, 24 AWG, 4 пари, одножильна, 305 м.)	900	1	900
Кабель кручена пара неекранована 5е категорії (5 UTP, 24 AWG, 4 пари, одножильна, 305 м.)	1000	1	1000
Комутатор Ethernet/ Fast Ethernet (8 портів, 10/100 Mbps)	300	3	900
Комутатор Ethernet /Fast Ethernet (16 портів, 10/100 Mbps)	600	4	2400
Разом, грн.			5200

Палац творчості учнів має майже все вказане мережне обладнання (комутатори, концентратори), але концентратори Ethernet мають тільки 8 портів. Відповідно, у комп'ютерних класах, де кількість робочих станцій становить 15 комп'ютерів, використовуються два таких концентратора. Проте, кращим рішенням буде використання концентраторів з 16 портами.

2.5 Рішення з програмного забезпечення

Інформаційна система Палацу творчості учнів призначена для автоматизації функцій, що виконують педагоги та керівництво закладу освіти.

Користувачами системи є керівники (директор та його заступники, голова наглядової ради) та педагоги Палацу творчості учнів. Також необхідний адміністратор системи, який буде підтримувати роботу системи.

Інформаційне забезпечення системи складається з:

- Бази даних, у якій зберігається вся інформація про успішність, відвідування, завдання, розклад та інша необхідна інформація.
- Файлів електронної записної книжки, у яких зберігаються записи, зроблені користувачем на конкретні дні, а також інформація про вільні (немає запису) та зайняті дні (є запис).
- Файлів аутентифікації, у яких зберігаються логіни та паролі (у зашифрованому виді) користувачів системи. З цими файлами безпосередньо працює тільки адміністратор системи.

Система не вносить змін в організаційну структуру Палацу творчості учнів. Адміністратором системи може бути адміністратор існуючої локальної мережі школи.

Технічне забезпечення системи складається з одного чи декількох серверів і робочих станцій, встановлених у навчальних класах. Для максимальної продуктивності та масштабуємості системи необхідно використовувати 3 сервери (хоча для коректної роботи системи досить і одного сервера, на який потрібно установити все програмне забезпечення):

- Web-сервер.
- Сервер доступу до даних або сервер EJB-об'єктів.
- Сервер баз даних (БД).

Програмне забезпечення системи складається з декількох пакетів програм:

- Пакет програм для забезпечення функціональності, яка необхідна педагогам та керівникам навчального закладу (описана вище).
- Пакет програм для адміністратора системи.

Кожен пакет програм складається з певної кількості програм (модулів), що реалізують окремі функції системи. У даній роботі буде описана розробка пакета програм для педагогів.

Бізнес-процеси реалізуються сервлетами та JSP-сторінками, які працюють на Web-сервері.

Доступ до бази даних здійснюється за допомогою EJB-об'єктів, наданих EJB-контейнером серверу Payara, і технології JDBC. EJB-контейнер сервера

Payara та встановлені на ньому EJB-об'єкти виконують роботу сервера доступу до даних.

База даних є реляційною і може працювати під управлінням реляційної СУБД, що підтримує стандарт SQL-92 та має JDBC і/або ODBC драйвера.

Операційна система, встановлена на Web-сервері і сервері доступу до даних, повинна мати реалізацію Java та Enterprise Java Beans версії не нижче 3 (операційна система, що рекомендується – Ubuntu 20.04 LTS). Сервер баз даних не має ніяких обмежень, крім тих, котрі накладає СУБД, яка використовується.

На Web-сервері системи повинен бути встановлений HTTP-сервер з підтримкою HTTP 1.1, сервлетів, JSP і можливістю аутентифікації користувача.

2.5.1 Структура програмного забезпечення

Виходячи з аналізу функціональної моделі інформаційної системи (див. пункт 2.1.2), можна розробити структуру програмного забезпечення. У даній роботі буде описана розробка програмного забезпечення для педагогів. Оскільки програмне забезпечення для педагога є розподіленим (працює на декількох комп'ютерах), то воно повинно мати модульну структуру. Схема взаємодії модулів програмного забезпечення представлена на рисунку 2.7.

Перший модуль, що отримує управління, є модуль аутентифікації. Користувач вводить свій логін і пароль. Аутентифікацію потрібно здійснювати засобами Web-сервера, тому цей модуль розробляти не потрібно. Модуль аутентифікації звертається до файлів аутентифікації, у яких зберігаються логіни та паролі всіх користувачів.

Наступний модуль, якому передається управління (у випадку правильної аутентифікації), – модуль головного вікна. Він виводить на екран головне вікно програми, з яким буде взаємодіяти користувач. Модуль головного вікна викликає модуль меню, що генерує меню в головному вікні, використовуючи для роботи з базою даних модуль доступу до даних для меню. Аналогічно, модуль головного

вікна викликає модуль виводу тижня і уроку, що взаємодіє з модулем сервера часу.

Головне вікно програми залишається активним завжди (тільки звіти завантажуються в інші вікна). Воно містить меню, за допомогою якого можна викликати інші модулі програмного забезпечення. При цьому вся інформація й інтерфейс користувача, що надають різні модулі, будуть з'являтися в основній частині головного вікна. Таким чином, користувач працює тільки з одним вікном, у якому є меню й інформація. Інформація змінюється в залежності від того, який пункт меню вибирається.

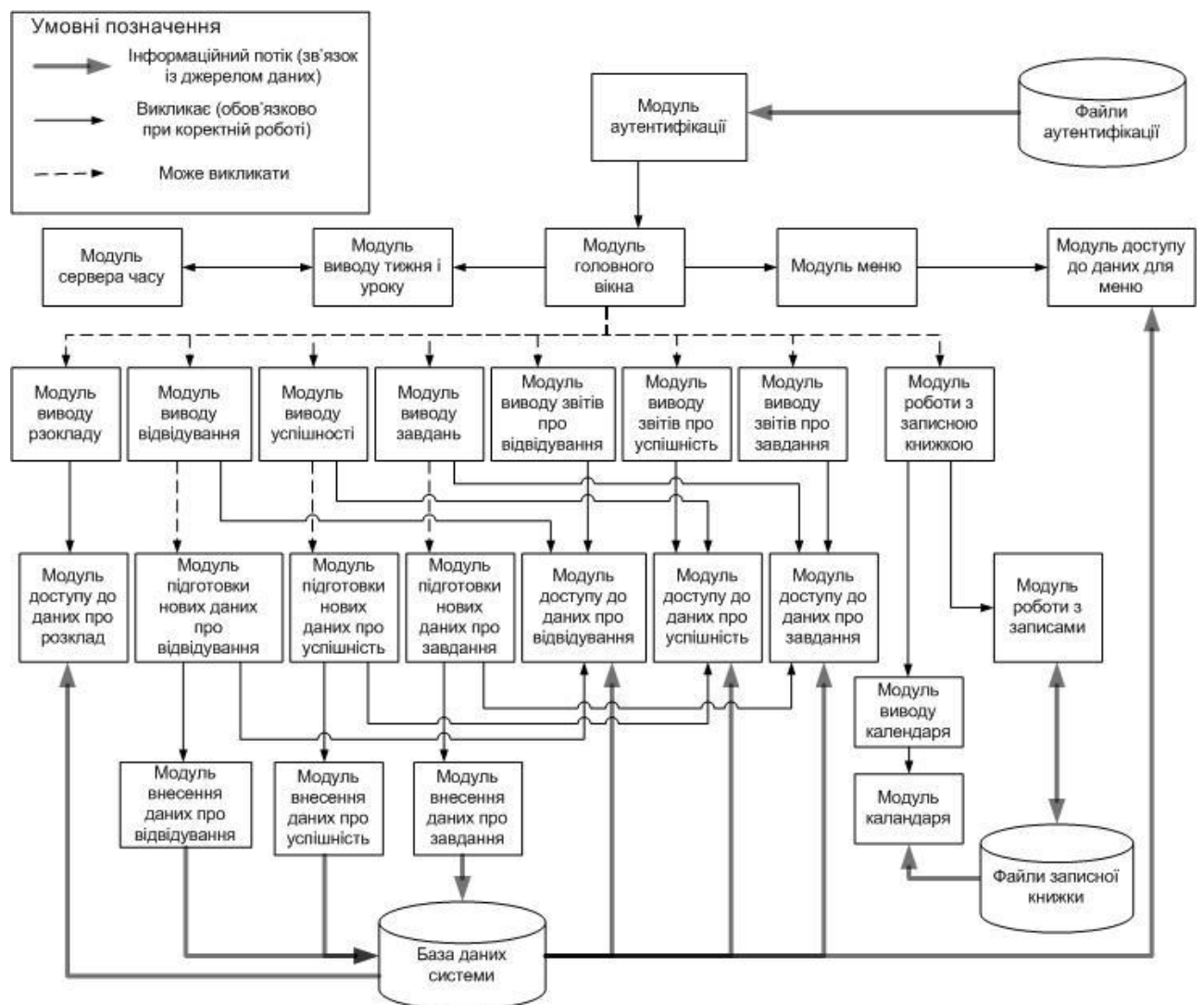


Рисунок 2.7 – Схема взаємодії модулів програмного забезпечення системи

З модуля головного вікна можна викликати (тимчасово передати управління) наступні модулі:

- Модуль виводу розкладу.
- Модуль виводу відвідування.
- Модуль виводу успішності.
- Модуль виводу завдань.
- Модуль виводу звітів про відвідування.
- Модуль виводу звітів про успішність.
- Модуль виводу звітів про завдання.
- Модуль роботи з записною книжкою.

