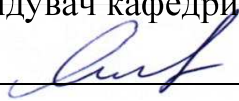


МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ

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.

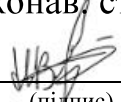
" ____ " _____ 2025 р.

КВАЛІФІКАЦІЙНА РОБОТА

на здобуття ступеня вищої освіти "бакалавр"

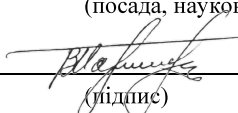
на тему: Розроблення вебзастосунку закладу позашкільної освіти

Виконав: студент групи 4141



Дума М.В.
(підпис)

Керівник роботи:

(посада, науковий ступінь вчене звання)


Парнас В.К.
(підпис)

м. Миколаїв – 2025 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. Гайда А.Ю.

" ____ " ____ 2025 р.

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

Студенту Думі Максиму Вікторовичу

(прізвище, ім'я, по батькові)

1. Тема роботи: Розроблення вебзастосунку закладу позашкільної освіти

Керівник роботи: к.т.н., доц. Пармас Віктор Кирилович

Затверджені наказом ректора № 295-уч від "27" березня 2025 року.

2. Термін подання роботи: 20 червня 2025 р.

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

4. Перелік питань, що належать до розробки (найменування розділів):
аналіз предметної області і постановка задачі, розробка проекту інформаційної системи, реалізація проекту інформаційної системи, розділ з охорони праці, висновки.

5. Перелік презентаційних матеріалів Організаційна структура, Концептуальна модель ІС, Діаграма прецедентів, Діаграма діяльності, ER-діаграма (сутність-зв'язок), Фізична модель даних, Діаграма класів, Використані технології, Реалізація вебзастосунку, Результати впровадження

6. Консультанти розділів роботи

Розділ	Прізвище, ініціали та посада консультанта	Підпис, дата	
		завдання видав	завдання прийняв
1	Партас В.К.	03.02.2025	22.02.2025
2	Партас В.К.	22.02.2025	21.03.2025
3	Партас В.К.	21.03.2025	25.04.2025
4	Партас В.К.	25.04.2025	07.06.2025

7. Дата видачі завдання 14 лютого 2022 р.

КАЛЕНДАРНИЙ ПЛАН

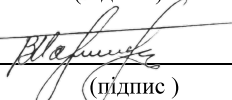
№ з/п	Назва етапів дипломного проекту (роботи)	Термін виконання етапів проекту	Примітка
1	Аналіз предметної області	13.02.2025	
2	Постановка задачі	15.02.2025	
3	Розробка концепції	22.02.2025	
4	Розробка проекту ІС	21.03.2025	
5	Реалізація проекту	25.04.2025	
6	Розробка питань з охорони праці	07.06.2025	
7	Розробка графічних матеріалів	12.06.2025	
8	Подання роботи на перевірку на плагіат	10.06.2025	
9	Подання роботи рецензенту	15.06.2025	
10	Подання роботи на захист	20.06.2025	

Студент


(підпис)

Дума Максим Вікторович
(ПІБ)

Керівник роботи


(підпис)

Партас Віктор Кирилович
(ПІБ)

АНОТАЦІЯ

Дана дипломна робота має спрямованість на набуття навичок у розробці проекту інформаційної системи, на прикладі розроблення вебзастосунку інформаційної системи закладу позашкільної освіти.

Робота виконана на _78_ сторінках, містить _4_ розділи, _12_ рисунків, _4_ додатки і список використаних джерел з _15_ найменувань.

Ключові слова: Java, MySQL, СУБД, Spring, вебзастосунок, інформаційна система.

ABSTRACT

This diploma work is aimed at acquiring skills in creating and development an web application for information system project, on the example of developing an information system for accounting students of out-of-school educational institutions.

The work is presented on _78_ pages, contains _4_ sections, _12_ pictures, _4_ applications and _15_ references to the sources.

Keywords: Java, MySQL, DBMS, Spring, web application, information system.

ЗМІСТ

ВСТУП	
РОЗДІЛ 1. ФОРМУВАННЯ ВИМОГ ДО ВЕБЗАСТОСУНКУ ЗАКЛАДУ ПОЗАШКІЛЬНОЇ ОСВІТИ, РОЗРОБКА ЇЇ КОНЦЕПЦІЇ І ПОСТАНОВКА ЗАДАЧІ...	
1.1 Опис предметної області і аналіз інформаційних процесів позашкільного навчального закладу	
1.2 Формування вимог до інформаційної системи	
РОЗДІЛ 2. РОЗРОБКА ТЕХНІЧНОГО ПРОЕКТУ ВЕБЗАСТОСУНКУ ІНФОРМАЦІЙНОЇ СИСТЕМИ ПОЗАШКІЛЬНОГО НАВЧАЛЬНОГО ЗАКЛАДУ....	
2.1 Загальносистемні рішення	
2.2 Рішення з інформаційного забезпечення	
2.3 Рішення з програмного забезпечення	
2.4 Рішення з технічного забезпечення	
РОЗДІЛ 3. РЕАЛІЗАЦІЯ ВЕБЗАСТОСУНКУ ІНФОРМАЦІЙНОЇ СИСТЕМИ ПОЗАШКІЛЬНОГО НАВЧАЛЬНОГО ЗАКЛАДУ.....	
РОЗДІЛ 4. ОХОРОНА ПРАЦІ ТА ТЕХНІКА БЕЗПЕКИ	
ВИСНОВОК	
СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ	
ДОДАТОК А. ТЕХНІЧНЕ ЗАВДАННЯ НА РОЗРОБКУ	
ДОДАТОК Б. ПРОГРАМА І МЕТОДИКА ВИПРОБУВАНЬ	
ДОДАТОК В. ПРИКЛАДИ ЕКРАННИХ ФОРМ	
ДОДАТОК Г. КОД ПРОГРАМИ ТА КОД СТВОРЕННЯ ТАБЛИЦЬ БАЗИ ДАНИХ.....	

ВСТУП

Темою дипломної роботи є «Розроблення вебзастосунку закладу позашкільної освіти». Розроблення зазначеного вебзастосунку у межах інформаційної системи позашкільного закладу є актуальним тому, що використання її у позашкільному навчальному закладі зможе полегшити облік учнів, підвищити продуктивність керівників гуртків та педагогів.

Метою роботи є автоматизація роботи позашкільного навчального закладу через розробку вебзастосунку інформаційної системи позашкільних навчальних закладів, яка забезпечить більш комфортні умови праці, зручну структурування груп та підвищить продуктивність співробітників. Об'єкт дослідження – корпоративні застосунки, призначені для автоматизації роботи навчальних закладів, на прикладі вебзастосунку інформаційної системи Миколаївського Палацу творчості учнів. Предметом дослідження є інформаційні процеси, що виконуються співробітниками навчального закладу та засоби їхньої автоматизації.

Аналіз роботи педагогів та керівників гуртків позашкільного закладу сформував вимоги до проекту інформаційної системи. Концепція вирішує завдання реєстрації учнів, обліку, розподілу між групами, і створення звітів. Для реалізації системи обрано сучасні програмні та технічні засоби. Розробляються інструкції користувача, формування бази даних та початку роботи з системою. Проведено аналіз небезпечних факторів у закладі, розроблено техніку безпеки та заходи для зменшення їх впливу на робочому місці.

РОЗДІЛ 1 ФОРМУВАННЯ ВИМОГ ДО ВЕБЗАСТОСУНКУ ЗАКЛАДУ ПОЗАШКІЛЬНОЇ ОСВІТИ, РОЗРОБКА ЇЇ КОНЦЕПЦІЇ І ПОСТАНОВКА ЗАДАЧІ

1.1 Опис предметної області і аналіз інформаційних процесів позашкільного навчального закладу

Позашкільний навчальний заклад (заклад позашкільної освіти) - складова системи позашкільної освіти, яка надає знання, формуючи вміння та навички за інтересами, забезпечує потреби особистості у творчій самореалізації та інтелектуальний, духовний і фізичний розвиток, підготовку до активної професійної та громадської діяльності, створює умови для соціального захисту та організації змістовного дозвілля відповідно до здібностей, обдарувань та стану здоров'я вихованців, учнів і слухачів [1].

Позашкільні навчальні заклади займаються організацією навчання дітей та підлітків згідно їх інтересів та спрямованості у позашкільний час. Учні поділяються на окремі групи згідно з їх віком, класом, навичкам та ін.

Приблизна організаційна структура окремого закладу позашкільної освіти:

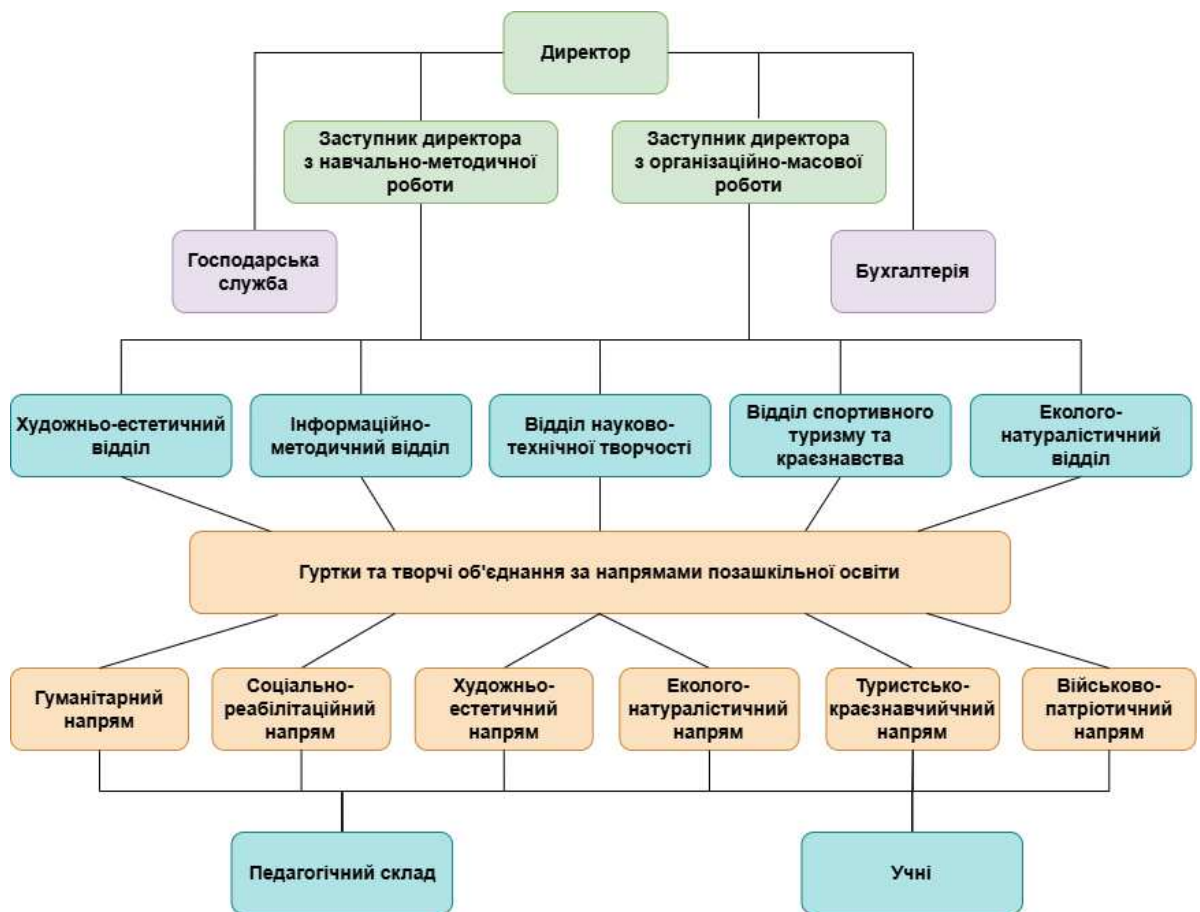


Рисунок 1.1 – Організаційна структура позашкільного навчального закладу

Директор керує закладом позашкільної освіти, стежить за виконанням обов'язків іншими працівниками закладу, виконує представницьку функцію та посередницьку функцію між Міністерством Освіти та педагогами.

Господарська служба виконує роботи з забезпечення умов праці, поточного ремонту та обліку наявної техніки.

Бухгалтерія виконує бухгалтерський облік, виплату заробітної плати та займається плануванням бюджету навчального закладу.

Заступники директора виконують комунікацію між директором та педагогічним складом, стежать за виконанням обов'язків.

Гуртки та творчі об'єднання за напрямками позашкільної освіти існують відповідно до спрямування навчального закладу, створюють відповідні

навчальні програми, організовують навчальний процес, повідомляють про набір до гуртків, беруть участь в конкурсах, мають педагогічний склад.

Проведений аналіз роботи бухгалтерії та господарської служби підтвердив наявність простих та якісних налагоджених рішень, що дозволяють ефективно налагодити фінансову та господарську діяльність в позашкільному навчальному закладі.

За результатом аналізу роботи гуртків та творчих об'єднань було виявлено, що набір учнів проводиться педагогами використовуючи застарілі системи та в паперовому вигляді, вибірки для звітів виконуються в ручному вигляді, не використовуючи будь-які засоби автоматизації. Це підтверджує зниження ефективності педагогів, зменшення часу на підготовку навчального процесу, а відповідно і зменшення якості позашкільної освіти. Є необхідність вирішення проблеми продуктивності та ефективності роботи педагога.

Можливі проблеми при здійсненні роботи педагога:

- 1) Неправильне або некоректне заповнення заявок до гуртка батьками учнів;
- 2) Зберігання некоректної або неповної інформації про групи;
- 3) Несвоєчасна перевірка заявок до гуртка чи творчого об'єднання;
- 4) Некоректні вибірки даних для звітів.

Шляхами вирішення зазначених проблем можуть бути:

- 1) Використання форм з автоматичною валідацією даних;
- 2) Автоматичне сповіщення вчителя про отримання нових заявок до групи;
- 3) Генерація вчителем відповідних вибірок учнів для звітів.

За результатами проведеного аналізу сформовано вимоги до інформаційної системи.

1.2 Формування вимог до інформаційної системи

На основі проблем, з якими стикаються у навчальному закладі, сформовано такі вимоги за функціоналом:

- 1) Структуроване зберігання даних;
- 2) Приймання даних відбувається після їх перевірки;
- 3) Автоматичне сповіщення вчителя про створені заяви;
- 4) Автоматичне сповіщення учнів про результат розгляду заяви;
- 5) Шаблонне виведення даних для звітів.

Для виконання функціональних вимог необхідні наступні програмні засоби:

- 1) Операційна системи;
- 2) Мови програмування для розробки веб-додатку;
- 3) Реляційна СУБД;
- 4) Веб-браузер для користування інформаційною системою.

Для розробки необхідні такі програмні засоби:

- 1) IDE – інтегроване середовище розробки;
- 2) Система контролю версій;
- 3) Фреймворк генерування документації програмного коду;
- 4) Графічний редактор для створення діаграм.

1.3 Розробка концепції інформаційної системи

За результатом сформованих вимог розроблено концепцію інформаційної системи та її модель, відповідно до функціональних вимог.

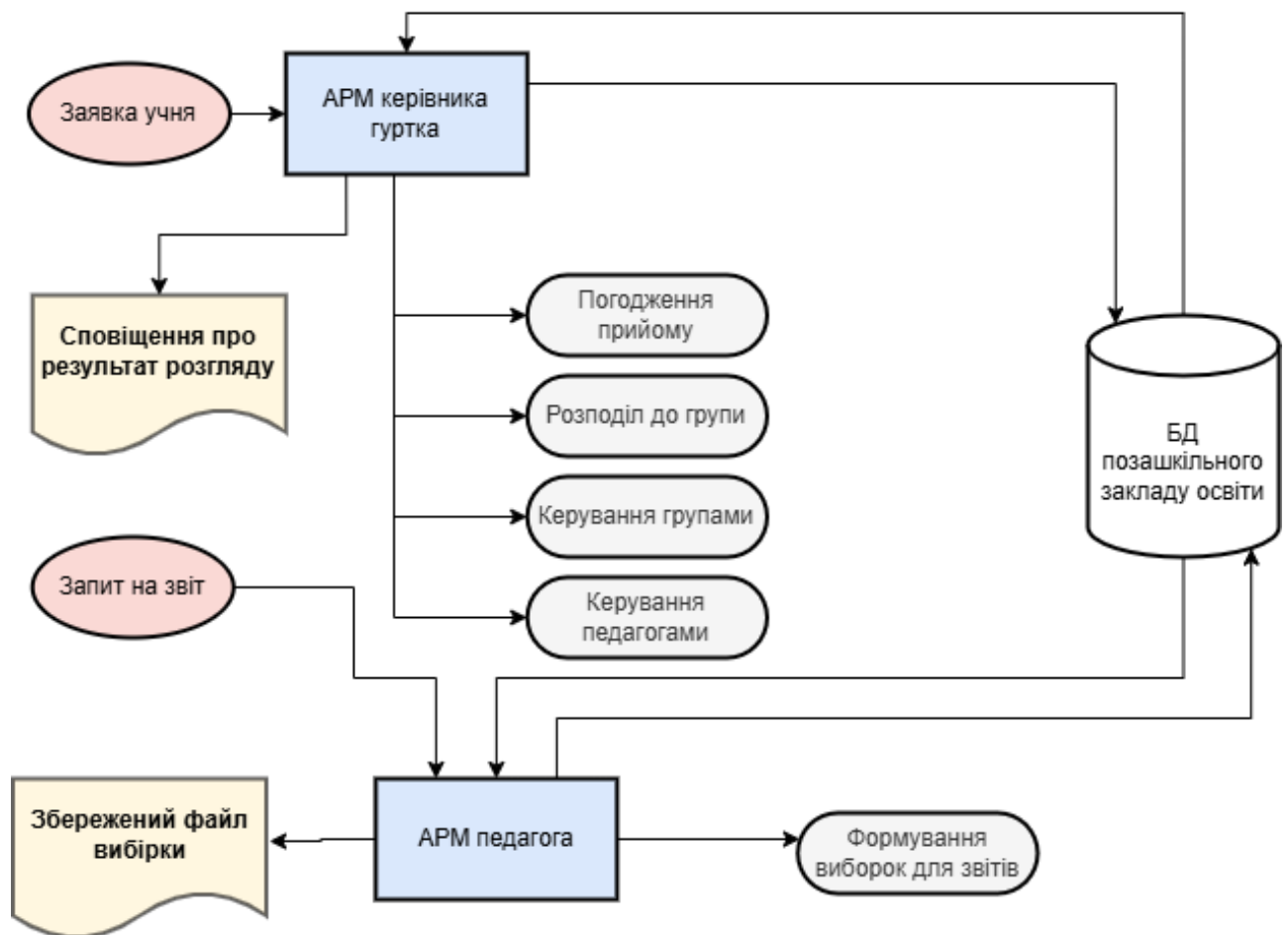


Рисунок 1.2 – Модель інформаційної системи

Виходячи зі схеми моделі, певний користувач, підключившись до інформаційної системи використовуючи додаток, може працювати зі своїм автоматизованим робочим місцем [2], всі дані для кого будуть отримані і будуть зберігатися в базі даних позашкільного навчального закладу.

За аналізом розробленої моделі виділено основні частини:

- 1) АРМ керівника гуртка;
- 2) АРМ педагога;
- 3) Система управління базою даних;
- 4) Технічні засоби;
- 5) Програмні засоби;

АРМ керівника гуртка автоматизує наступні задачі:

- 1) Розподіл до групи: при отриманні заявки нового учня, обирає відповідну групу для нього;
- 2) Погодження прийому: здійснюється перевірка наявності місць в групах та коректність вказаних даних;
- 3) Керування групами: здійснюється створення, заміна, видалення груп відповідно до потреб гуртка;
- 4) Керування педагогами: здійснюється прикріплення педагогів до груп, додавання нових, зміна їх даних.

АРМ педагога автоматизує формування вибірок для звітів: здійснюється на основі певного запиту на звіт, дозволяє згенерувати документ для збереження вибірки;

Для створення вебзастосунку корпоративної інформаційної системи на платформі Java найчастіше обирають фреймворк Spring. При використанні цього фреймворку інформаційна система, що створюється, буде мати трирівневу структуру (рис.1.3)



Рисунок 1.3 – Тривірнева структура вебзастосунку Spring

Рівень представлень, або Web Layer – це шаблони сторінок застосунку, з якими безпосередньо взаємодіє користувач, засоби html, css та JavaScript а також класи контролерів, що можуть розроблятися мовами Java, Kotlin, Groovy тощо.

Сервісний рівень – рівень бізнес-логіки, реалізується мовами програмування високого рівня.

Рівень сховища та доступ до даних реалізується за рахунок взаємодії засобів мов програмування Java, Kotlin тощо з засобами СУБД. Найпоширенішими з яких є MySQL, PostgreSQL, H2, SQLite.

У цій роботі системою управління базою даних було обрано вільну об'єктно-реляційну систему – PostgreSQL.

Технічні засоби

Для нормальної роботи клієнтської сторони, вона повинна мати наступні технічні характеристики:

- 1) Процесор – Intel Core 2 Duo 2 GHz і продуктивніші;
- 2) Графічна карта – 256 MB;
- 3) Оперативна пам'ять – 2 GB і більше;
- 4) Мережева карта – 100 Mbit;
- 5) Мишка;
- 6) Клавіатура;
- 7) Монітор, роздільна здатність дисплею якого 1024x768 та більше.

Для серверної сторони рекомендованими системними вимогами до комплектуючих комп'ютера є:

- 1) Процесор – Intel Core 2 Duo 2.5 GHz і продуктивніші;
- 2) Графічна карта – 256 MB;

- 3) Оперативна пам'ять – 4 GB і більше;
- 4) Мережева карта – 100 Mbit;
- 5) Місце на жорсткому диску – 20 GB;
- 6) Мишка;
- 7) Клавіатура;
- 8) Монітор, роздільна здатність дисплею якого 1024x768 і більше.

Відповідно, в кожного педагога має бути доступ до робочого комп'ютера підключеного до мережі Інтернет або смартфона з доступом до веб-додатку.

Програмні засоби

Розробка веб-додатку буде виконуватись використовуючи об'єктно-орієнтовану мову програмування Java, використовуючи IDE (інтегроване середовище розробки) IntelliJ IDEA Community Edition. Для розробки веб-додатку було обрано типове рішення - фреймворк Spring, для налаштування та зручної конфігурації було обрано один з проєктів Spring - Spring Boot. Для розробки інтерфейсу користувача додатка обрано Vue.JS - JavaScript-фреймворк і його бібліотеку UI-компонентів Buefy. У якості системи контролю версій має використовуватися Git та веб-сервіс GitHub, який інтегрується з середою розробки IntelliJ IDEA.

Завдяки тому, що система має бути у вигляді веб-додатку, вона є кросплатформною і для роботи з нею потрібна будь-яка операційна система та будь-який веб-браузер. Операційною системою для комп'ютерів обрано Linux Mint, а для смартфонів - Android, в обох випадках браузер Google Chrome.

Для створення діаграм було обрано онлайн графічний редактор Diagrams.net. Для документації коду потрібно використовувати Javadoc - автоматичний генератор документації в HTML-форматі з коментарів

вихідного коду мовою Java, а також Swagger - фреймворк для документування REST-сервісу.

Сервісні засоби, що необхідні для функціонування системи:

- 1) Стабільне інтернет підключення;
- 2) Електронна пошта;

Для інтернет підключення всередині позашкільного навчального закладу використовуватиметься технологія Ethernet.

У процесі роботи позашкільних навчальних закладів з інформаційною системою виникає потреба у отриманні повідомлень про нові заявки до гуртка, а також для взаємодії між співробітниками навчального закладу. Для покриття цього завдання необхідно використати електронну пошту. Для цього було обрано Gmail у якості поштового сервісу – безкоштовна поштова служба від компанії Google.

В навчальному закладі на стаціонарних комп'ютерах зв'язок з сервером буде здійснюватися за допомогою мережі Інтернет. Для доступу до серверу зі смартфона, використовується 4G – четверте покоління мобільного радіозв'язку.

Висновок

Отже, реалізація інформаційної системи дозволить досягнути цілей закладу позашкільної освіти за рахунок автоматизації типових процесів та раціонального використання часу співробітників. Завдяки пришвидшенню виконання типової роботи і структуруванню даних, педагоги матимуть більше часу на підготовку до занять, створення навчального плану та навчальних матеріалів, що покращить рівень навчального процесу.

РОЗДІЛ 2. РОЗРОБКА ТЕХНІЧНОГО ПРОЕКТУ ВЕБЗАСТОСУНКУ ІНФОРМАЦІЙНОЇ СИСТЕМИ ПОЗАШКІЛЬНОГО НАВЧАЛЬНОГО ЗАКЛАДУ

2.1 Загальносистемні рішення

Виходячи з концепції визначено, що користувачами інформаційної системи будуть педагоги, керівник гуртка, адміністрація позашкільного навчального закладу і учні (їх батьки). Інформаційні процеси користувачів інформаційної системи та їх взаємодію відображено на діаграмі прецедентів.