Кожний з цих модулів, крім модуля роботи з записною книжкою, викликає чи може викликати інші модулі.

Модуль виводу розкладу викликає модуль доступу до даних про розклад, що взаємодіє з базою даних.

Модуль виводу відвідування викликає модуль доступу до даних про відвідування, що взаємодіє з базою даних. Модуль виводу відвідування може викликати (при натисканні відповідного елемента управління) модуль підготовки нових даних про відвідування, що взаємодіє з базою даних за допомогою модуля внесення даних про відвідування і модуля доступу до даних про відвідування.

Модуль виводу успішності викликає модуль доступу до даних про успішність, що взаємодіє з базою даних. Модуль виводу успішності може викликати (при натисканні відповідного елемента управління) модуль підготовки нових даних про успішність, що взаємодіє з базою даних за допомогою модуля внесення даних про успішність і модуля доступу до даних про успішність.

Модуль виводу завдань викликає модуль доступу до даних про завдання, що взаємодіє з базою даних. Модуль виводу завдань може викликати (при натисканні відповідного елемента управління) модуль підготовки нових даних

про завдання, що взаємодіє з базою даних за допомогою модуля внесення даних про завдання і модуля доступу до даних про завдання.

Модуль виводу звітів про відвідування викликає модуль доступу до даних про відвідування, що взаємодіє з базою даних.

Модуль виводу звітів про успішність викликає модуль доступу до даних про успішність, що взаємодіє з базою даних.

Модуль виводу звітів про завдання викликає модуль доступу до даних про завдання, що взаємодіє з базою даних.

Модуль роботи з записною книжкою викликає модуль виводу календаря і модуль роботи з записами, що працюють з файлами записної книжки. Модуль виводу календаря викликає модуль календаря.

2.5.2 Операційна система

При розробці концепції інформаційної системи загальноосвітньої школи для Палацу творчості учнів, у якості серверної операційної системи була вибрана Ubuntu 20.04 LTS. Все програмне забезпечення, описане вище, буде працювати під управлінням цієї операційної системи.

Ubuntu 20.04 LTS – це багатозадачна мережна операційна система масштабу підприємства. Вона має високі показники масштабуємості, продуктивності, безпеки, стабільності, широкий вибір програм. Її основні характеристики представлені в таблиці 2.23. Інформація, представлена в таблиці 2.23, отримана на офіційному сайті співтовариства Ubuntu, який підтримується компанією Canonical (www.ubuntu.com).

Таблиця 2.23 – Характеристики Ubuntu 20.04 LTS

Характеристика	Значення
Процесор	Pentium 2000 МГц і вище
Максимальна ємкість пам'яті, Гбайт	64
Мінімальна ємкість пам'яті, Мбайт	2048
Ємкість пам'яті, що рекомендується, Мбайт	4096
Жорсткий диск	500 Гбайт і більше (не менш 40 Гбайт вільного місця)
Сертифікація програм	Сертифікат Ubuntu compatible
Продавець	Ubuntu Community, Canonical corp.
Апаратна сумісність	Стандартна x86 або x64 архітектура

РОЗДІЛ 3. РОЗРОБКА РОБОЧОЇ ДОКУМЕНТАЦІЇ

У технічному завданні (постановці задачі) на розробку системи визначені вимоги до документації (дивиться п. 1.4.9). Відповідно до цих вимог були розроблені наступні документи:

- Загальний опис системи (Додаток А).
- Керівництво користувача (Додаток Б).
- Інструкція з формування і ведення бази даних (Додаток В).
- Програма та методика випробувань (Додаток Г)
- Текст програми (Додаток Д).

РОЗДІЛ 4. РОЗРАХУНОК ЕКОНОМІЧНОГО ЕФЕКТУ ВІД ВПРОВАДЖЕННЯ СИСТЕМИ ПАЛАЦУ ТВОРЧОСТІ УЧНІВ

4.1 Загальні положення

На даний момент сучасні обчислювальні засоби широко використовуються в повсякденній діяльності людини. Сьогодні обчислювальна техніка дозволяє вирішувати найрізноманітніші задачі: від звільнення людини від монотонної ручної праці до розробки складних технічних розрахунків, 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 підсистема «Робоче місце вчителя» призначена для автоматизації задач, що виконують вчителі загальноосвітньої школи. Впровадження системи приведе до значної економії, що виражається в наступних показниках:

- підвищення продуктивності праці;
- вивільнення робочого часу.

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 і впровадження системи

Витрати на розробку 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.

Виходячи з того, що основна заробітна плата розробників програмного забезпечення складає 600 грн./міс, вартість сучасної ПЕОМ складає 3000 грн. і вартість кіловат-години електроенергії складає 0,2436 грн, розрахуємо вартість розробки системи.

1) Розрахунок заробітної плати зробимо по формулі 4.1:

$$З_{зп} = ЗП_{раз} \cdot T_{раз}, \quad (4.1)$$

де $ЗП_{раз}$ – зарплата розробника, за місяць;

$T_{раз}$ – тривалість розробки (дослідження, створення, налагодження і впровадження).

Для розробки даної системи необхідно 6 місяців (за експертною оцінкою часу на розробку аналогічних систем). З цього випливає, що загальна сума витрат на заробітну плату складе:

$$З_{зп} = ЗП_{раз} \cdot T_{раз} = 600 \cdot 6 = 3600 \text{ (грн)};$$

2) витрати на амортизацію ЕОМ, на якій виконується розробка, розраховуються по формулі 4.2:

$$З_{\text{амор}} = Ц_{\text{ЭВМ}} \cdot A \cdot T_{\text{разр}} \quad (4.2)$$

де $Ц_{\text{ЭВМ}} = 3000$ грн. – балансова вартість ЕОМ;

$A = 60\%$ - амортизація за рік

$T_{\text{сл}} = 5$ років – термін служби ЕОМ;

$T_{\text{разр}} = 0,5$ року – час, необхідний для розробки системи.

$$З_{\text{амор}} = 3000 \cdot 0.6 \cdot 0.5 = 900 \text{ (грн);}$$

3) витрати на експлуатацію ЕОМ, на якій виконується розробка, полягають в оплаті споживаної нею електричної енергії і розраховуються по формулі 4.3:

$$З_{\text{експл}} = P_{\text{ЭВМ}} \cdot T_{\text{разр}} \cdot N_{\text{рд}} \cdot N_{\text{рч}} \cdot Э_{\text{ст}}, \quad (4.3)$$

де $P_{\text{ЭВМ}} = 0,4$ кВт/год – потужність ЕОМ;

$T_{\text{разр}} = 6$ міс. – тривалість розробки;

$N_{\text{рд}} = 21$ день – число робочих днів у місяці;

$N_{\text{рч}} = 8$ годин – число годин у робочому дні;

$Э_{\text{ст}} = 1,68$ грн – вартість 1 кВт електроенергії.

$$З_{\text{експл}} = 0,4 \cdot 6 \cdot 21 \cdot 8 \cdot 1,68 = 677,38 \text{ (грн);}$$

4) витрати на матеріали і комплектуючі вироби враховують витрати на магнітні носії даних, папір для друкувальних пристроїв, літературу, необхідну для розробки данної системи, картридж і тонер до принтера, а також непередбачені витрати. Розрахунок приведений у таблиці 4.1 (Ціни взяті відповідно до прайслистів та договірних цін на даний вид матеріалів).

Таблиця 4.1 – Розрахунок витрат на матеріали і комплектуючі вироби

№	Найменування виробів	Вартість, грн.
1	Папір	120.00
2	Картридж і тонер до принтера	350.00
3	Флеш-накопичувачі	120.00
4	Література	900.00
5	Непередбачені витрати	100.00
Разом		1590.00

5) для розробки системи необхідні наступні засоби розробки (таблиця 4.2)
Ціни взяті відповідно до прайслистів та договірних цін на даний вид матеріалів.

Таблиця 4.2 – Програмні засоби розробки

№	Найменування	Вартість, грн.
1	IntelliJ IDEA Ultimate	4 023.00
Разом		4 023.00

б) загальний кошторис витрат на створення системи, з урахуванням п.1) – п.5) приведений в таблиці 4.3.

Таблиця 4.3 – Загальний кошторис витрат на створення системи

№	Найменування витрат	Сума, грн.
1	Заробітна плата основна	3 600.00
2	Заробітна плата додаткова (20% від п.1)	720.00
3	Витрати на амортизацію ЕОМ	900.00
4	Витрати на експлуатацію ЕОМ	98.00
5	Витрати на матеріали і комплектуючі вироби	445.00
6	Витрати на програмні засоби розробки	4 023.00
Разом на створення системи		6 967.00

До витрат на впровадження системи можна віднести витрати на придбання технічного забезпечення, вартість програмного забезпечення, вартість навчання кадрів, витрати на монтаж і налаштування мережі.

Оскільки склад і параметри технічних засобів, які вже є, відповідають вимогам, то їх вартість при розрахунку витрат враховувати не будемо.

Виходячи з вимог до програмного забезпечення, а також проаналізувавши цінову політику можемо прийняти наступне (див. таблицю 4.4).

Таблиця 4.4 – Перелік програмного забезпечення, необхідного для впровадження системи

№	Найменування ПО	Сума, грн.	Кількість	Разом, грн.
1	Ubuntu Linux 20.04	0.00	3	0.00
2	СУБД MariaDB	0.00	1	0.00
3	Payara 5 Community Server	0.00	1	0.00
Разом				0.00

Витрати на програмне забезпечення склали **0.00** грн.

Інвестиційні витрати на впровадження системи з урахуванням вартості навчання кадрів і витрат на монтаж і налаштування мережі приведені в таблиці 4.5.

Таблиця 4.5 – Інвестиційні витрати на впровадження системи

№	Найменування інвестицій	Сума, грн.
1	Вартість устаткування	—
2	Вартість програмного забезпечення	0.00
3	Вартість підготовки кадрів	5000.00
4	Монтаж і налаштування локальної мережі	5000.00
5	Супроводження системи	30000.00
Разом на впровадження системи		40000.00

Разом загальна сума витрат на створення і впровадження системи складе **135 570.20** грн.

4.3 Економічна ефективність розробки і впровадження системи палацу творчості учнів

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.

За даними, які були отримані шляхом експериментальної оцінки і вчителів загальноосвітньої школи, було встановлено, що використання даної системи дозволить вивільнити 0,1 робочого часу вчителя. Оскільки на даний момент у школі працює 85 педагогів, то загалом система умовно вивільнить 8,5 робітника.

Визначимо пряму економічну ефективність ґрунтуючись на тому, що впровадження системи вивільнить 8,5 працівника .

Зарплата 8,5 працівників у рік складає $\Delta C = (5600 \cdot 12) \cdot 8,5 = 571\,200$ (грн.)

Річний економічний ефект розраховується по формулі 4.4:

$$\mathcal{E}_{год} = \Delta C_n - \Pi_{comp} - E_n \cdot k, \quad (4.4)$$

де E_n – нормативний коефіцієнт економічної ефективності капітальних вкладень у галузь – для обчислювальної техніки приймається – 0,6;

Π_{comp} – вартість супроводження системи;

k – додаткові капітальні вкладення з урахуванням витрат на проектування, створення і функціонування системи

$$\mathcal{E}_{год} = 571200 - 40000 - 0,6 \cdot 135570,20 = 560065,68 \text{ (грн.)}$$

Строк окупності системи розраховується по формулі 4.5:

$$T = \frac{k}{\Delta C} = \frac{135570,20}{560065 - 30000} \approx 0,24 \text{ (року)} \quad (4.5)$$

Отже строк окупності системи складає майже 3 місяці.

РОЗДІЛ 5. ОХОРОНА ПРАЦІ

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.

5.1 Аналіз небезпечних і шкідливих факторів

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 запиленість 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), мутагенні (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 гранично-допустимими 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, потребуючою 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 персоналі 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.

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 II-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 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 °C
	Відносна вологість	40...60 %
	Швидкість руху повітря	до 0,1 м/с
Теплий	Температура повітря в приміщенні	23...25 °C
	Відносна вологість	40...60 %
	Швидкість руху повітря	0,1...0,2 м/с

Шумом називають усякий несприятливо діючий на людину звук. З фізичної точки зору звук являє собою механічні коливання пружного середовища [2].

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/м².

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:	20 кВ/м
для дорослих користувачів	
для дітей дошкільних установ і учнів	15 кВ/м
середніх спеціальних і вищих навчальних закладів	

5.2 Розрахунок системи штучного освітлення приміщення

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 накаливання);
- більш тривалий термін служби.

Вхідні дані:

Розрахунок освітлення проводиться для кімнати площею 20 м², ширина якої 4 м, висота – 3 м. Висота робочої поверхні 1 м. Для освітлення приймається світильник типу ППД2-500. Мінімальна освітленість лампи накаливання по нормах $E_{\min} = 300$ лк. Коефіцієнт відображення стелі $S_{\pi}=70\%$, стін $S_c=50\%$ і робочої поверхні $S_p=30\%$. Напруга мережі 220В.

Скористаємося методом світлового потоку.

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

$$F = \frac{E_{\min} \cdot K_z \cdot S \cdot z}{N \cdot \eta}$$

де

F – світловий потік, що розраховується, лм;

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

S – площа освітлюваного приміщення (у нашому випадку $S = 20$ м²);

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

K_z – 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:

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

η – 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 η визначимо по таблиці коефіцієнтів

використання різних світильників. Для цього обчислимо індекс приміщення по формулі:

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

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 , по таблиці 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 $P_{\text{обиц}} = P_{\text{л}} \cdot N = 200 \cdot 4 = 800 \text{ (Вт)}$.

5.3 Розробка заходів щодо зменшення впливу шкідливих факторів

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.

Раціональне 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 ізоляцій 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 [20].

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, регулюємість 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 відеотермінала, 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 відеотермінала наступні:

- голова не повинна бути нахилена більш ніж на 20°,
- плечі повинні бути розслаблені,
- лікті – під кутом 80°...100°,

– 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 [1].

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 [1].

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

Категорія роботи з ВДТ чи ПЕОМ	Рівень навантаження за робочу зміну при видах			Сумарний час регламентованих перерв, хв	
	Група А, кількість екранів	Група Б, кількість екранів	Група В, годин	При 8-годинній зміні	При 12-годинній зміні
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 ПЕОМ із попереднім запитом;

група Б: робота з введення інформації;

група В: творча робота в режимі діалогу з ЕОМ

РОЗДІЛ 6. ОХОРОНА НАВКОЛИШНЬОГО СЕРЕДОВИЩА

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.

6.1 Забруднення навколишнього середовища

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, 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 сучасних умовах є невід'ємною частиною соціального розвитку суспільства.

6.2 Автотранспорт як основне джерело забруднення атмосферного повітря

До мобільних джерел забруднення відносяться автомобілі і транспортні механізми, що пересуваються по землі, по воді і по повітря. 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 ³	До 10 мкг/м ³

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	
	За	На
Оксид вуглецю	752	23,7 г/км
Незгорілі вуглеводні	29,4	0,93 г/км
Оксиди азоту	33,2	1,05 г/км
Свинець	1,11	0,035 г/км
Сумарна кількість вихлопних газів	28,95	0,914 м ³ /км
Середня витрата палива	2,75	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 розмаїтності 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 відходних 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 [9].

І, наприклад, установка вологих фільтрів при газоочищуванні дозволяє скоротити забруднення повітря, але веде до ще більшого забруднення води. Уловлені з відходних 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 [9].

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 теплоелектроцентрально 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 аерозолей, 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 тумана і бризів. У другому випадку – нейтралізація газо- і пароподібних домішок.

Очищення від аерозолей 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 [11].

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.

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, наносимої Природі Людиною.

Висновки

У даній роботі була виконана розробка інформаційної системи Палацу творчості учнів. Для цього були вирішені наступні задачі:

- Проведений аналіз сучасних інформаційних технологій, що використовуються для розробки інформаційних систем.
- Сформовані вимоги до системи.
- Розроблена концепція системи.
- Проведене техніко-економічне обґрунтування розробки системи.
- Розроблені загальносистемні рішення, рішення з інформаційного, організаційного, технічного та програмного забезпечення.
- Розроблене необхідне програмне забезпечення системи.
- Розроблена відповідна документацію.

Для аналізу сучасних інформаційних технологій, що використовуються для розробки інформаційних систем, був проведений огляд літературних і Internet-джерел. У результаті дослідження були описані необхідні технології для розробки інформаційної системи.

При формуванні вимог до системи була вивчена діюча неавтоматизована інформаційна система і її недоліки, обґрунтована необхідність автоматизації діючої системи, визначені цілі, критерії, обмеження, функції та задачі розробленої інформаційної підсистеми.

При розробці концепції системи були обрані кілька варіантів, кожний з яких задовольняє вимогам до системи. Для вибору оптимального варіанта концепції системи було проведене техніко-економічне обґрунтування розробки системи засобами функціонально-вартісного аналізу. Обраний оптимальний варіант передбачає використання Web-сервера (операційна система – Ubuntu 20.04; HTTP-сервер – Apache Server 2.1), сервера доступу до даних (операційна система – Ubuntu 20.04) і сервера баз даних (операційна система – Ubuntu 20.04; система управління базами даних – MySQL 5.5). У якості засобів розробки обрані Java SE + Java EE та середовище розробки IntelliJ IDEA.

У процесі розробки рішень з інформаційного забезпечення були розроблені концептуальна, логічна і фізична модель даних для бази даних системи, структури файлів аутентифікації і записної книжки педагога.

У якості рішень з технічного забезпечення була спроектована локальна комп'ютерна мережа Палацу творчості учнів, яка необхідна для функціонування інформаційної системи.

Відповідно до обраного оптимального варіанта концепції інформаційної системи було реалізоване необхідне програмне забезпечення системи, що має модульну структуру і складається з 27 модулів.

Розроблена документація до системи включає наступні документи: загальний опис системи, керівництво користувача, інструкція з формування і ведення бази даних, керівництво адміністратора з установки програмного забезпечення на серверах, програма і методика випробувань.