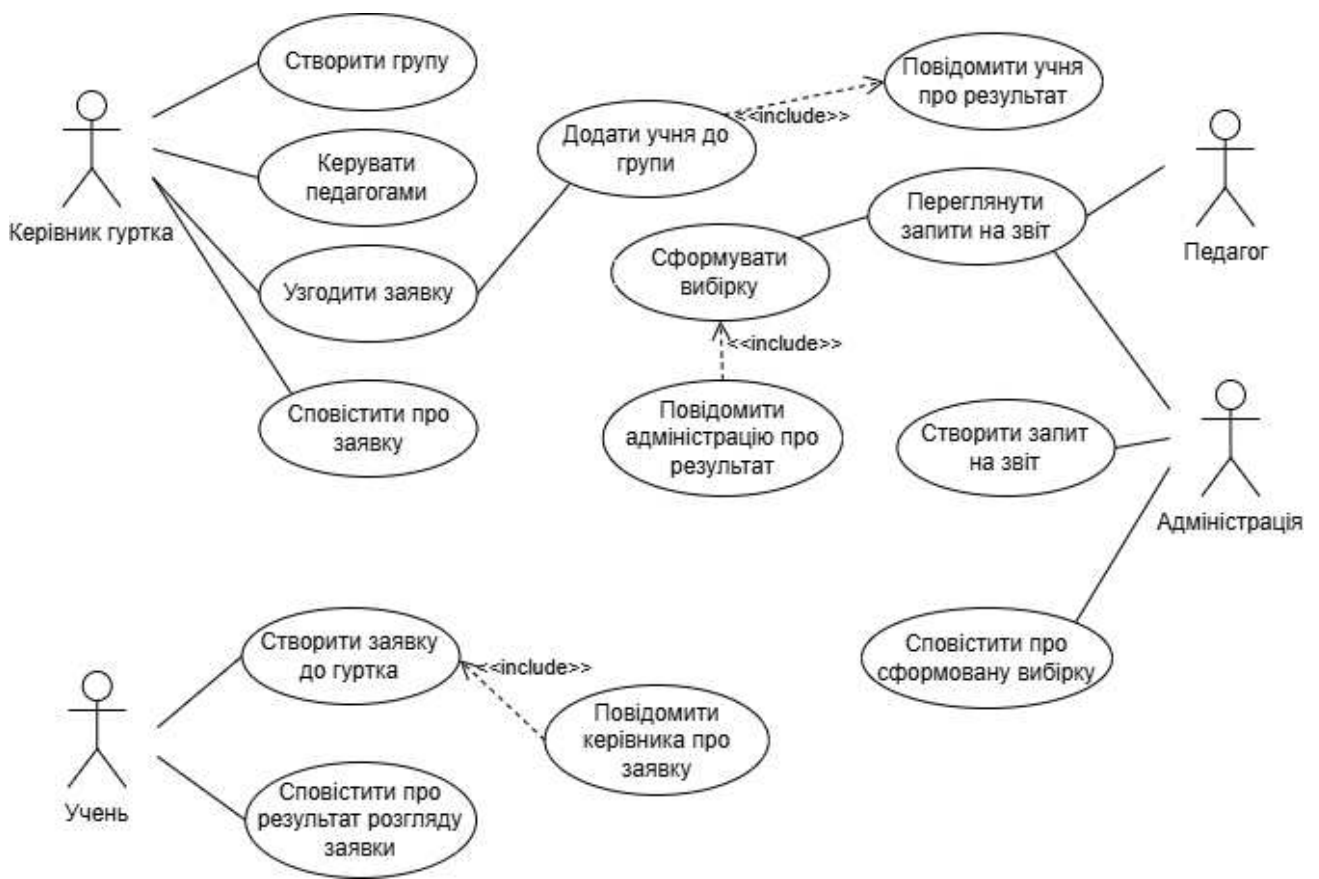


Рисунок 2.1 – Діаграма прецедентів

Можливі варіанти використання інформаційної системи приведені у таблицях 2.1-2.3:

Таблиця 2.1. Створення групи

Створення групи	
Призначення	Використовується для створення нової групи
Користувач	Керівник гуртка
Передумова	Необхідне створення нової групи для учнів
На що реагує	Натискання на кнопку «Створити групу».
Сценарій	Керівник гуртка натискає кнопку «Створити групу», заповнює інформацію про групу, натискає кнопку «Зберегти».
Що відбудеться	Інформація про нову групу буде збережена в базі даних.

Таблиця 2.2. Створення запиту на звіт

Створення запиту на звіт	
Призначення	Використовується для створення запиту на звіт
Користувач	Адміністрація
Передумова	Було отримано запит на звіт від навчального закладу, необхідно зібрати актуальні дані по гурткам.
На що реагує	Натискання на кнопку «Створити запит на звіт».
Сценарій	Адміністрація натискає кнопку «Створити запит на звіт», заповнює інформацію про звіт, призначає вчителя, натискає кнопку «Відправити».
Що відбудеться	Призначений вчитель отримує повідомлення про новий запит на звіт. Інформація про запит буде збережена в базі даних.

Таблиця 2.3. Узгодження заявки

Узгодження заявки	
Призначення	Використовується для узгодження отриманої заявки до гуртка
Користувач	Керівник гуртка
Передумова	Учнем було створено заявку до гуртка, отримано повідомлення про нову заявку. Необхідна перевірка заявки та розподілення учня до групи.
На що реагує	Натискання на кнопку «Перевірити заявки».
Сценарій	Керівник гуртка натискає кнопку «Перевірити заявки», перевіряє дані введені в заявці, змінює статус заявки, за потребою призначає до групи і натискає на кнопку «Зберегти».
Що відбудеться	Результат розгляду заявки відправлено учню. Інформація про результат перевірки збережена в базі даних.

На діаграмі прецедентів зображено що є задачі, над якими користувачі працюють самостійно, а деякі виконуються декількома користувачами. Наприклад, для узгодження заявки керівником гуртка потребується завчасне створення заявки потенційним учнем, який вводить основні дані та відомості про себе, про що буде сповіщено керівника гуртка і він зможе перевірити заявку на відповідність учня до певних наявних вимог для зарахування у гурток, наявність місць в групах гуртка та приймає рішення щодо результату розгляду заявки. Учень повідомляється про результат розгляду, а за умови позитивного результату – учень додається до певної групи. Отже, завдяки такому опрацюванню заявок, будуть одразу формуватися групи в гуртках, учні будуть повідомлені про результат розгляду. Таким чином, це допоможе підвищити ефективність роботи керівника гуртка та спростить сповіщення кожного учня про його зарахування.

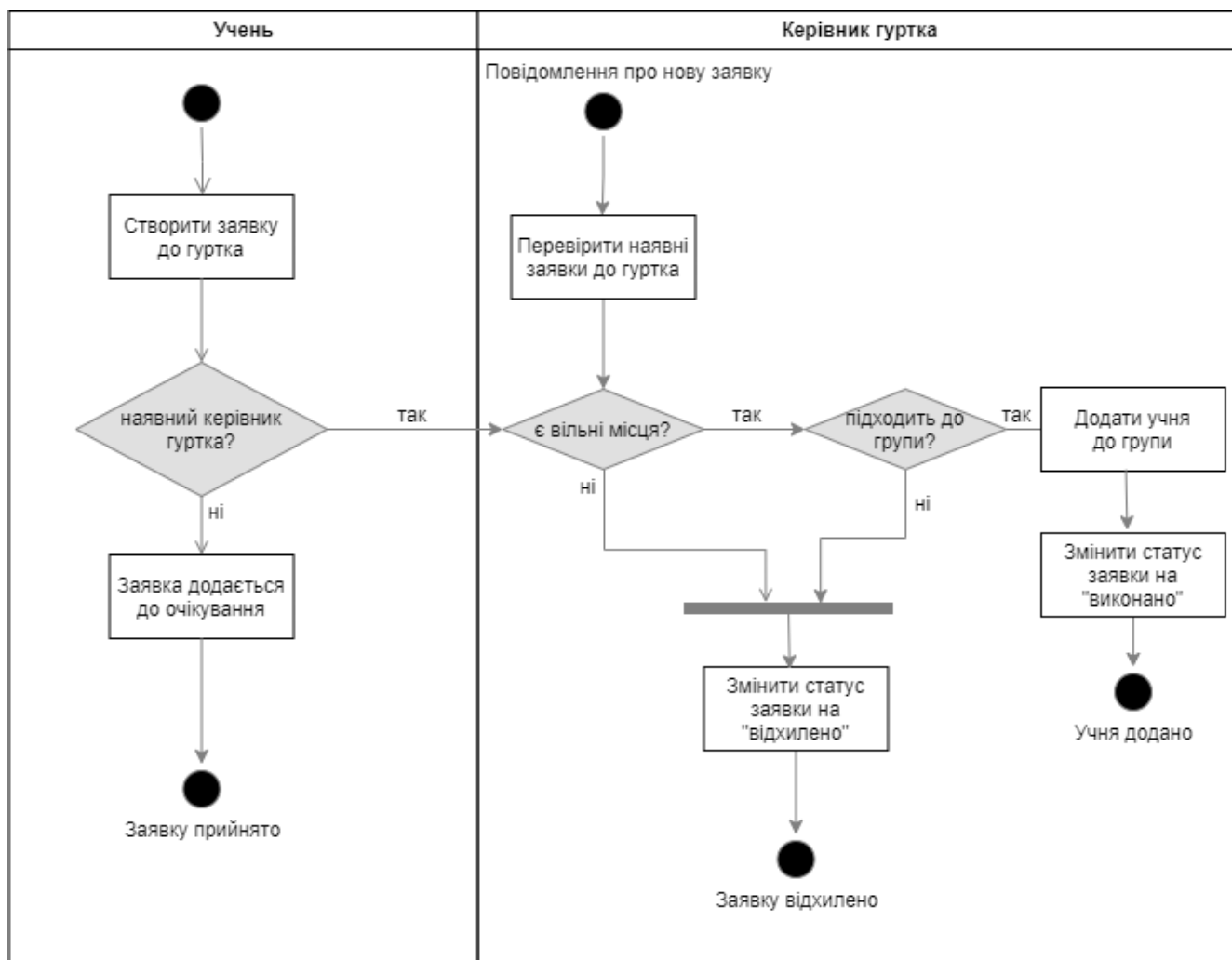


Рисунок 2.2 – Діаграма діяльності процесу обробки заявки до гуртка

Варіант колективної роботи з заявкою до гуртка зображено на діаграмі діяльності. На першому етапі учень створює заявку до гуртка заповнивши відповідну інформацію про себе. У випадку відсутності керівника гуртка до якого подається заявки – заявка додається до очікування та зберігається. Якщо керівник гуртка існує, то він отримує повідомлення про отриману заявку, переглядає наявні заявки до гуртка, після чого перевіряє наявність місць у групах, потім перевіряє дані введені учнем, базуючись на яких перевіряє чи підходить учень до будь-якої групи. Якщо заявка підходить після усіх перевірок – він додається до певної групи, статус заявки до гуртка змінюється на «Виконано», дані зберігаються. У випадку відсутності місць або у випадку що учень не підходить до групи за певними критеріями – керівник змінює

статус заявки на «Відхилено». Після розгляду заявки учень отримує повідомлення про результат.

Для ефективного виконання колективної роботи, до інформаційної системи висунуті наступні вимоги:

- 1) Простий та інтуїтивно зрозумілий користувацький інтерфейс;
- 2) Надійність інформаційної системи;
- 3) Розрахованість на роботу багатьма користувачами;
- 4) Універсальність інформаційної системи;
- 5) Простий початок користування інформаційною системою навчальним закладом;
- 6) Використання загальноприйнятих стандартів розробки.

Простий та інтуїтивно зрозумілий користувацький інтерфейс. Для інформаційної системи однією з важливих умов є простий та інтуїтивно зрозумілий користувацький інтерфейс, адже це дозволить будь-якому користувачу використовувати інформаційну систему без додаткової допомоги та інструкцій. Для забезпечення цієї вимоги інтерфейс має бути графічним. Графічний інтерфейс – тип інтерфейсу, який дає змогу користувачам взаємодіяти з інформаційною системою через графічні зображення та візуальні вказівки на дисплеї. Завдяки простому графічному інтерфейсу користувач буде розуміти які дії він має виконати та як працювати з інформаційною системою.

Надійність інформаційної системи. Надійність — властивість технічних об'єктів зберігати протягом встановленого часу значення всіх параметрів, які характеризують здатність виконувати потрібні функції в заданих режимах та умовах застосування, технічного обслуговування, зберігання та транспортування. Під технічними об'єктами розуміють пристрої, прилади, механізми, машини, комплекси обладнання, будівельні конструкції і споруди, технологічні операції і процеси, системи зв'язку, інформаційні системи, автоматизовані системи управління технологічними процесами тощо. [3]

Для надійності даних інформаційної системи на сервері потрібно розробити підсистему для реєстрації та надійної автентифікації користувачів, а при необхідності встановити антивірусне програмне забезпечення.

Розрахованість на роботу багатьма користувачами. Для можливості роботи декільком користувачам з системою, доступ до певних дій буде доступний лише користувачам які мають відповідні права доступу. Кожен користувач, в залежності від типу, матиме певні права до дій та запитів у базу даних.

Універсальність інформаційної системи. Універсальна система може використовуватись також іншими закладами зі схожою структурою для виконання певних дій. Наприклад, частина системи з генерацією вибірок може бути в нагоді для закладів шкільної освіти, частина з заявками до гуртка може використовуватися приватними компаніями що проводять набір людей на курси та тренінги.