Розроблена інформаційна система позашкільного закладу освіти цілком відповідає технічному завданню. Її перша версія буде впроваджена у Палаці творчості учнів міста Миколаєва.

СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ

- 1 Mark J. Collins. Pro HTML5 with CSS, JavaScript, and Multimedia. Complete Website Development and Best Practices. – Apress, 2017. – 560p.
- 2 Kishori Sharan. Beginning Java 9 Fundamentals. Arrays, Objects, Modules, JShell, and Regular Expressions. – Apress, 2017. – 1037p.
- 3 Derek DeJonghe. NGINX Cookbook. - O'Reilly Media, 2021. – 189p.
- 4 Dave WolfA, J. Henley. Java EE Web Application Primer. – Apress, 2017. – 145p.
- 5 Ryan Stephens, Ron Plew, Arie D. Jones. Sams Teach Yourself SQL in 24 hours. – Pearson Education, Inc. 2011. – 680p.
- 6 Ted Hagos. Beginning IntelliJ IDEA: Integrated Development Environment for Java Programming. – Apress 2021. – 274p.
- 7 Чен Р.М. – Методичні рекомендації до виконання лабораторної роботи з теми "Використання CASE-технологій у проектуванні інформаційних систем та БД". — Харків: Вид. ХНЕУ, 2008. — 52 с.
- 8 Справочник по Автоматизации. – М.: Издательский отдел «Русская Редакция» ТОО «Channel Trading Ltd.», 1998. – 440 с.
- 9 Елманов В.И., Терновая Г.Г. Охрана атмосферного воздуха. – Г.: Юридическая литература, 1984. – 220с.
- 10 Клименко Л.П. Техноэкология: Учебник для студентов высших учебных заведений. – Ник.: ВМФ Наукма, 2000. – 304с.
- 11 Коржинський С.Н. – Настольная книга Web-мастера: Эффективное применение HTML,CSS, JAVASCRIPT. – СПб: Издательство «КноРус», 2000. – 320 с.
- 12 Кристиансен Т. Торкингтон Н. – Perl. Библиотека программиста – СПб: Питер, 2001. – 736 с.
- 13 Морган М. – Java 2. Руководство разработчика. – М.: Издательский дом «Вильямс», 2000. – 720 с.
- 14 Новиков Ю.В. Экология, окружающая среда и человек: Учебное пособие для

- вузов. – М.: Агентство “ФАИР”, 1998. – 295с.
- 15 Охрана окружающей среды. Под редакцией Белова С.В. – М.: Высшая школа, 1983. – 195с.
 - 16 Принципы проектирования и разработки программного обеспечения. – М.: Издательско-торговый дом «Русская Редакция», 2000. – 608 с.
 - 17 Разработка приложений на Microsoft Visual C++ 6.0. – М.: Издательско-торговый дом «Русская Редакция», 2000. – 576 с.
 - 18 Самгин Э.Б. Освещение рабочих мест. – М.: МИРЭА, 1989. – 186с.
 - 19 Страуструп Б. – Язык программирования C++, 3-е изд. – СПб.; М.: «Невский Диалект» – «Издательство БИНОМ», 1999. – 991 с
 - 20 Тихомиров Ю.В. – MS SQL Server 2000: разработка приложений. – СПб.: БХВ-Петербург, 2000. – 368 с.
 - 21 Томас М. – Секреты программирования для Internet на Java. – СПб.: Питер, 1997. – 640 с.
 - 22 Трельсен Э. – Модель COM и применение ATL 3.0. – СПб.: BHV – Санкт-Петербург, 2000. – 928 с.
 - 23 Фронковик Д., Гарсиа М., Уолен Э. – Руководство администратора Microsoft SQL Server 7.0. – М.: Издательско-торговый дом «Русская Редакция», 2000. – 672 с.

ДОДАТКИ

Додаток А

Загальний опис системи

Призначення системи

Інформаційна система документообігу позашкільного закладу освіти призначена для автоматизації задач, що виконують співробітники закладу позашкільної освіти. Система розробляється для Палацу творчості учнів міста Миколаєва.

Система реалізує наступні функції:

- Аутентифікація користувачів.
- Облік успішності учнів.
- Облік відвідування занять учнями.
- Видача завдань учням педагогами.
- Надання педагогу інформації про розклад його занять (організація робочого часу).
- Створення звітів про успішність, відвідування та видані завдання, які придатні для друку.
- Ведення електронної записної книжки (щоденника) педагога.
- Функції, які виконує адміністратор системи: ведення бази даних і настройка аутентифікації системи.

Опис системи

Користувачами системи є керівники та педагоги Палацу творчості учнів. Також необхідний адміністратор системи, який повинен підтримувати роботу системи.

Інформаційне забезпечення системи складається з:

- Бази даних, у якій зберігається вся інформація про успішність, відвідування, завдання, розклад та інша необхідна інформація.
- Файлів електронної записної книжки, у яких зберігаються записи, зроблені вчителем на конкретні дні, а також інформація про вільні (немає записи) та зайняті дні (є запис).
- Файлів аутентифікації, у яких зберігаються логіни та паролі (у зашифрованому виді) користувачів системи. З цими файлами безпосередньо працює тільки адміністратор системи.

Система не вносить змін в організаційну структуру Палацу творчості учнів. Адміністратором системи може бути один з існуючих адміністраторів серверів кафедри.

Технічне забезпечення системи складається з одного чи декількох серверів і робочих станцій, встановлених у навчальних аудиторіях. Для максимальної продуктивності та масштабуємості системи необхідно використовувати 3 сервери (хоча для коректної роботи системи досить і одного сервера, на який потрібно установити все програмне забезпечення):

- Web-сервер.
- Сервер доступу до даних чи сервер EJB-об'єктів.
- Сервер баз даних (БД).

Програмне забезпечення системи складається з декількох пакетів програм:

- Пакет програм для забезпечення функціональності, необхідної вчителям (описана вище).
- Пакет програм для адміністратора системи.

Кожен пакет програм складається з певної кількості програм (модулів), що реалізують окремі функції системи.

Бізнес-процеси реалізуються сервлетами та JSP-сторінками, що встановлюються на Web-сервері.

Доступ до бази даних здійснюється за допомогою EJB-об'єктів, наданих EJB-контейнером, і технології JDBC. EJB-контейнер та EJB-об'єкти, що містяться в ньому встановлюються на сервер доступу до даних.

База даних є реляційною і може працювати під управлінням реляційної СУБД, що підтримує стандарт SQL-92 та має JDBC драйвера.

Операційна система, встановлена на Web-сервері і сервері доступу до даних, повинна мати реалізацію JDK (операційна система, що рекомендується – Ubuntu 20.04 LTS). Сервер баз даних не має ніяких обмежень, крім тих, котрі накладає СУБД, яка використовується.

На Web-сервері системи повинен бути встановлений HTTP-сервер з підтримкою HTTP 1.1, JSP і можливістю аутентифікації користувача.

Опис взаємозв'язків системи з іншими системами

База даних системи Палацу творчості учнів може використовуватися іншими інформаційними системами (наприклад, система «Загальний освітній простір міста» може отримувати інформацію про відвідування занять учнями певного класу і так далі).

Додаток Б

Керівництво користувача

Вступ

Інформаційна система Палацу творчості учнів розроблена для Палацу творчості учнів міста Миколаєва.

Система дозволяє автоматизувати функції, що виконують педагоги палацу. Система має наступні функції:

- Облік успішності учнів.
- Облік відвідування занять учнями.
- Видача завдань учням педагогами.
- Надання вчителю інформації про розклад його занять (організація робочого часу).
- Створення звітів про успішність, відвідування і видані завдання, придатних для друку.
- Ведення електронної записної книжки (щоденника) педагога.

Користувачами системи є вчителі Палацу творчості учнів. Їхній рівень підготовки повинен відповідати рівню користувача ПЕОМ. Вчителям потрібно ознайомитися з даним керівництвом для освоєння системи.

Призначення та умови застосування

Інформаційна система Палацу творчості учнів призначена для автоматизації задач, що виконують педагоги. Система розроблена для Палацу творчості учнів.

Система реалізує наступні функції, що може використовувати користувач:

- Аутентифікація користувача. (Педагог вводить свій логін і пароль.)
- Облік успішності учнів. (Педагог ставить оцінки учням, що виконали (чи не виконали) завдання.)

- Облік відвідування занять учнями. (Педагог відмічає відсутніх на занятті учнів.)
- Видача завдань учням вчителями. (Педагог видає завдання учням.)
- Надання педагогу інформації про розклад його занять (організація робочого часу). (Вчитель отримує інформацію про свій розклад.)
- Створення звітів про успішність, відвідування і видані завдання, придатних для друку. (Вчитель робить паперові копії інформації, що необхідна йому для прийняття рішень.)
- Ведення електронної записної книжки (щоденника) вчителя. (Вчитель записує необхідну йому інформацію на кожен день.)

Користувачі повинні працювати на віддалених персональних комп'ютерах локальної обчислювальної мережі чи Internet. Налагодження мережі і її вузлів не є обов'язком користувача; цим займаються адміністратор, лаборанти й інші відповідальні за це особи. Комп'ютер користувача (клієнтська машина) повинен мати відповідне технічне і програмне забезпечення для коректної і комфортної роботи із системою.

Вимоги до програмного забезпечення клієнтської машини:

- Операційна система: Будь-яка операційна система з можливістю установки віддаленого мережного з'єднання і графічним інтерфейсом. Операційні системи, що рекомендуються – MS Windows 8/7/XP, Linux.
- Web-оглядач: Будь-який Web-оглядач з підтримкою HTML 4.0, JavaScript 1.0, CSS 2. Web-оглядачі, що рекомендуються: MS Internet Explorer 9,10+ , Google Chrome 16+.

Для роботи з інформаційною системою користувачу потрібний тільки Web-оглядач.

Клієнтські машини можуть мати різні технічні характеристики, які можуть впливати на комфортність роботи користувача. Технічне забезпечення також залежить від операційної системи, що використовується. У якості клієнтських машин рекомендується використовувати персональні комп'ютери з наступними характеристиками:

- Процесор: мікропроцесор класу і3 2000 МГц і вище.
- Оперативна пам'ять: RAM 4096 Мб і більше.
- Жорсткий диск з вільним місцем: 10 Гб і більше.
- Мережна карта Ethernet.

Підготовка до роботи

Клієнтська машина повинна мати відповідні програмне і технічне забезпечення, описані вище. Користувач повинен мати можливість встановити з'єднання з Web-сервером системи по протоколу HTTP 1.1.

Для початку роботи із системою користувач повинен запустити Web-оглядач і ввести адресу Web-сторінки з посиланням на інформаційну підсистему «Робоче місце вчителя». Після натискання на посиланні буде проведена аутентифікація; потім (у випадку успішної аутентифікації) користувач зможе працювати із системою.

Опис операцій

Після проходження успішної аутентифікації, користувач входить у систему. Головне вікно системи представлене на рисунку Б1. Воно розділено на кілька частин (frames):

- Ліворуч – інтерактивне деревоподібне меню, за допомогою якого можна завантажити будь-яку іншу сторінку системи.
- По середині – робоча область вікна, в яку будуть завантажуватися відповідні сторінки системи.
- Угорі – область з назвою системи (заголовок).
- У лівому верхньому куті – елемент (Java-апплет), що надає інформацію про поточні дату і час, а також про поточні тиждень (над рисою/під рисою) і урок, що йде в даний момент. Апплет одержує інформацію про дату і час з Web-сервера.
- Унизу – інформація про розробника системи.

Меню є інтерактивним і деревоподібним (як у програми «Провідник» Windows). Склад меню (елементи) залежить від користувача – елементи меню генеруються програмно і, як правило, індивідуальні для кожного вчителя. Умовно можна сказати, що меню складається з папок і документів. Папки можуть містити вкладені папки і документи. Документи, у даному контексті, є посиланнями на HTML-сторінки чи CGI-сценарії, що генерують HTML-сторінки. Обов'язкові елементи меню (однакові для усіх вчителів):

- Коренева папка меню «АРМ Педагога».
- Документ «Розклад занять».
- Документ «Календар».
- Документ «Записна книжка».
- Документ «Вихід».

Необов'язкові елементи меню: папки з назвами предметів, що викладає вчитель; вкладені в них папки з номерами класів учнів, для яких викладають предмети; вкладені в них папка «Звіти» і документи «Відвідування», «Успішність» і «Завдання»; у папці «Звіти» також містяться документи «Відвідування», «Успішність» і «Завдання», що трохи відрізняються від названих вище.

Розгорнута структура меню представлена на рисунку Б2.

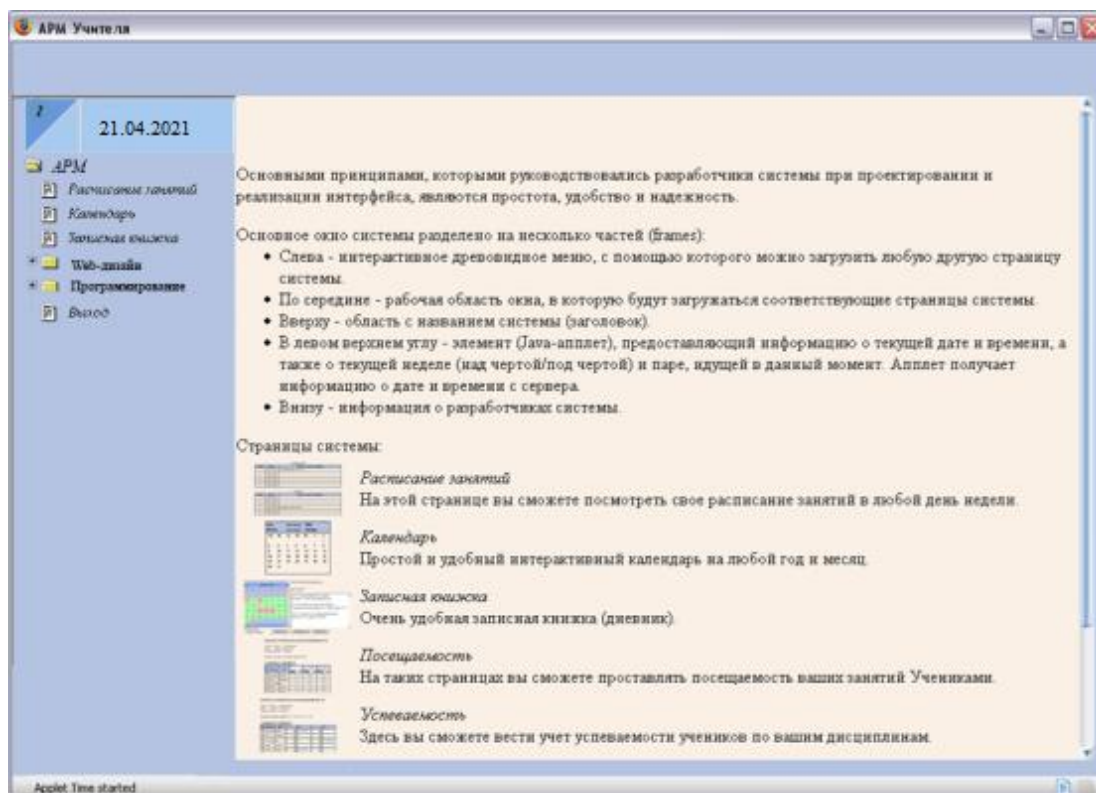


Рисунок Б1 – Головне вікно системи «Робоче місце вчителя»

Опишемо детально кожен елемент меню.

Документ «АРМ Вчителя / Розклад занять». Містить розклад занять вчителя

Документ «АРМ Вчителя / Календар». Містить простий і зручний календар на будь-який рік і місяць. Натискаючи на посилання «Вперед» і «Назад», можна змінювати рік і місяць.

Документ «АРМ Вчителя / Записна книжка». Містить календар на будь-який рік і місяць і форму для редагування записів щоденника. Щоб завантажити запис на якийсь день, потрібно натиснути на відповідному дні календаря. Для вилучення запису потрібно вилучити весь текст у відповідному полі форми.

Документ «АРМ Вчителя / Вихід». При натисканні на цьому елементі меню головне вікно системи закривається і здійснюється вихід із системи.

Документ «АРМ Вчителя / Назва дисципліни / Номер класу / Відвідування». На цій сторінці можна відзначати відвідуваність занять. Можна ввести дату і номер уроку, на якій проходить заняття і відзначити відсутніх

учнів. Також на цій сторінці міститься інформація про кожного учня: кількість пропущених і відвіданих їм занять, відсоток відвідування й інша.

Документ «АРМ Вчителя / Назва предмета / Номер групи / Успішність». На цій сторінці вчитель зможе виставляти оцінки учням за виконані. У відповідних колонках таблиці вказані завдання, які учень повинен виконати; в ячейках потрібно вибрати оцінку зі списку і вказати дату здачі роботи. Якщо зі списку вибору оцінки вибрати порожній рядок, то оцінка буде вилучена. Також на цій сторінці міститься інформація про завдання: відсоток виконання кожного завдання класом учнів, назва, опис, термін здачі завдання й інша.

Документ «АРМ Вчителя / Назва дисципліни / Номер класу / Завдання». На цій сторінці можна вводити нові завдання, вилучати і змінювати старі. Зі списку потрібно вибрати вид завдання, вказати його номер, опис і термін здачі. Завдання, що вже здавалися учнями, вилучити не можна. Якщо зі списку вибору виду завдання вибрати порожній рядок, то завдання буде вилучено.

Документи «АРМ Вчителя / Назва дисципліни / Номер класу / Звіти / Відвідування», «АРМ Вчителя / Назва дисципліни / Номер класу / Звіти / Успішність» і «АРМ Вчителя / Назва дисципліни / Номер класу / Звіти / Завдання» аналогічні описаним вище, але не мають форм для додавання і зміни даних.