Простий початок користування інформаційною системою навчальним закладом. При впровадженні інформаційної, вона повинна легко і без проблем запускатися в навчальному закладі.

Використання загальноприйнятих стандартів розробки. Завдяки використанню загальноприйнятих стандартів, інша людина, яка буде займатися підтримкою проекту, зможе без витрачання часу виконувати подальші роботи по розширенню, модернізації системи та виправленню несправності.

2.2 Рішення з інформаційного забезпечення

Концептуальне проектування — побудова семантичної моделі предметної області, тобто інформаційної моделі найбільш високого рівня абстракції. Така модель створюється без орієнтації на якусь конкретну СУБД і модель даних. [4]. Базується на аналізі предметної області, вимог до інформаційної системи та потреб замовника, побудову концептуальної моделі

даних та відтворення взаємодіючих об'єктів. Об'єктами концептуальної моделі даних є відповідні реальні об'єкти в даній області.

Усі об'єкти предметної області відзначаються певними наборами атрибутів, які відображають властивості відповідного об'єкта.

Необхідні атрибути об'єкта обираються для визначення того, які дані повинні відображати об'єкт. Атрибути, за якими можливо однозначно визначити об'єкт називаються ідентифікатором об'єкта.

У предметній області можливо виокремити такі сутності: вчитель, учень, група, гурток, статус, заявка до групи, запит на звіт.

Отже, в наслідок концептуального проектування розроблено модель бази даних інформаційної системи для позашкільного навчального закладу у вигляді ER-діаграми, зображено на рисунку 2.3.

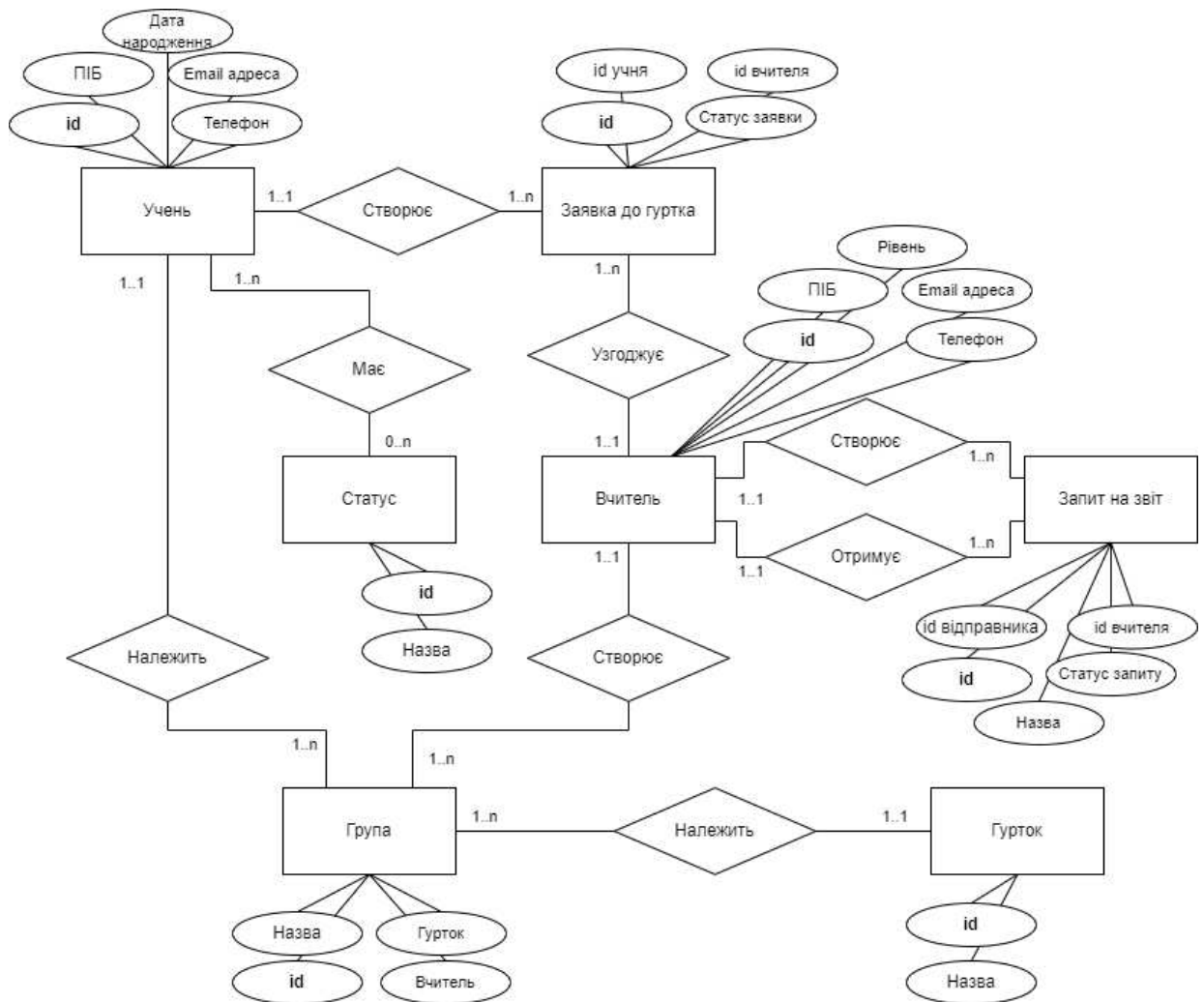


Рисунок 2.3 – Концептуальна ER-діаграма

Під час проектування закладається основа потенційної бази даних, визначається необхідна структура бази даних, спираючись на обрану предметну область.

Визначена логічна структура бази даних відповідає реляційним таблицям, їх склад та логічні відносини між ними. Під час формування структури певної таблиці визначаються поля таблиці, які відповідають атрибутам певного об'єкту, додається опис, тип даних, обмеження та інші властивості. Також повинно бути визначено унікальний ключ таблиці, який може складати з себе одне або декілька полів, та визначати окремий запис в таблиці.

Фізична структура бази даних – структура створеної бази даних для комп'ютера, яка визначає тип, властивості та обмеження даних, які будуть записані в пам'ять комп'ютера.

Фізичне моделювання бази даних являє собою визначення складу файлів і їх заповнення вихідними даними, відповідно до предметної області та обмежень даних. Проектування бази даних виглядає у створенні таблиць в СУБД. Також при фізичному проектування бази даних необхідно визначити імена таблиць та імена полям таблиць.

Фізична структура бази даних інформаційної системи обліку учнів позашкільних навчальних закладів відображена на рисунку 2.4.

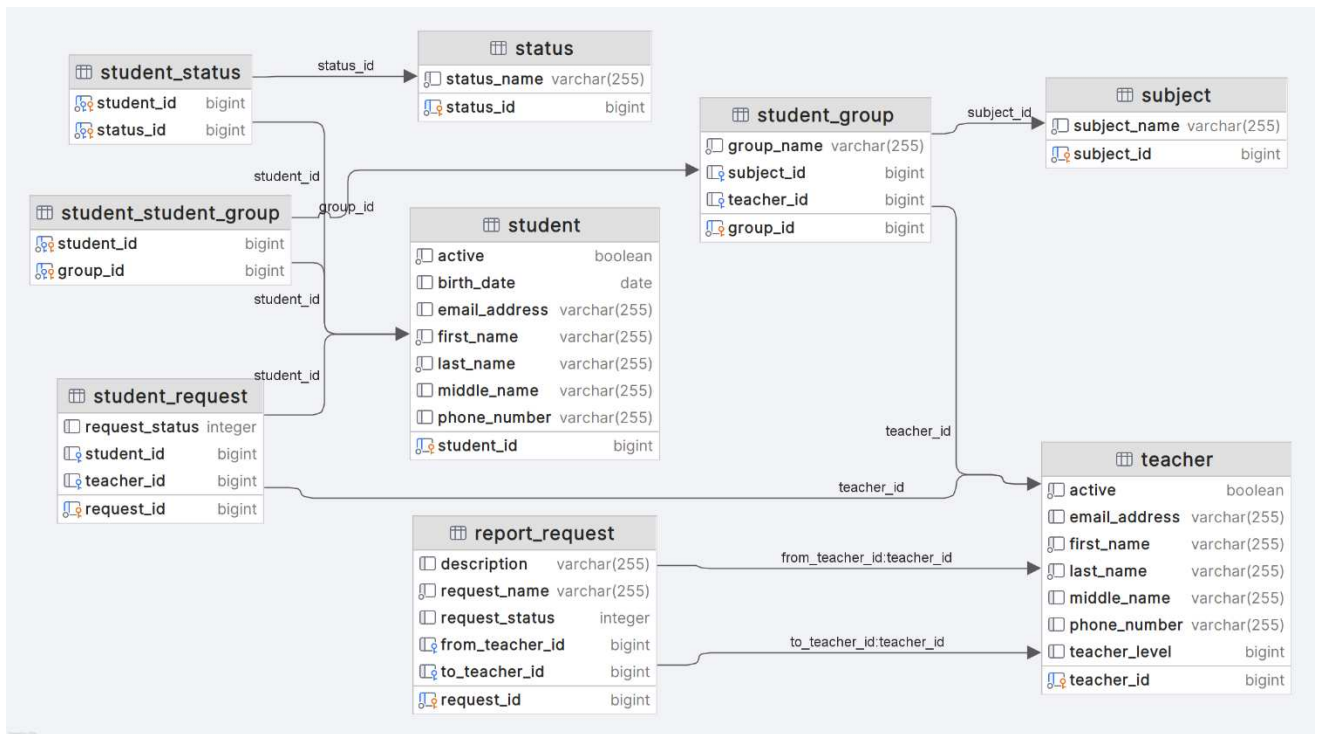


Рисунок 2.4 – Фізична модель бази даних

Фізичною моделлю створено структуру бази даних, визначено об'єкти системи, відношення між ними, встановлено характеристики та властивості об'єктів, створені поля, визначено первинні ключі.

Використовуючи рядки таблиць бази даних, сутності представляються у вигляді об'єктів класів в програмах серверної, а також клієнтської частин. Відношення та зв'язки між класами інформаційної системи позашкільного навчального закладу зображено на діаграмі класів, рисунок 2.5.

Код створення таблиць міститься у додатку Г.

Веб-додаток інформаційної системи було розроблено мовою Java. Java — це широко використовувана об'єктно-орієнтована мова програмування та програмна платформа, яка працює на мільярдах пристроїв, включаючи ноутбуки, мобільні пристрої, ігрові консолі, медичні пристрої та багато інших. Однією з основних переваг розробки програмного забезпечення на Java є її кросплатформеність.

Для створення програми за допомогою Java, потрібно завантажити Java Development Kit (JDK). Розробляється програма мовою програмування Java, потім компілятор перетворює програму в байт-код Java — набір інструкцій для віртуальної машини Java (JVM), яка є частиною середовища виконання Java (JRE). Байт-код Java працює без змін у будь-якій системі, яка підтримує JVM, що дозволяє запускати Java-код всюди.

Для розробки використовувалось інтегроване середовище розробки (IDE) IntelliJ IDEA Ultimate Edition. Основними перевагами IntelliJ IDEA є широкий вибір інструментів, які воно надає, щоб допомогти в розробці програмного забезпечення, а також ефективність і прискорення процесу розробки.

IntelliJ IDEA містить велику кількість інтегрованих інструментів, які дозволяють розробляти програмне забезпечення ефективніше та продуктивніше. Одним з них є високорозвинений редактор, який може надати користувачам інструмент, який допомагає під час кодування.

Також для розробки було використано групу фреймворків Spring. Spring Framework (Spring) — це програма з відкритим вихідним кодом, яка забезпечує підтримку інфраструктури для розробки додатків Java. Один з найпопулярніших фреймворків Java Enterprise Edition (Java EE), Spring допомагає розробникам створювати високопродуктивні програми за допомогою простих об'єктів Java (POJO). Spring вважається надійним, недорогим і гнучким каркасом. Spring покращує ефективність кодування та скорочує загальний час розробки додатків, оскільки він легкий, ефективний у

використанні системних ресурсів і має велику підтримку [7]. Для швидкої конфігурації було використано один з проектів Spring – Spring Boot. Spring Boot — це веб-фреймворк Java з відкритим вихідним кодом на основі мікросервісів. Фреймворк Spring Boot створює повністю готове до виробництва середовище, яке можна повністю налаштувати за допомогою попередньо створеного коду в його базі коду. Архітектура мікросервісів надає розробнику повністю закритий додаток, включаючи вбудовані сервери додатків.

Для взаємодії з сутностями бази даних було використано Spring Data JPA. Spring Data JPA, частина більшого сімейства Spring Data, полегшує впровадження репозиторіїв на основі JPA. Цей модуль стосується розширеної підтримки рівнів доступу до даних на основі JPA. Це спрощує створення додатків на основі Spring, які використовують технології доступу до даних.