Год:		Назад		Вперед		2021	
Месяц:		Назад		Вперед		Март	
Пн	Вт	Ср	Чт	Пт	Сб	Вс	
1	2	3	4	5	6	7	
8	9	10	11	12	13	14	
15	16	17	18	19	20	21	
22	23	24	25	26	27	28	
29	30	31					

Рисунок Б2 – Календар

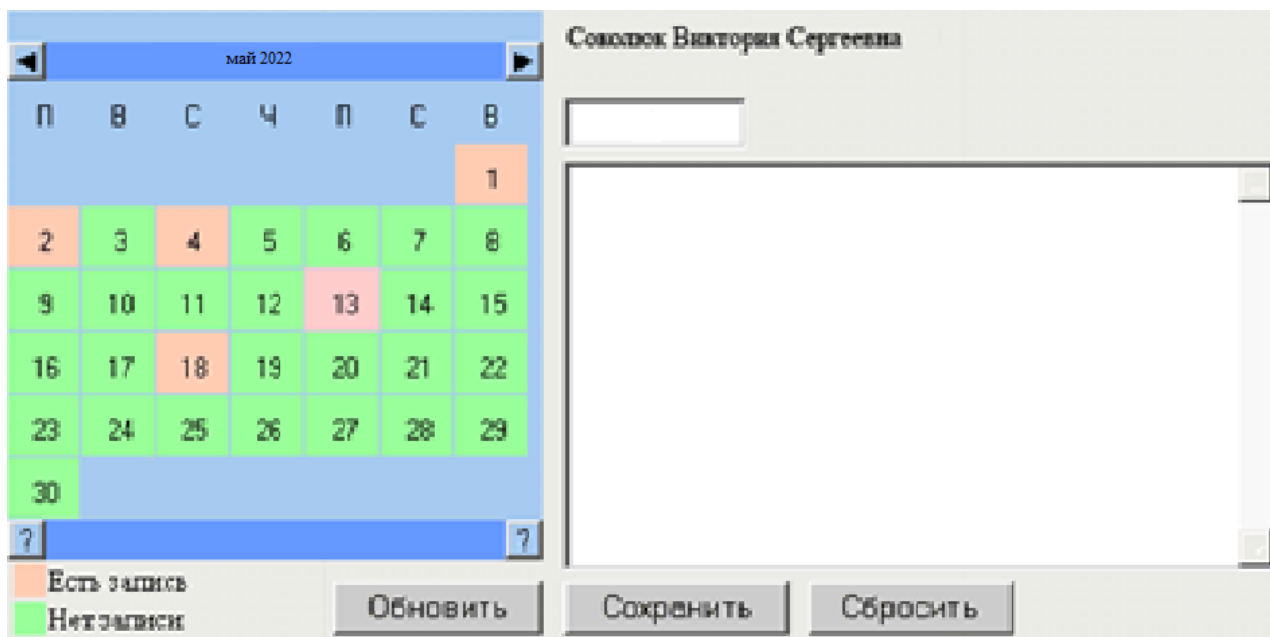


Рисунок Б3 – Записна книжка

Аварійні ситуації

Якщо програма працює неправильно і не видає повідомлень про помилки, то необхідно перевірити наявність відповідного програмного і технічного забезпечення. Приклад такої неправильної роботи – частина функцій системи не працює. Якщо проблема не усунута самостійно, то потрібно звернутися по допомогу до лаборанта чи адміністратора.

Якщо програма видає повідомлення про помилки (наприклад, «лічильник завдань переповнився» чи «неможливо створити об'єкт»), то це означає, що програма працює правильно, але сервера (Web-сервер, сервер доступу до даних і сервер баз даних) налагоджені невірно. Необхідно звернутися до адміністратора системи.

Рекомендації щодо освоєння

Інформації, яка представлена в даному документі, повинно бути досить для освоєння системи. При виникненні додаткових питань користувачі можуть звертатися до адміністратора системи.

Додаток В

Інструкція з формування і ведення бази даних

Правила підготовки даних

Перш, ніж інформаційна система Палацу творчості учнів зможе коректно працювати, необхідно підготувати дані і заповнити базу даних.

Більшість необхідних даних вже існує у відповідних документах і відомостях. Так, існують документи, що містять інформацію про:

- Вчителів Палацу (прізвище, ім'я, по батькові). Наприклад, Сорокіна Олена Ігорівна.
- Учнів (Номер, прізвище, ім'я, по батькові, номер групи). Наприклад, 5, Бондаренко Валерій Олександрович, 10.
- Групи учнів (Номер класу, кількість учнів). Наприклад, 10, 23.
- Групи вчителів (Назва дисципліни, номер групи, прізвища, імена та по батькові вчителів), що викладають предмет у конкретної групи учнів. Наприклад, програмування, 10, Сорокіна Олена Ігорівна, Орман Галина Дмитрівна.
- Дисципліни (Назва дисципліни), що викладаються у палаці. Наприклад, програмування на Delphi.
- Види завдань (Скорочена назва, назва). Наприклад, ДР, домашня робота.
- Оцінки (Назва), що використовуються вчителями для оцінювання роботи учнів. Наприклад, 11 чи відмінно.
- Заняття (Умовний номер заняття, час початку, час закінчення). Наприклад, 1, 8:00, 9:00.
- Дні тижня (Назва). Наприклад, п'ятниця.
- Тижні (Опис). Наприклад, тиждень над рисою.
- Розклад вчителів (Тиждень, день, урок, назва дисципліни, номер групи, аудиторія). Наприклад, тиждень над рисою, п'ятниця, 1, дизайн, 10, 308.

Кожному пункту інформації, що представлена в списку, повинні відповідати спеціальні форми для введення даних у базу даних.

Інформація про тижні, дні, уроки, оцінки і види завдань вже внесена в базу даних, що поставляється з дистрибутивом системи.

Порядок і засоби заповнення бази даних

Базу даних можна заповнювати декількома способами:

- За допомогою спеціально розробленого для цього програмного забезпечення для адміністратора бази даних.
- За допомогою програмного забезпечення для вчителів.
- За допомогою команд СУБД, що використовується.

Програмне забезпечення для адміністратора бази даних повинне використовуватися адміністратором для заповнення бази даних. Перш, ніж із системою зможуть працювати користувачі, адміністратор повинен подбати про занесення інформації в базу даних у наступному порядку:

- Інформація про вчителів школи (Таблиця «Вчитель», форма «Вчитель»).
- Інформація про групи учнів (Таблиця «Група», форма «Група учнів»).
- Інформація про учнів (Таблиця «Учень», форма «Учень»).
- Інформація про дисципліни (Таблиця «Дисципліни», форма «Дисципліна»).
- Інформація про групи вчителів (Таблиці «Група вчителів» і «Вчитель у групі», форма «Група вчителів»).
- Інформація про види завдань (Таблиця «Вид завдання», форма «Вид завдання»).
- Інформація про оцінки (Таблиця «Оцінка», форма «Оцінка»).
- Інформація про уроки (Таблиця «Заняття», форма «Заняття»).
- Інформація про дні тижня (Таблиця «День», форма «День»).
- Інформація про тижні (Таблиця «Тиждень», форма «Тиждень»).

- Інформація про розклад вчителів (Таблиця «Розклад», форма «Розклад»).

Інформація про тижні, дні, уроки, оцінки і види завдань вже занесена в базу даних, що поставляється з дистрибутивом системи.

Користувачі системи заповнюють деякі таблиці бази даних, інформацію в який не має права змінювати ніхто інший. За допомогою програмного забезпечення для вчителів користувачі можуть вводити дані про заняття, відвідування, завдання й успішність. Використання програмного забезпечення для вчителя докладно описано в керівництві користувача.

Для додавання, вилучення і зміни даних можна також використовувати команди СУБД (у даному випадку це MySQL 5.5): INSERT, DELETE, UPDATE і BULK INSERT. Докладніше про ці команди читайте в документації СУБД. Використовувати цей метод не рекомендується, тому що можна порушити цілісність даних.

Процедури зміни і контролю бази даних

Після того, як адміністратор ввів усю необхідну інформацію, що описана вище, він може додавати, вилучати або змінювати дані при необхідності (наприклад, з'явився новий вчитель, про якого немає інформації в базі даних). Та інформація, що вводить адміністратор перед введенням системи в експлуатацію, є практично статичною.

Адміністратору не потрібно вести контроль таблиць, дані в який вводять вчителі. Весь необхідний контроль здійснює програмне забезпечення для вчителя. Адміністратор повинен втрутитися лише в тому випадку, якщо програмне забезпечення видає повідомлення про помилки (наприклад, лічильник завдань переповнився).

Після закінчення навчального півріччя адміністратор повинен зробити копію бази даних і покласти в архів. Після цього потрібно знову підготувати базу даних для наступного півріччя, тому що могли змінитися номери класів учнів, їхній склад і інша інформація.

Порядок і засоби відновлення бази даних

Відповідно до проведеного розрахунку, розмір бази даних на одне півріччя не перевищить 10 Мбайт. Таким чином, база даних має маленький розмір і найбільш доцільно використовувати повне резервне копіювання.

Раз у тиждень адміністратор повинен створювати повну резервну копію за допомогою засобів СУБД чи копіюванням файлів бази даних. У СУБД MySQL 5.5 для цих цілей можна використовувати MySQL Workbench чи PHP MyAdmin. Резервне копіювання потрібно робити на знімний жорсткий диск ємкістю не менше 100 Гбайт або на інші доступні носії інформації.

При виникненні збою в роботі сервера баз даних пов'язаного з інформацією, яка зберігається в базі даних, потрібно скористатися останньою повною резервною копією бази даних для відновлення інформації.