Для автентифікації користувачів використовується Spring Security. Spring Security — це фреймворк, який зосереджений на забезпеченні механізмів автентифікації та авторизації для додатків Spring.

Для документації API було використано інструмент Swagger. Swagger дозволяє описати структуру API, щоб машини могли їх читати. Здатність API описувати власну структуру є коренем усієї приголомшливості Swagger. Читаючи структуру API, автоматично створюється інтерактивна документація API. Також можливе автоматичне створення клієнтської бібліотеки для API багатьма мовами та досліджування інших можливостей, наприклад автоматичне тестування. [8]

Для створення графічного інтерфейсу користувача використано Vue.js. Vue.js — JavaScript-фреймворк що використовує шаблон MVVM для створення інтерфейсів користувача на основі моделей даних, через реактивне зв'язування даних. Для простого створення графічного інтерфейсу було використано Vuefy, бібліотеку компонентів інтерфейсу користувача, створену на основі Vue.js і Bulma; два свіжі, але добре відомі фреймворки. Vuefy має два основних принципи: залишати речі простими та бути легкими.

Розроблено 12 модулів інформаційної системи та їх призначення, що зображено в таблиці 2.4.

№	Назва модуля	Призначення модуля
1	Додавання вчителя	Форма вводу даних про вчителя закладу.
2	Редагування даних вчителя	Форма для перегляду даних про вчителя з можливістю редагувати певні поля.
3	Створення групи	Форма вводу даних для створення нової групи.
4	Редагування групи	Форма перегляду групи з можливістю редагування. Містить перелік учнів, що належать до цієї групи.
5	Створення заявки до гуртка	Форма вводу даних про учня, з подальшим відправленням заявки до керівника гуртка..
6	Керування заявками до гуртка	Відображає весь список заявок до гуртка, з можливістю редагування даних про учня та розподілення учня до групи.
7	Створення запиту на створення звіту	Форма вводу даних про запит на звіт до вчителя.
8	Формування вибірки для звіту	Відображає список учнів з необхідної перспективи за певними критеріями.
9	Додавання учня	Форма вводу для даних про учня.
10	Редагування даних про учня	Форма для перегляду даних про учня з можливістю редагувати деякі поля
11	Створення нової категорій учня	Форма для створення нової категорії, до якої можуть відноситись учні.
12	Редагування категорії	Форма для перегляду даних про категорію з можливістю редагування.

Таблиця 2.4. Призначення програмних модулів інформаційної системи

Для прикладу, в додатку Г міститься програмний код модуля для роботи з групами, який містить в собі роботу з учнями та вчителями.

Методи розробки програмного забезпечення

При розробці інформаційної системи було вирішено використовувати інкрементальний метод розробки, так як навчальному закладу система потрібна до початку наступного навчального року і не має потреби запровадити систему якнайшвидше, отже є достатньо часу для визначення

вимог до інформаційної системи, розробки поступової стратегії розробки інформаційної системи. На початку розробляється поступова стратегія розробки, відповідний графік впровадження нового функціоналу системи. Інкрементальна модель забезпечує побудову спочатку невеликої частини системи з її подальшим розширенням у декілька етапів.

Інформаційною системою забезпечується зберігання та обробка структурованих даних у вигляді об'єктів, отже у якості методології програмування є об'єктно-орієнтоване програмування (ООП). Така методологія у майбутньому полегшить процес розробки, дозволяє структурувати розробку за модулями, а також полегшить підтримку проекту та його розширення.

Операційна система

У якості операційної системи для комп'ютерів позашкільного навчального закладу використовується ОС Linux, дистрибутив Mint.

Лінукс (англ. Linux, повна назва — GNU/Linux) — узагальнена назва операційних систем, розроблених на базі ядра Linux і належать до класу UNIX-подібних систем. Ці операційні системи є видатним прикладом вільного програмного забезпечення та програмного забезпечення з відкритим кодом. Відмінною особливістю Лінукс є відкритість вихідного коду, що надає користувачам можливість вільно використовувати, модифікувати та поширювати програмне забезпечення, на відміну від власницьких операційних систем, таких як Microsoft Windows або MacOS. [9]

Дистрибутив – набір файлів, що підготовлені до установки. Набір файлів містить в собі всі відповідні компоненти, інсталятор та інший додатковий контент.

Перевагами Linux Mint є:

- легко налаштовується;
- досить легкий для початківців;
- включає в себе все з Linux;

- найкращий менеджер програмного забезпечення;
- попередньо встановлені програми.

Типові програмні засоби

Інформаційна система функціонує з використанням веб-додатка, доступ до якого здійснюється через веб-браузери, такі як Google Chrome. Google Chrome представляє сучасний браузер, розроблений компанією Google, що характеризується високою стабільністю, безпекою, швидкодією, широким набором доступних розширень та зручним інтерфейсом користувача.

Для створення діаграм застосовується графічний редактор diagrams.net, що є програмним забезпеченням з відкритим вихідним кодом, орієнтованим на створення діаграм. Це рішення отримало значне поширення серед кінцевих користувачів завдяки доступності через веб-браузери. Для генерації фізичної моделі бази даних використовується функціонал, інтегрований у середовище розробки IntelliJ IDEA.

Документування API реалізується із застосуванням технології Swagger. Документування коду здійснюється за допомогою генератора Javadoc, який створює документацію у форматі HTML на основі коментарів до вихідного коду мовою програмування Java. Javadoc визнано стандартом для документування класів Java, і більшість інтегрованих середовищ розробки програмного забезпечення забезпечують автоматичне створення HTML-документації на основі цього інструмента. [10]

Сервісне програмне забезпечення

Інтернет з'єднання в позашкільному навчальному закладу відбувається за допомогою Wi-Fi. Wi-Fi - торгова марка Wi-Fi Alliance та загальноновживана назва для стандарту IEEE 802.11 передавання цифрових потоків даних по радіоканалах. Обладнання, що відповідає стандарту IEEE 802.11, може бути протестоване Wi-Fi Alliance та отримати відповідний сертифікат і право нанесення логотипу Wi-Fi. Поширеним на сьогодні є протокол IEEE 802.11n.

Встановлення Wireless LAN доцільне для побудови мереж, де розгортання кабельної системи є неможливим або економічно недоцільним. Поточні реалізації Wi-Fi дозволяють отримати швидкість передавання даних понад 100 Мбіт/с, при цьому користувачі можуть переміщуватися між точками доступу на території покриття мережі Wi-Fi, використовуючи мобільні пристрої (КПК, смартфони, PSP і ноутбуки), оснащені клієнтськими приймально-передавальними пристроями Wi-Fi та отримувати доступ в Інтернет. [11]

Для відправлення сповіщень та повідомлень про нові запити, між співробітниками навчального закладу використовується електронна пошта. У якості поштового сервісу заклади зазвичай використовують Gmail – зручний і простий поштовий сервіс від компанії Google, що сумісний з іншими Google сервісами, надає безкоштовний простір у хмарному сховищі.

Стратегія резервного копіювання вимагає інформаційного сховища, «вторинного простору для зберігання даних», який агрегує резервні копії «джерел» даних. Репозиторій може бути таким простим, будь-яким зі списку усіх носіїв резервних копій, або може включати комп'ютеризований індекс, каталог або реляційну базу даних.

Дані резервної копії необхідно зберігати, що вимагає схеми чергування резервних копій, яка є системою резервного копіювання даних на комп'ютерних носіях, яка обмежує кількість резервних копій різних дат, що зберігаються окремо, шляхом відповідного повторного використання носія для зберігання даних та шляхом перезапису резервних копій, які більше не потрібні. Схема визначає, як і коли кожен фрагмент певного сховища використовується для операції резервного копіювання і як довго він зберігається, коли на ньому зберігаються дані резервної копії. [12]

- Існує декілька видів резервного копіювання:
- Повне резервне копіювання;

- Диференційне резервне копіювання;
- Інкрементне резервне копіювання
- Клонування;
- Резервне копіювання як образ;
- Резервне копіювання в режимі реального часу;
- Холодне резервування;
- Гаряче резервування.

Сховищами резервної копії можуть бути:

- стрічкова бібліотека або стример — запис резервних копій відбувається на магнітну стрічку стримера;
- дискове сховище — запис резервних копій відбувається на диски, що можуть бути об'єднані в RAID або на дискову систему збереження;
- Virtual Tape Library — запис резервних копій на диски, але саме сховище виглядає для клієнта як стрічкова бібліотека;
- «хмарний» бекап — запис резервних даних за «хмарною» технологією через онлайн-служби спеціальних провайдерів.

Для резервного копіювання бази даних інформаційної системи обліку учнів позашкільних навчальних закладів буде використано `pg_dump` — вбудована у PostgreSQL утиліта для спрощення резервного копіювання однієї бази даних. Цю команду потрібно запустити як користувач з дозволом на читання бази даних, яку ми маємо створити резервну копію.

Отже, розроблена інформаційна система відповідає вимогам, що закладені у систему, протестована, перевірена та захищена від втрати даних.

2.4 Рішення з технічного забезпечення

Популярними на даний час типами носіїв даних в інформаційній системах різних компаній та закладів є HDD та SSD. HDD (hard disk drive) – жорсткий магнітний диск, SSD (solid-state drive) – твердотільний накопичувач.

Порівнюючи такі носії, основними перевагами SSD є швидкість та надійність. HDD є більш старою технологією носіїв, основна перевага перед SSD – вони дешевші.

Що стосується ємності, SSD для комп'ютерів доступні ємністю від 120 ГБ до 30,72 ТБ, тоді як жорсткі диски можуть бути від 250 ГБ до 20 ТБ. При вимірюванні вартості ємності жорсткі диски виходять на перше місце, але в міру зниження ціни на SSD це стане менш відмінним для жорстких дисків. Однак із твердотільними накопичувачами ви отримаєте набагато більше роботи на сервер, що призводить до розгортання меншої кількості пристроїв для отримання того ж результату, що й жорсткий диск. Як результат, SSD мають нижчу TCO (загальна вартість володіння).

Надійність визначається як те, чи зберігаються дані, як задумано, у непошкодженому стані. SSD загалом надійніше жорстких дисків, що є функцією відсутності рухомих частин. Це тому, що без руху на SSD не впливає вібрація або пов'язані з цим проблеми з температурою.

SSD зазвичай споживають менше енергії і мають триваліший термін служби акумулятора, оскільки доступ до даних відбувається набагато швидше, а пристрій частіше простоює. Завдяки дискам, що обертаються, жорсткі диски вимагають більше енергії під час запуску, ніж SSD. [13]

NAND – це енергонезалежна флеш-пам'ять, яка може зберігати дані, навіть якщо вона не підключена до джерела живлення. Можливість зберігати дані, коли живлення вимкнено, робить NAND чудовим варіантом для внутрішніх, зовнішніх і портативних пристроїв. USB-накопичувачі, твердотільні накопичувачі та SD-карти використовують флеш-технологію,

забезпечуючи пам'ять для таких пристроїв, як мобільний телефон або цифрова камера.

На ринку є кілька типів NAND. Простіше кажучи, що відокремлює кожен тип, так це кількість бітів, які можна зберегти в комірці. Біти представляють електричний заряд, який може містити лише одне з двох значень, 0 або 1, увімкнено/вимкнено.

Основними відмінностями між типами NAND є вартість, ємність і витривалість. Витривалість визначається кількістю циклів програмування-стирання (P/E), які може пройти елемент флеш-пам'яті, перш ніж він почне зношуватися. Цикл P/E – це процес стирання та запису осередку, і чим більше циклів P/E може витримати технологія NAND, тим краще витривалість пристрою.

Поширеними типами флеш-сховища NAND є SLC, MLC, TLC і 3D NAND.

SLC NAND

Плюси: Найвища витривалість. Мінуси: Дорогі та невеликі ємності

Однорівневий осередок (SLC) NAND зберігає лише 1 біт інформації на клітинку. У клітинці зберігається або 0, або 1, і в результаті дані можуть бути записані та отримані швидше. SLC забезпечує найкращу продуктивність і найвищу витривалість із 100 000 циклів P/E, тому він прослужить довше, ніж інші типи NAND. Однак його низька щільність даних робить SLC найдорожчим типом NAND.

MLC NAND

Плюси: дешевше, ніж SLC. Мінуси: повільніше і менш витривалий, ніж SLC

NAND багаторівневої комірки (MLC) зберігає кілька біт на клітинку, хоча під терміном MLC зазвичай мають на увазі, що кількість дорівнює 2 бітам на клітинку. MLC має вищу щільність даних, ніж SLC, тому його можна виробляти на більші ємності. MLC має гарне поєднання ціни, продуктивності та витривалості. Однак MLC більш чутливий до помилок даних із 10 000

циклів P/E і, отже, має меншу витривалість порівняно з SLC. MLC зазвичай зустрічається в споживчих продуктах, де витривалість менш важлива.