Додаток Г

Програма і методика випробувань

Об'єкт випробувань

Об'єктом випробувань є інформаційна система Палацу творчості учнів, призначена для автоматизації роботи учителів позашкільного закладу освіти.

Цілі випробувань

Цілі випробувань: перевірка придатності системи до використання по призначенню, її відповідність заданим вимогам і документації; перевірка працездатності і правильності роботи системи.

Вимоги до системи

Повний список вимог до системи приведений у пункті 1.5 пояснювальної записки. Нижче приводяться функціональні вимоги до системи.

Система повинна задовольняти наступним функціональним характеристикам:

- 1) Аутентифікація користувача.
- 2) Ідентифікація користувача (надання користувачу тільки тих даних, що необхідні йому для роботи).
- 3) Конфіденційність передачі даних по комп'ютерній мережі.
- 4) Облік успішності учнів:
 - додавання проведеного заняття (уроку);
 - виставляння оцінок учням, що виконали завдання;
 - зміна оцінок учням, що виконали завдання;
 - вилучення оцінок.

5) Облік відвідування учнів:

- додавання проведеного заняття (уроку);
- додавання відмітки про відсутність учня на занятті;
- вилучення відмітки про відсутність учня на занятті.

6) Видача завдань учням:

- додавання нових завдань (лабораторних, домашніх і інших робіт);
- вилучення існуючих завдань;
- зміна існуючих завдань.

7) Надання користувачу (вчителю) інформації про його розклад занять.

8) Надання зручного Web-інтерфейсу:

- однакове представлення інформації незалежно від платформи і програмного забезпечення машини користувача;
- сортування даних;
- проста навігація;
- можливості настройки шрифтів інтерфейсу й інших параметрів інтерфейсу;
- можливість збереження налаштувань інтерфейсу між сеансами роботи;
- можливість автоматичного заповнення полів форм.

9) Надання можливості створення паперових звітів про видані завдання, успішність і відвідування учнів; друк розкладу занять, які проводить вчитель.

10) Ведення записної книжки (щоденника) вчителя.

- додавання запису;
- зміна запису;
- вилучення запису.

11) Можливість віддаленої роботи з інформаційною системою як у локальній мережі, так і в Internet.

Вимоги до програмної документації

У комплект програмної документації повинні входити наступні документи:

- Загальний опис системи.
- Керівництво користувача.
- Інструкція з формування і ведення бази даних.
- Керівництво адміністратора з установки програмного забезпечення на серверах.
- Програма і методика випробувань.
- Текст програми.

Засоби і порядок випробувань

Вимоги до складу і параметрів технічних і програмних засобів вказані в пункті 1.5 пояснювальної записки. Для правильної установки і настройки програмного забезпечення системи необхідно скористатися інструкцією з формування і ведення бази даних і керівництвом адміністратора з установки програмного забезпечення на серверах. Сервера повинні бути об'єднані в комп'ютерну мережу з декількома клієнтами. Як мережне середовище можна використовувати кабелі типу «вита пара» категорії 3, 5 і вище. Мережне обладнання повинно підтримувати швидкість не менше 10 Мбіт/сек.

Випробування системи будуть проводитися протягом двох тижнів. Замовник і/або його представники самостійно проводять тестування усіх функцій системи в присутності розробника. При наявності відповідної можливості, у процесі тестування можуть додатково брати участь програмісти, тестувальники і системотехніки, яки не брали участь у розробці системи.

Перед початком (не пізніше 3 днів) випробувань уточнюється кінцевий склад групи, що буде проводити випробування. Кожний член групи отримує необхідну документацію до системи (технічне завдання, загальний опис системи,

керівництво користувача, інструкція з формування і ведення бази даних, керівництво адміністратора з установки програмного забезпечення на серверах, програма і методика випробувань).

Порядок проведення випробувань:

- Виконуються контрольні тести в довільному порядку. Результати заносяться у відповідний журнал.
- Робиться аналіз отриманих результатів і встановлюється відповідність програми заданим вимогам і системним документам.

При виявленні помилок у роботі системи складається їхній перелік і обговорюється термін їхнього виправлення розробником. Після цього замовник проводить повторне тестування в повному обсязі (можливо використання нових чи додаткових тестів).

Методика випробувань

Процес тестування являє собою експлуатацію системи в контрольованих умовах і вивчення отриманих результатів. При цьому перевіряється робота системи з нормальними і помилковими даними і подіями. Тестування розподілених систем сильно ускладнено великою кількістю взаємозв'язків. Кожна залежність, Web-сторінка, компонент, база даних, а також код графічного інтерфейсу, програмне забезпечення середнього рівня і мережна інфраструктура повинні тестуватися не тільки на коректність роботи, але і на сумісність один з одним у різних конфігураціях.

Найкраще для тестування розподілених систем застосовувати метод «знизу-нагору». Спочатку розробник повинен перевірити кожен компонент (елемент системи) окремо; потім – протестувати взаємодію компонента з іншими компонентами системи на автономному комп'ютері; і тільки потім – перевірити роботу всієї системи в розподіленому середовищі. Усі ці види тестування виконуються розробником системи без участі замовника. Розробник не зобов'язаний надавати тести і протоколи результатів тестування кому-небудь.

При прийманні системи проводяться випробування системи в розподіленому середовищі. Замовник і/або його представники самостійно проводять тестування усіх функцій системи методом «чорного ящика» у присутності розробника.

Програма випробувань

Тести складаються замовником і/або його представниками. При цьому перевіряється функціонування системи з нормальними і помилковими даними і подіями.

Можливий перелік тестів, які дозволять протестувати основні функції системи:

- 1) Аутентифікація користувача з правильними даними.
- 2) Аутентифікація користувача з неправильними даними.
- 3) Перегляд розкладу.
- 4) Перегляд форми з обліком успішності.
- 5) Перегляд форми з обліком відвідування.
- 6) Перегляд форми з виданими завданнями.
- 7) Перегляд записів у записній книжці.
- 8) Додавання проведеного заняття (уроку) до успішності із правильними даними.
- 9) Додавання проведеного заняття (уроку) до успішності з неправильними даними.
- 10) Зміна дати проведеного заняття (пари) в успішності з правильними даними.
- 11) Зміна дати проведеного заняття (уроку) в успішності з неправильними даними.
- 12) Виставляння оцінок в успішність.
- 13) Зміна оцінок в успішності.
- 14) Вилучення оцінок з успішності.

- 15) Додавання проведеного заняття (уроку) до відвідування із правильними даними.
- 16) Додавання проведеного заняття (уроку) до відвідування з неправильними даними.
- 17) Зміна дати проведеного заняття (уроку) у відвідуванні з правильними даними.
- 18) Зміна дати проведеного заняття (уроку) у відвідуванні з неправильними даними.
- 19) Додавання відмітки про відсутність учня на занятті до відвідування.
- 20) Вилучення відмітки про відсутність учня на занятті з відвідування.
- 21) Додавання нових завдань із правильними даними.
- 22) Додавання нових завдань з неправильними даними.
- 23) Вилучення існуючих завдань.
- 24) Зміна існуючих завдань із правильними даними.
- 25) Зміна існуючих завдань з неправильними даними.
- 26) Перегляд звітів про успішність.
- 27) Перегляд звітів про відвідування.
- 28) Перегляд звітів про видані завдання.
- 29) Додавання запису до записної книжки з правильними даними.
- 30) Додавання запису до записної книжки з неправильними даними.
- 31) Зміна запису в записній книжці з правильними даними.
- 32) Зміна запису в записній книжці з неправильними даними.
- 33) Вилучення запису з записної книжки.

Під неправильними чи помилковими даними в контексті даної системи потрібно розуміти:

- Логін і/або пароль користувача, яких немає у файлах аутентифікації (для перших двох тестів).
- Неіснуюча дата (для всіх інших тестів).

Програмне забезпечення розроблено таким чином, що користувач не може ввести які-небудь інші неправильні дані в систему через наданий йому

інтерфейс. Таким чином, будь-які дані, що не є неправильними, потрібно вважати правильними.

Для визначення переліку вхідних даних для кожного тесту можна скористатися формами для введення відповідних даних. У формах зазначені необхідні поля.

Вихідні дані чи результат при правильних даних повинні відповідати виконанню функції, для тестування якої створений тест; при неправильних даних – відмовлення в доступі до системи (тест 2) чи повідомлення про помилку.