TLC NAND

Плюси: Найдешевша і велика ємність - Мінуси: Низька витривалість

Трирівневий осередок (TLC) NAND зберігає 3 біти на клітинку. Додаючи більше бітів на клітинку, це зменшує вартість і збільшує ємність. Однак це негативно позначається на продуктивності та витривалості, оскільки має лише 3000 циклів P/E. Багато споживчих товарів використовують саме TLC, оскільки це найдешевший варіант.

3D NAND

За останні десять років 3D NAND була однією з найбільших інновацій на ринку флеш-пам'яті. Виробники флеш-пам'яті розробили 3D NAND, щоб виправити проблеми, з якими вони стикалися, зменшивши масштаб 2D NAND, щоб досягти більшої щільності за меншою ціною. У 2D NAND клітинки, які зберігають дані, розміщені горизонтально, поруч. Це означає, що кількість простору, на якому можна розмістити клітини, обмежена, а спроба зменшити клітини зменшує їх надійність. [14]

Отже, для сервера інформаційної системи було обрано SSD TLC, так як вони є досить швидкими та для зменшення вартості впровадження. Для бекапів бази даних обрано HDD диски, оскільки вони дешевші і обсяги даних можуть бути великими. На клієнтських комп'ютерах достатньо мати жорсткі диски, оскільки вони не матимуть зберігати велику кількість даних, всі дані зберігаються на стороні серверу.

На основі системних вимог було обрано таке технічне забезпечення:

- 1) Комп'ютер для клієнта: Intel Core i3-10100 3.6 GHz, 4 GB DDR4, 120 GB HDD;
- 2) Комп'ютер для сервера: Intel Xeon Silver 4219 3.2 GHz, 16 GB DDR4, SSD 120 GB, HDD 500 GB;
- 3) Джерело безперебійного живлення, для серверу: LogicPower LPY-W-PSW-2000VA+;

- 4) Монітор: Lenovo ThinkVision E24-28;
- 5) Комплект клавіатура і мишка: Logitech Desktop MK120

Висновок

Вибір технічного забезпечення та підхід до його впровадження дозволили створити ефективну інформаційну систему, яка відповідає встановленим вимогам щодо обліку даних учнів позашкільних навчальних закладів. Використання серверів із SSD для швидкої обробки даних і HDD для довгострокового зберігання у поєднанні з оптимально підібраними клієнтськими комп'ютерами забезпечує плавну роботу системи. Інтеграція інтуїтивно зрозумілого інтерфейсу дозволяє спростити використання програмного забезпечення та забезпечує зручне управління даними, що робить систему доступною для користувачів з різним рівнем підготовки.

РОЗДІЛ 3. РЕАЛІЗАЦІЯ ВЕБЗАСТОСУНКУ ІНФОРМАЦІЙНОЇ СИСТЕМИ ПОЗАШКІЛЬНОГО НАВЧАЛЬНОГО ЗАКЛАДУ

В інформаційній системі реалізовано декілька типових модулів для керування різними частинами, наприклад, сторінка керівництва з можливістю переглядати дані про усі структури навчального закладу та вносити зміни до даних про студентів, групи, вчителів та інші. Приклади екранних форм зображено у додатку В.

Інструкція користувача

В інструкції описуються основні дії користувача. Область використання інформаційної системи обліку учнів – позашкільні навчальні заклади. Система автоматизує ведення даних про учнів, групи, створення заявок до гуртків, запити на звіти.

Завдяки інтуїтивно зрозумілому інтерфейсу програмного забезпечення, користувач з будь-яким рівнем підготовки зможе легко почати користуватися інформаційною системою. Модулі та функції на сторінках проекту розташовані структуровано, відповідно до розділів. Всі дані зображено у вигляді таблиць, натискання на будь-який рядок відобразить більш детальну інформацію, можливість редагування, збереження та видалення даних. Усі елементи, що пов'язані зі збереженням інформації або її додаванням мають зелений колір, а пов'язані з видаленням – червоний. Також деякі таблиці мають поля для фільтрування. Приклади елементів графічного інтерфейсу, через які виконується управління інформаційною системою представлено у таблиці 3.1.

Таблиця 3.1. Елементи графічного інтерфейсу користувача

№	Зовнішній вигляд	Опис
1		Створення нового запису

2	Прізвище	Форма пошуку записів за певними критеріями
3	1 Ігор Іванов	Рядок, натиснувши на який відкриється детальніша інформація
4	 Обрати дату...	Поле вибору дати
5	 Видалити	Кнопка видалення запису
6	 Зберегти	Зберегти зміни

Інструкція по формуванню і веденню бази даних

Для підготовки даних спочатку проводиться збір паперових документів, що містять початкові, вже існуючі дані для введення. База даних для інформаційної системи створюється, налаштовується та керується за допомогою графічного клієнту для управління базою даних PostgreSQL - pgAdmin4. Співробітники навчального закладу та керівництво додають дані в систему за допомогою програмного забезпечення, а саме веб-застосунку. Після вводу, дані зберігаються до спільної бази даних, яка використовується для всіх автоматизованих робочих місць та зберігає унікальні та актуальні дані. Редагування та видалення даних можуть виконувати тільки користувачі з відповідним рівнем доступу. Системний адміністратор інформаційної системи та бази даних стежить за роботою бази даних, наявністю вільного місця, стежить за створенням бекапів. Резервне копіювання повинне відбуватися постійно: щоденно, щотижнево, щомісячно та щорічно.

Копіювання, що створюється щоденно, повинно зберігатися тиждень. Щотижневое копіювання зберігатиметься місяць, щомісячні бекапи повинні зберігатися протягом двох років. Щорічні повинні зберігатися постійно.

Підготовка до роботи інформаційної системи

Для початку роботи системи, системний адміністратор повинен ознайомитися з інструкцією по формуванню і веденню бази даних. Після заповнення бази даних та налаштування резервного копіювання, співробітники навчального закладу можуть почати працювати з інформаційною системою використовуючи веб-додаток. Також, для цього потрібно створити облікові записи користувачів, та налаштувати їх ролі, відповідно до яких вони матимуть дозвіл на виконання відповідних дій. Налаштування комп'ютерів та реєстрацію користувачів здійснює системний адміністратор.

Інструкція щодо використання системи

Починаючи роботу, працівник повинен увійти до свого облікового запису, використовуючи свій логін та пароль. Якщо на комп'ютері встановлено пароль безпеки, відходячи від комп'ютера рекомендується блокувати його. Якщо пароль не встановлено, рекомендується виходити з акаунту користувача. Це потрібно для забезпечення безпеки даних, що містить в собі інформаційна система. Після входу, користувач може здійснювати необхідні доступні для нього дії.

В інформаційній системі запроваджено логування (запис дій користувачів), що дозволить швидше знаходити та виправляти недоліки.

Сервер і комп'ютери користувачів повинні мати безперебійне живлення, всі комп'ютери мають бути під'єднані до стабільного Інтернету. Доступ до системи повинні мати тільки співробітники навчального закладу, в системі повинні зберігатися актуальні та коректні дані.

Інструкція на випадок аварійних ситуацій

У випадку тривалої, або критично некоректної роботи програмного забезпечення або технічних засобів, необхідно повідомити про проблему системному адміністратору, який в залежності від помилки, визначить

проблему. Якщо проблема у технічній частині – слід виконати ремонт, віднести на сервісне обслуговування або купити нову. У випадку, якщо проблема відбулась на стороні серверу, у програмному забезпеченні, необхідно звернутися до розробника, що відповідає за підтримку відповідного програмного забезпечення. Розробник знайде спосіб рішення цієї помилки та надає виправлений код або інше рішення. У випадку небажаного втручання до інформаційної системи, порушення цілісності бази даних, втрату даних, системний адміністратор повинен відновити дані в базі даних за допомогою одного з останніх бекапів, що зберігаються. Під час проведення відновлення даних, користувачам забороняється користуватися інформаційною системою, про що повинен повідомити системний адміністратор.

Висновок

Завдяки реалізації даного проекту, підвищується ефективність роботи співробітників позашкільного навчального закладу, а також зменшується час, що витрачається на типову роботу, за рахунок раціонального використання часу та трудових ресурсів закладу.

РОЗДІЛ 4. ОХОРОНА ПРАЦІ ТА ТЕХНІКА БЕЗПЕКИ

Вступ

Охорона праці — це комплексна система забезпечення безпечних умов для життя та здоров'я працівників у процесі здійснення ними трудової діяльності. Її структура охоплює правові, соціально-економічні, організаційні, технічні, гігієнічні, а також лікувально-профілактичні заходи, що спрямовані на мінімізацію ризиків виникнення травм та професійних захворювань.

Основоположною метою охорони праці є створення умов, які гарантують максимально ефективний захист від негативного впливу виробничих факторів, здатних спричинити шкоду життю чи здоров'ю працівників. Для реалізації цієї мети впроваджуються науково обґрунтовані процеси та технології, що відповідають сучасним стандартам безпеки та сприяють оптимізації трудової діяльності на рівні конкретного підприємства, установи чи організації.

Важливим аспектом охорони праці є моніторинг та оцінювання потенційних ризиків, пов'язаних із виконанням професійних обов'язків у різних галузях. Це включає аналіз умов праці, оцінку небезпечних і шкідливих факторів виробничого середовища, а також розробку заходів щодо їх усунення або мінімізації. До таких заходів належать вдосконалення технічних засобів, забезпечення належного мікроклімату і освітлення робочих приміщень, впровадження новітніх технологій виробництва, а також навчання персоналу правилам безпеки.

Крім того, охорона праці передбачає підготовку та впровадження профілактичних програм, спрямованих на збереження фізичного, психологічного та соціального здоров'я працівників. Це включає регулярні медичні огляди, психологічну підтримку, а також заходи, що сприяють підвищенню рівня обізнаності працівників у сфері безпеки праці.

Таким чином, охорона праці є невід'ємною складовою організації трудового процесу, яка комплексно забезпечує безпеку, зменшує ризики та сприяє підвищенню ефективності виробничої діяльності.

Під час роботи працівник взаємодіє з виробничим середовищем, предметами праці та засобами виробництва. У процесі він зіштовхується з різними факторами, що впливають на його здоров'я та здатність виконувати обов'язки. Виробничі фактори бувають небезпечними або шкідливими.

Небезпечний фактор може спричинити травму або погіршення здоров'я. Шкідливий фактор призводить до захворювань або зниження працездатності. Шкідливі фактори можуть стати небезпечними залежно від сили та тривалості їхнього впливу. За типом впливу вони бувають фізичними, хімічними, біологічними та психофізіологічними.

Фізичні фактори включають несприятливий мікроклімат, електромагнітні поля, високий рівень шуму та вібрацій, недостатнє освітлення, барометричний тиск і іонізуюче випромінювання.

Хімічні фактори – це органічні та неорганічні речовини, з якими контактують працівники.

Біологічні фактори представляють собою мікроби, грибки, а також різні продукти мікробіологічного синтезу.

Психофізіологічні фактори включають емоційну напругу через надмірне навантаження на нервову систему, динамічні та статичні перевантаження, тривалий тиск предметів на частини тіла, швидкий темп роботи та недостатню рухову активність.

Техніка безпеки - це система засобів і правил, що забезпечує безпечні умови праці. Виконання цих вимог є одним із важливих заходів держави щодо охорони праці. Кожна галузь має свої правила з техніки безпеки, розроблені науковими установами та затверджені профспілками.

Керівництво підприємств повинно дотримуватися цих правил. Порушення правил техніки безпеки тягне за собою адміністративну та кримінальну відповідальність.

Законодавчо, охорона праці регулюється Конституцією України та Законом про охорону праці. У статті 43 Конституції України вказано: «Кожен має право на належні, безпечні та здорові умови праці, на заробітну плату, не нижчу від визначеної законом».

Закон «Про охорону праці» визначає основні положення щодо реалізації конституційного права працівників на охорону їхнього життя та здоров'я під час виконання трудової діяльності, на належні, безпечні та здорові умови праці, за участю відповідних органів державної влади регулює відносини між роботодавцем і працівником з питань безпеки, гігієни праці та виробничого середовища а також встановлює єдиний порядок організації охорони праці в Україні. [15].

Будь-яке підприємство має прагнути забезпечити комфортні умови для роботи працівників для створення сприятливих умов праці. Перш за все, це стосується чистоти повітря у виробничих приміщеннях. Основні шкідливі виробничі фактори поділяються на хімічні речовини та промисловий пил. У свою чергу, хімічні речовини поділяються на:

- Промислові отрути, що використовуються на виробництві;
- Сільськогосподарські отрутохімікати;
- Побутова хімія;
- Препарати лікувального призначення;
- Отруйні речовини.

Великі обсяги промислового пилу також є проблемою деяких виробництв та підприємств. Негативний вплив пилу в тому, що він здатен ініціювати розвиток захворювань легень. Отже, забезпечення захисту від негативної дії шкідливих факторів є дуже актуальним у виробничій сфері.