Додаток Д

Текст програми

Вступ

Програмне забезпечення інформаційної системи Палацу творчості учнів складається з 27 модулів, 26 з яких було реалізовано розробником з використанням декількох мов програмування: C++, Java, JavaScript, HTML, CSS. Розмір первісного програмного коду всіх модулів наближається до 10000 рядків, тому наводити його повністю недоцільно. Для прикладу, наведемо текст двох модулів:

Текст модуля аутентифікації – сторінка входу

```
<%@ page language="java" contentType="text/html; charset=UTF-8"
pageEncoding="UTF-8"%>
<!DOCTYPE html PUBLIC "-//W3C//DTD HTML 4.01 Transitional//EN"
"http://www.w3.org/TR/html4/loose.dtd">
<html>
<head>
<meta http-equiv="Content-Type" content="text/html; charset=UTF-8">
<title>Login</title>
</head>
<body>
<p>Please enter your username and password to access the application</p>
<form method="POST" action="j_security_check">
<table cellpadding="0" cellspacing="0" border="0">
<tr>
<td align="right">Username:&nbsp;  </td>
<td><input type="text" name="j_username"></td>
</tr>
<tr>
<td align="right">Password:&nbsp;  </td>
<td><input type="password" name="j_password"></td>
</tr>
<tr>
<td></td>
<td><input type="submit" value="Login"></td>
</tr>
</table>
</form>
</body>
```


</html>

Текст модуля авторизації – сторінка «Помилка»

```
<%@ page language="java" contentType="text/html; charset=UTF-8"
pageEncoding="UTF-8"%>
<!DOCTYPE html PUBLIC "-//W3C//DTD HTML 4.01 Transitional//EN"
"http://www.w3.org/TR/html4/loose.dtd">
<html>
<head>
  <meta http-equiv="Content-Type" content="text/html; charset=UTF-8">
  <title>Login Error</title>
</head>
<body>
  There was an error logging in. Please try again.<br />
  <form method="POST" action="j_security_check">
    <table cellpadding="0" cellspacing="0" border="0">
      <tr>
        <td align="right">Username:&nbsp;  </td>
        <td><input type="text" name="j_username"></td>
      </tr>
      <tr>
        <td align="right">Password:&nbsp;  </td>
        <td><input type="password" name="j_password"></td>
      </tr>
      <tr>
        <td></td>
        <td><input type="submit" value="Login"></td>
      </tr>
    </table>
  </form>
</body>
</html>
```

Текст модуля «Опрацювання результатів»

```
import javax.servlet.ServletException;
import javax.servlet.http.HttpServletRequest;
import javax.servlet.http.HttpServletResponse;
import javax.servlet.http.HttpSession;
import java.io.BufferedReader;
import java.io.IOException;
import java.io.InputStream;
import java.io.InputStreamReader;
import java.net.URL;
import java.util.*;
import java.util.logging.Level;
import java.util.logging.Logger;

@WebServlet(name = "StartTestServlet", urlPatterns
= {"/teststart", "/testrun", "/testfinish", "/sendanswer"})
public class StartTestServlet extends javax.servlet.http.HttpServlet {
```

```

public static final String FILENAME = "qcpp.txt";
public static final int NUM_OF_QUESTIONS = 20;

List<Question> questionsList;

@Override
public void init() throws ServletException {
    super.init();
    questionsList = readBlock(FILENAME);
}

protected void doPost(javax.servlet.http.HttpServletRequest request,
    javax.servlet.http.HttpServletResponse response) throws
    javax.servlet.ServletException, IOException {
    processRequest(request, response);
}

protected void doGet(javax.servlet.http.HttpServletRequest request,
    javax.servlet.http.HttpServletResponse response) throws
    javax.servlet.ServletException, IOException {
    //      String parameter = request.getParameter("test");
    //      if (parameter != null) {
    //          questionsList = readBlock(parameter+".txt");
    //      }
    processRequest(request, response);
}

private void processRequest(HttpServletRequest request,
    HttpServletResponse response) throws ServletException, IOException {
    String servletPath = request.getServletPath();
    switch (servletPath) {
        case "/teststart":
            starttest(request, response);
            break;
        case "/testrun":
            runtest(request, response);
            break;
        case "/testfinish":
            finishtest(request, response);
            break;
        case "/sendanswer":
            sendanswer(request, response);
            break;
        default: response.sendRedirect("index.jsp");
    }
}

private void sendanswer(HttpServletRequest request, HttpServletResponse
response) throws IOException {
    HttpSession session = request.getSession();
    if (session.isNew() || session.getAttribute("questions")==null) {
        response.sendRedirect("index.jsp");
    }
}

```

```

    } else {
        Integer current = (Integer) session.getAttribute("current");
        List<Question> questions = (List<Question>)
session.getAttribute("questions");
        int numberQ = questions.size();
        String ans = request.getParameter("ans");
        if ("Next".equals(ans)) {
            current = (current + 1) % numberQ;
        }
        if ("Prev".equals(ans)) {
            current = (current + numberQ - 1) % numberQ;
        }
        if ("Accept".equals(ans)) {
            int[] answers = (int[]) session.getAttribute("answers");
            String q = request.getParameter("q");
            if (q != null) {
                int a = Integer.parseInt(q);
                answers[current] = a;
                session.setAttribute("answers", answers);
            }
            current = (current + 1) % numberQ;
        }
        if ("Finish".equals(ans) &&
"1".equals(request.getParameter("finish"))) {
            int[] answers = (int[]) session.getAttribute("answers");
            String q = request.getParameter("q");
            if (q != null) {
                int a = Integer.parseInt(q);
                answers[current] = a;
                session.setAttribute("answers", answers);
            }
            response.sendRedirect("testfinish");
        } else {
            session.setAttribute("current", current);
            response.sendRedirect("testrun");
        }
    }
}

```

```

private void finishtest(HttpServletRequest request, HttpServletResponse
response) throws ServletException, IOException {
    HttpSession session = request.getSession();
    if (session.isNew() || session.getAttribute("questions")==null) {
        response.sendRedirect("index.jsp");
    } else {
        List<Question> questions = (List<Question>)
session.getAttribute("questions");
        int[] answers = (int[]) session.getAttribute("answers");
        List<TriBean> result = new ArrayList<>();
        int totalCorrect = 0;
        for (int i = 0; i < answers.length; i++) {
            TriBean tb = new TriBean(i, answers[i],
questions.get(i).getCorrect());

```

```

        if (answers[i] == questions.get(i).getCorrect()) {
            totalCorrect++;
        }
        result.add(tb);
    }
    request.setAttribute("result", result);
    request.setAttribute("total", totalCorrect);
    request.setAttribute("wrong", answers.length-totalCorrect);
    request.setAttribute("fio", session.getAttribute("fio"));
    request.setAttribute("group", session.getAttribute("group"));
    request.setAttribute("starttime",
session.getAttribute("starttime"));
    session.invalidate();
    request.getRequestDispatcher("/finish.jsp").forward(request,
response);
    }
}

private void runtest(HttpServletRequest request, HttpServletResponse
response) throws IOException, ServletException {
    HttpSession session = request.getSession(false);
    if (session == null || session.isNew() ||
session.getAttribute("current") == null) {
        response.sendRedirect("index.jsp");
    } else {
        Integer current = (Integer)session.getAttribute("current");
        Integer answer =
((int[])session.getAttribute("answers"))[current];
        List<Question> questions = (List<Question>)
session.getAttribute("questions");
        Question question = questions.get(current);
        String pict = question.getPicture();
        String picture = "";
        if (!pict.trim().isEmpty()){
            picture = "<img
src=\"http://berkut.homelinux.com/download/test/img/"+pict+"\" /> <br/>";
        }
        request.setAttribute("picture", picture);
        AnswerVariant av = new AnswerVariant(question.getAnswers(),
answer);
        request.setAttribute("av", av);
        request.setAttribute("q", question);
        request.getRequestDispatcher("/run.jsp").forward(request,
response);
    }
}

private void starttest(HttpServletRequest request, HttpServletResponse
response) throws IOException {
    String fio = request.getParameter("fio");
    String group = request.getParameter("group");
    HttpSession session = request.getSession();
    List<Question> questions = createTest();
    session.setAttribute("questions", questions);
    session.setAttribute("group", group);

```

```

        session.setAttribute("fio", fio);
        session.setAttribute("starttime", new Date());
        session.setAttribute("current", 0);
        session.setAttribute("total", questions.size());
        int[] answers = new int[questions.size()];
        for (int i=0; i<answers.length; i++) {
            answers[i] = -1;
        }
        session.setAttribute("answers", answers);
        response.sendRedirect("testrun");
    }
    private List<Question> createTest() {
        List<Question> result = new ArrayList<>(questionsList);
        Collections.shuffle(result);
        return result.subList(0, NUM_OF_QUESTIONS);
    }
    private List<Question> readBlock(String fileName) {
        List<Question> block = new ArrayList<>();
        try
            (InputStream openStream = new
URL("http://ndtu.mk.ua/download/test/"+fileName).openStream()) { // url
            BufferedReader in = new BufferedReader(new
InputStreamReader(openStream, "cp1251")); // url
            String question;
            while ((question=in.readLine())!=null) {
                String pictureName = in.readLine();
                String addition = in.readLine();
                String[] answers = new String[4];
                List<String> ansList = new ArrayList<>(4);
                int correct = -1;
                for (int i=0; i<answers.length; i++) {
                    ansList.add(in.readLine());
                }
                Collections.shuffle(ansList);
                for (int i=0; i<answers.length; i++) {
                    String ans = ansList.get(i);
                    if (!ans.isEmpty() && ans.charAt(0)=='+') {
                        correct = i;
                        answers[i] = ans.substring(1);
                    } else answers[i] = ans;
                }
                Question q = new Question(question, pictureName, addition,
answers, correct);
                block.add(q);
                in.readLine();
            }
            return block;
        } catch (IOException ex) {
            Logger.getLogger(StartTestServlet.class.getName()).log(Level.SEVERE, null,
ex);
        }
        return block;
    }
}

```