Незважаючи на всі існуючі заходи, спрямовані на позбавлення від шкідливого впливу факторів, досягти ідеальних умов праці є неможливим. Це неможливе через певні особливості технологічних процесів, продукцію та сировину для її виготовлення. Отже, для керівництва підприємств та закладів захист своїх співробітників від шкідливих виробничих факторів є

першочерговим завданням. Керівники повинні слідувати наступним пріоритетам:

- Усунути небезпечний фактор або знизити ризик його впливу;
- Використовувати максимально безпечні методи роботи;
- Здійснювати можливу боротьбу з небезпечними факторами та їх джерелами;
- За необхідністю забезпечити ефективне використання засобів індивідуального захисту.

Згідно зі статистики Управління Держпраці – за весь час незалежності, в Україні травмувалися близько 1 млн. людей, з яких загинули майже 35 тисяч. Отже, можна зі упевненістю сказати, що питання охорони праці є постійно актуальним, особливо якщо взяти до уваги що постійно на підприємствах запроваджуються нові технології, які можуть впливати на людей.

Викладачі та інші особи, які працюють в закладах освіти, мають ознайомлюватися з правилами та стандартами охорони праці. Законодавством передбачено проведення обов’язкових навчальних інструктажів та курсів для таких осіб. Інструктажі та курси спрямовані на підвищення рівня обізнаності робітників в галузі безпеки праці.

Існуючі види інструктажів з охорони праці зображено на таблиці 4.1.

Таблиця 4.1 Види інструктажів для співробітників навчального закладу.

Вид інструктажу	Умови проведення інструктажу
Вступний	Проводиться для осіб, які вперше виходять на разову або штатну роботу в навчальному закладі, а також для осіб, які приймають участь в навчальному процесі або виконують інші доручення підприємства.

Первинний	Проводиться для осіб, прийнятих на роботу до навчального закладу, переведених на іншу посаду в тому самому закладі, функціональні обов'язки яких змінено, а також тих, які перебувають у закладі за програмою обміну досвіду або у відрядженні.
Повторний	Проводиться для окремого співробітника або для групи співробітників, службові обов'язки яких потребують постійного оновлення знань.
Позаплановий	Здійснюється у випадку набрання чинності новими регуляторними актами з охорони праці, при здійсненні модернізації робочого процесу, початком роботи з новим обладнанням, при поновленні на роботу після тривалої перерви.
Цільовий	Виконується з метою роз'яснення порядку боротьби зі стихійним лихом або для усунення наслідків аварій, а також для осіб, на яких складається наказ, розпорядження.

4.1 Аналіз небезпечних і шкідливих факторів в позашкільному навчальному закладі

Успішний аналіз і вивчення виробничого травматизму та захворювань притаманним певним професіям можливі тільки за умови детального вивчення реальних причин їх виникнення. Причини виробничого травматизму та професійної захворюваності прийнято поділяти на декілька груп: організаційні, технічні, санітарно-гігієнічні, економічні та психофізіологічні.

Виявлення та знаходження шляхів вирішення проблем, що пов'язані з забезпеченням безпечних умов праці людини – одна з найбільш важливих задач у розробці нових технологій та систем виробництва. Створенню безпечних та сприятливих умов для праці людини сприяє вивчення та виявлення можливих причин нещасних випадків на підприємстві, професійних захворювань, пожеж, аварій, а також розробка відповідних заходів і вимог, що направлені на усунення можливих проблем. На продуктивність і ефективність робітників більш за все впливають комфортні та безпечні умови праці.

Робота вчителів позашкільного навчального закладу та адміністрації закладу використовуючи інформаційну систему, пов'язана з комп'ютером, з цього впливає велика група шкідливих факторів, які можуть значно знижувати продуктивність їх праці. До таких факторів відносяться:

1. Підвищений рівень шуму, що погіршує умови праці, вплив на організм людини проявляється у зниженні уваги, гостроти зору, слуху та підвищенні кров'яного тиску. Тривалий та гучний шум може призвести до змін у нервовій та серцево-судинній системі.
2. Недостатня освітленість приміщення та робочого місця призводить до збільшення напруги на зір, зменшує увагу, а також збільшує ймовірність передчасної стомленості.
3. Надмірна освітленість робочого місця призводить до роздратування, білю в очах, а також осліплення.
4. Нераціонально організоване освітлення робочого місця, за напрямком та яскравістю. Правильний напрям світла прямо впливає

на сприятливі умови праці, запобігає травматизм і професійні захворювання під час трудової діяльності, підвищує продуктивність та ефективність роботи робітника.

5. Ураження електричним струмом може виникати через відсутність заземлення комп'ютерів або при неправильно виконаній прокладці кабелів. Поверхневий електростатичний потенціал не має перевищувати 500 В.
6. Хвороби шкіри можуть виникати через високий рівень інфрачервоного випромінювання на робочому місці.
7. Некоректна організація робочого місця, розташування стільців та столів, відстані між робочими місцями призводять до незручної пози працівника. Це може призвести до швидкої втомленості, болі у м'язах, суглобах, а також до порушень постави та зору.

При щоденній роботі людини за комп'ютером, за умови недотримання принципів ергономіки, вимог санітарії, а також відповідного режиму роботи, це може довести до «ергономічних» захворювань. Такий термін медики почали використовувати для нових захворювань, що у більшості пов'язані із впливом комп'ютерів на здоров'я тих, хто ними користується. Наприклад, за даними Всесвітньої організації охорони здоров'я, у користувачів комп'ютерів виявляються нові типи захворювань, до яких відносяться: синдром комп'ютерного стресу оператора, травми повторних навантажень і фотоепілептичні приступи.

Згідно з Державними санітарними правилами та нормами, виникають наступні вимоги:

- 1) Загальне освітлення повинно бути виконано у вигляді суцільних або переривчастих ліній світильників.
- 2) Яскравість великих поверхонь (вікна, світильники та інше), що знаходяться у полі зору, не повинна перевищувати 200 кд/кв. м.
- 3) Відеомонітори на електронно-променевих трубках можуть бути потенційними джерелами гігієнічно значимих рівнів

електромагнітних випромінювань в діапазоні частот 50 Гц-300 МГц і статичного електричного поля.

- 4) В кабінетах та класах учбових закладів, де навчання проводиться з застосуванням комп'ютерів, температура повітря повинна бути 19.5 +- 0.5 град. С, а відносна вологість повітря 60 +- 5%, швидкість руху повітря не більше 0.1 м/с.
- 5) В кабінетах та класах учбових закладів повинен бути забезпечений 3-кратний обмін повітря за 1 годину. Для охолодження та очищення повітря від пилу в кабінетах та класах можуть бути встановлені побутові кондиціонери, які мають позитивний гігієнічний висновок державної санітарно-гігієнічної експертизи ГСЕУ МОЗ.
- 6) Візуальні ергономічні параметри відеомоніторів повинні задовольняти вимогам даних ДСанПіН при проектній відстані спостереження від 400 до 800 мм та при зовнішній освітленості екрана до 250 Лк.

4.2 Розробка заходів щодо зменшення впливу виробничих факторів на працюючого в позашкільному навчальному закладі.

Робочим місцем є просторова зона, що обладнана необхідними засобами, використовується для здійснення трудової діяльності працівника або групи працівників, які сумісно займаються виконанням певної роботи. Робоче місце призначене для виконання частини технологічного або виробничого процесу та визначається на основі трудових нормативно-правових актів.

Основними заходами з попередження або усунення причин виробничого травматизму і професійної захворюваності визначають технічні та організаційні заходи.

До технічних заходів, що запобігають впливу на робітників небезпечних виробничих чинників, відносять розроблення та впровадження безпечного устаткування, автоматизацію та механізацію певних технологічних процесів,

використання автоматичних блокувальних засобів, а також запобіжних пристосувань, правильно розташування органів керування обладнанням, впровадження нових нешкідливих та безпечних технологічних процесів.

До організаційних заходів належить правильна організація роботи, навчання, організація планового ремонту обладнання, контролю та нагляду з охорони праці, впровадження безпечних методів та наукової організації праці, дотримання трудового законодавства, проведення оглядів, лекцій та агітації з питань охорони праці.

У якості систем підтримання мікроклімату можуть використовуватися опалення та системи кондиціювання, які мають бути встановлені так, щоб повітря від них не було направлено на співробітників закладу та учнів.

Робоче місце, а також обладнання, що входить до складу робочого місця мають відповідати фізичним, антропометричним і психологічним вимогам. Також робоче місце повинне мати достатню кількість робочого простору, що дозволить швидко і зручно виконувати необхідні рухи.

Сидяча робота за комп'ютером призведе до мінімального стомлення працівника. Все, що постійно необхідне для виконання роботи має бути розташовано в зоні досяжності особи, що працює за відповідним робочим місцем.

Велика увага у робочому місці приділяється до столу, він повинен відповідати наступним умовам:

- Висота стола обирається з урахуванням можливості комфортного сидіння, у зручній позі, при необхідності опираючись на підлокітники. Також, наразі популярність набувають столи з регулюванням висоти;
- Поверхня стола повинна мати властивості, що виключають появу відблисків та віддзеркалювання світла у поле зору робітника;
- Конструкцією стола має бути передбачено наявність ящиків для зберігання документації;

- Рекомендована висота поверхні столу – в межах 680-760 мм. Висота поверхні, на яку встановлюється клавіатура, повинна бути близько 650 мм.

Також увага приділяється до робочого крісла:

- Висота сидіння над рівнем підлоги має бути в межах 420-550 мм;
- Поверхня сидіння – м'якою;
- Передній край має бути закругленим;
- Кут нахилу спинки повинен мати можливість регулювання.

Екран монітора повинен мати положення відповідно до відстані зчитування (0.6-0.7 м), а також куту зчитування – який має бути на 20 градусів нижче горизонталі до центру екрана.

Відповідно до норм, робоче місце має бути розташоване на відстані не менш ніж 1.5 м від стіни з вікнами, а від інших стін на відстані 1 м. Також не рекомендується розміщувати робоче місце прямо напроти джерела світла.

Для того щоб зайняти робочу позу для роботи з комп'ютером потрібно виконати наступні дії: відрегулювати висоту робочого столу (за можливістю), крісла та підставки для ніг (за наявності). Найзручнішою робочою позою прийнято вважати таке положення, при якому ступні робітника розташовані прямо перпендикулярно до гомілок та горизонтально підлозі або підставці для ніг, стегна повинні бути розташованими горизонтально, верхня частина рук – вертикально. Вигин ліктя має знаходитись в межах 70-90°, зап'ястя мають бути зігнуті під кутом не більше ніж 20°, а нахил голови має бути приблизно 15-20°.

Вимогами безпеки перед початком роботи є:

- Яскравість екрана монітору – не менше 100 кд/1м²;
- Мінімальний розмір точки свічення має бути не більше 0.4 мм для монохромного монітора та не менше 0.6 мм для кольорового, контрастність зображення – не менше 0.8;
- При виявленні несправності, роботу заборонено розпочинати;

Вимоги безпеки під час роботи:

- Не дозволяються подразнюючі шуми;
- Під час роботи з комп'ютером слід сидіти відповідно до робочого місця.

Вимоги щодо безпеки завершення роботи:

- Зберегти усі файли комп'ютера, що знаходились в роботі;
- Вимкнути кондиціювання або опалення приміщення, освітлення;
- Прибрати робоче місце відповідно до правил догляду за обладнанням.

Висновок

Таким чином, охорона праці робітника позашкільного навчального закладу, здебільшого полягає у дотриманні вимог щодо проектування робочого місця (освітлення, вентиляція, мікроклімат, електробезпека, та інші технічні засоби), ергономічних вимог до робочих місць і вимог режиму роботи окремого працівника. Завдяки створенню сприятливих та комфортних умов праці, правильному проектуванню робочих місць у закладі, можливе полегшення праці, а також зменшення впливу шкідливих та небезпечних умов праці, що забезпечить продуктивнішу та ефективнішу роботу працівників.

ВИСНОВОК

У даній дипломній роботі розроблено вебзастосунок інформаційної системи позашкільного навчального закладу. При розробці було проведено аналіз предметної області позашкільного навчального закладу, інформаційні процеси та структуру закладу. Сформовано задачі, які дозволяють реалізувати модулі управління учнями, вчителями, групами, заявками до груп та запитами на звіт. На основі сформованих вимог розроблено концепцію інформаційної системи. Було розроблено технічний проект вебзастосунку інформаційної системи позашкільних навчальних закладів, а також визначено загальносистемні рішення, рішення з інформаційного, програмного та технічного забезпечення. Реалізовано проект інформаційної системи, створено документацію та коротку інструкцію користувача, а також інструкції з формування і ведення бази даних, початку роботи з інформаційною системою та у аварійних випадках, приведено приклади форми вводу даних та окремих модулів керування системою. Проведено аналіз небезпечних та шкідливих факторів в позашкільному навчальному закладі, а також розроблено заходи і засоби по зменшенню впливу небезпечних та шкідливих факторів та техніка безпеки на робочому місці під час виконання трудових обов'язків. Проведені випробування, за підсумками яких було помічено підвищення ефективності роботи та зменшення часу, що витрачається на типову роботу вчителями позашкільного навчального закладу, за рахунок більш раціонального використання часу та трудових, матеріальних ресурсів, а також автоматизації інформаційних процесів.

СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ

1. Заклад позашкільної освіти [Електронний ресурс]: Вікіпедія. Вільна енциклопедія:
https://uk.wikipedia.org/wiki/Заклад_позашкільної_освіти
2. Автоматизоване робоче місце [Електронний ресурс]: Вікіпедія. Вільна енциклопедія:
https://uk.wikipedia.org/wiki/Автоматизоване_робоче_місце
3. Надійність [Електронний ресурс]: Вікіпедія. Вільна енциклопедія:
<https://uk.wikipedia.org/wiki/Надійність>
4. Проектування бази даних [Електронний ресурс]: Вікіпедія. Вільна енциклопедія: https://uk.wikipedia.org/wiki/Проектування_бази_даних
5. PostgreSQL [Електронний ресурс]: <https://www.postgresql.org/about/>
6. Переваги PostgreSQL [Електронний ресурс]: Блог.
<https://fulcrum.rocks/blog/why-use-postgresql-database/>
7. Spring Framework [Електронний ресурс]: TechTarget:
<https://www.techtarget.com/searchapparchitecture/definition/Spring-Framework>
8. Swagger [Електронний ресурс]: Swagger:
<https://swagger.io/docs/specification/2-0/what-is-swagger/>
9. Linux [Електронний ресурс]: Вікіпедія. Вільна енциклопедія:
<https://uk.wikipedia.org/wiki/Linux>
10. Javadoc [Електронний ресурс]: Вікіпедія. Вільна енциклопедія:
<https://en.wikipedia.org/wiki/Javadoc>
11. Wi-Fi [Електронний ресурс]: Вікіпедія. Вільна енциклопедія:
<https://uk.wikipedia.org/wiki/Wi-Fi>
12. Резервне копіювання [Електронний ресурс]: Вікіпедія. Вільна енциклопедія: <https://en.wikipedia.org/wiki/Backup>

13. Порівняння SSD та HDD [Електронний ресурс]: Intel:
<https://www.intel.com/content/www/us/en/products/docs/memory-storage/solid-state-drives/ssd-vs-hdd.html>
14. Порівняння типів пам'яті [Електронний ресурс]: Kingston:
<https://www.kingston.com/en/blog/pc-performance/difference-between-slc-mlc-tlc-3d-nand>
15. Закон про «Охорону праці» [Електронний ресурс]: Сайт Верховної Ради України <https://zakon.rada.gov.ua/laws/show/2694-12#Text>

ДОДАТОК А – ТЕХНІЧНЕ ЗАВДАННЯ НА РОЗРОБКУ

1 Вступ

1.1 Назва – Вебзастосунок інформаційної система позашкільного навчального закладу

1.2 Призначення програми: надати можливість швидко та зручно керувати інформацією та процесами, необхідними при виконанні діяльності позашкільного навчального закладу.

1.3 Область застосування: позашкільні навчальні заклади.

2 Призначення розробки

Призначена для спрощення, прискорення та структурованості роботи педагогів позашкільного навчального закладу, підвищення ефективності роботи працівників закладу.

3 Вимоги до системи

3.1 Вимоги до функціональних характеристик

Система буде містити наступні функціональні характеристики:

- Управління інформацією про педагогів, учнів та гуртки, які відносяться до відповідного позашкільного навчального закладу;
- Своєчасне повідомлення про нові запити до вчителів та про результат розгляду заявки учнів;
- Управління групами, розподіл учнів за групами;

3.2 Вимоги до надійності

3.2.1 Вимоги до забезпечення надійності функціонування програми

Надійне (стійке) функціонування системи має бути забезпечено надійним (стійким) функціонуванням бездротової мережі і мережі електропостачання в позашкільному закладі, або у іншому місці, де буде розміщений сервер.

3.2.2 Відмова виконання програмного продукту через некоректні дії користувача.

- Перевірка формату вводу даних для коректної роботи.
- Функція автоматичного backup кожен день, тиждень, місяць, рік.

3.3 Умови експлуатації

Данною програмою будуть користуватися працівники, які є впевненими користувачами комп'ютера, та не потребують додаткового навчання по даному ПЗ, але можуть вимагати додаткову допомогу у роботі з вихідними даними.

3.4 Вимоги до інформаційної та програмної сумісності

3.4.1 Вимоги до інформаційних структур і методів розв'язання

Вимоги до інформаційних структур і методів розв'язання відсутні.

3.4.2 Вимоги до вихідних кодів і мов програмування

Додаткові вимоги не відсутні.

3.4.3 Вимоги до програмних засобів , які використовуються програмою

Програмні засоби що використовуються мають бути вільно поширювані

3.5 Вимоги до маркування та упаковки

3.5.1 Вимоги до маркування :

- Читабельність тексту, чіткість ілюстрацій;
- Достовірність інформації про систему

3.5.2 Вимоги до упаковки:

Вимоги до упаковки відсутні.

3.6 Вимоги до транспортування та зберігання

Вимоги до транспортування програми відсутні.

3.7 Спеціальні вимоги

Спеціальні вимоги відсутні.

4 Вимоги до програмної документації

4.1 Попередній склад програмної документації

Попередній склад програмної документації повинен включати в себе технічне завдання;

4.2 Склад програмної документації

Програмна документація складається з наступних програмних документів:

- Технічне завдання ;
- Опис системи ;
- Код програми ;
- Керівництво користувача.

Сертифікація містить список всіх програмних документів .

Технічне завдання містить опис завдання роботи, склад функціональних характеристик програми, опис вимог для розробки.

Опис програми містить опис логічної структури програми, форматів вхідних та вихідних даних, використовуваних технічних засобів.

Код програми містить опис блоків програми і запис блоків в програмному коді.

Керівництво користувача містить інформацію , необхідну для ознайомлення користувача з програмним забезпеченням.

5 Техніко-економічні показники

5.1 Економічні переваги розробки

Орієнтовна економічна ефективність не розраховується.

6 Стадії та етапи розробки програмного забезпечення

6.1 Стадії розробки

Розробка повинна бути проведена в 5 стадій:

1. Технічне завдання
2. Концепція проекту
3. Технічний проект
4. Робочий проект
5. Впровадження

ДОДАТОК Б – ПРОГРАМА І МЕТОДИКА ВИПРОБУВАНЬ

Об'єкт випробувань

Назва об'єкта випробувань: інформаційна система позашкільного навчального закладу, надалі називається «Програма». Програма дозволяє користувачеві пройти автентифікацію та отримати доступ до відповідного модуля.

Ціль випробувань

Перевірка придатності програми до використання за призначенням, її відповідність заданим вимогам і програмній документації. Перевірка реально досягнутих результатів програми.

Вимоги до програми

Програма повинна задовольняти наступним функціональним характеристикам:

- 1) перевірка правильності імені та паролю користувача;
- 2) перевірка правильності даних при додаванні запису про учня.

Користувач повинен вказати ім'я та пароль, ввести дані про учня.

Вимоги до програмної документації

У комплект програмної документації повинні входити наступні документи:

- Загальний опис системи.
- Посібник користувача – адміністратора.
- Інструкція з формування і ведення бази даних.
- Програма і методика випробувань.

- Текст програми.

Засоби і порядок випробувань

Вимоги до складу і параметрів технічних засобів:

Сервер на базі i586_64 або x86.

Вимоги до складу і параметрів програмних засобів :

- 1) операційна система Linux Mint;
- 2) встановлене з'єднання з сервером баз даних PostgreSQL.

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

Попередньо (не пізніше 3 днів) до початку випробувань уточнюються кінцевий склад групи, що буде проводити випробування. Кожен член групи одержує необхідну документацію до програми (список вимог до програми, опис програми, керівництво користувача, програму і методику випробувань).

Порядок проведення випробувань:

- послідовно виконуються контрольні тести; результати заносяться у відповідний журнал;
- виробляється аналіз отриманих результатів і встановлюється відповідність програми заданим вимогам і програмним документам.

Методи випробувань

Робота програми залежить від параметрів, що вводяться користувачем: від імені та паролю користувача, а також параметрів, що відповідають інформації про учня (ім'я, прізвище, електронна пошта, дата народження та інші).

Іспити будуть проводитися методом «чорної шухляди».

Тести будуть створюватися на основі класів еквівалентності.

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

- один тест може покривати тільки один неправильний клас еквівалентності;
- один тест повинний покривати як можна більше число непокритих правильних класів еквівалентності.

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

Таблиця Б.1 – Класи еквівалентності

Операція, що тестується	Вихідні умови	Правильні класи еквівалентності	Неправильні класи еквівалентності
Автентифікація користувача за ім'ям та паролем	Ім'я та пароль правильний	Так (1)	Ні (2)
Збереження даних про учня	Коректні дані про учня	Так (3)	Ні (4)

Тести

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

1) ім'я користувача – user , його пароль – password;

2) коректні дані про учня: ім'я – Петро, по-батькові – Володимирович, прізвище – Шевченко, дата народження: 18.04.2006, електронна пошта: shev@gmail.com, номер телефону: +380686868686.

Тест 1

Класи еквівалентності:

1

Вхідні дані:

Ім'я користувача: user

Пароль користувача: password

Результат:

Ім'я та пароль користувача знайдено. Надано доступ до інтерфейсу користувача.

Тест 2

Класи еквівалентності:

2

Вхідні дані:

Ім'я користувача: usr

Пароль користувача: passwd

Результати:

Ім'я та пароль користувача не знайдено. У доступі до інтерфейсу користувача відмовлено з повідомленням про помилку введення імені та паролю.

Тест 3

Класи еквівалентності :

3

Вхідні дані:

Ім'я: Петро

По-батькові: Володимирович

Прізвище: Шевченко

Дата народження: 18.04.2006

Електронна пошта: shev@gmail.com

Номер телефону: +380686868686.

Результати:

Дані перевірено на коректність та збережено. Видане повідомлення про успішне збереження даних про учня.

Тест 4

Класи еквівалентності:

4

Вхідні дані:

Ім'я: Петро

По-батькові: Володимирович

Прізвище: Шевченко

Дата народження: 18.04.2030

Електронна пошта: shevgmail.com

Номер телефону: 86.

Результати:

Дані перевірено на коректність та не збережено. Видане повідомлення про помилку в даних про учня, некоректність електронної пошти, дати народження та номеру телефону.

ДОДАТОК В – ПРИКЛАДИ ЕКРАННИХ ФОРМ

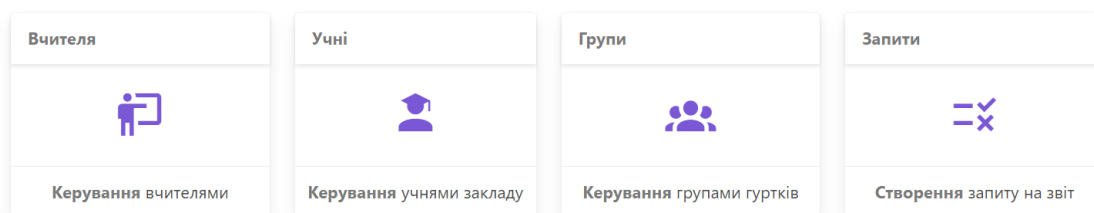


Рисунок В1 – Головна панель керівництва навчального закладу

Керування учнями

Додати нового учня

ID	Ім'я	Прізвище	По батькові	Дата народження	Email	Телефон
1	Ігор	Іванов	Володимирович	2003-01-17	ivanov@gmail.com	+380723777777
2	Іван	Шевченко	Микитович	1998-01-16	shevchenko@gmail.com	+380111111111
6	Анна	Шевчук	Йосипівна	2009-06-28	shevchuk@gmail.com	+380222222222
7	Віктор	Броваренко	Федорович	2008-01-13	brovarenko@gmail.com	+380777777888
8	Анатолій	Крамаренко	Тарасович	2012-08-30	kramarenko@gmail.com	+380777777666
12	Олексій	Пономарчук	Іванович	2011-05-19	ponomarchuk@gmail.com	+380772227777
13	Ніна	Мірошніченко	Янович	2007-03-04	miroshnichenko@gmail.com	+380777777000
21	Ганна	Гнатюк	Федорівна	2009-11-12	gnatiuk@gmail.com	+380777777999
22	Єлізавета	Мельниченко	Анатоліївна	2005-01-07	melnichenko@gmail.com	+380666777777
23	Євген	Кравчук		2005-11-02	krachuk@gmail.com	+380777777555
24	Михайло	Панасюк	Володимирович	2006-10-15	panasiukl@gmail.com	+380777777333

Рисунок В2 – Сторінка керування учнями

Редагування даних про учня

Додати до групи

ID Назва групи

Додати категорію учню

ID Назва категорії

ID

2

Ім'я

Іван

По батькові

Микитович

Прізвище

Шевченко

Email

shevchenko@gmail.com

Номер телефону

+380111111111

Зберегти

Видалити

Рисунок В3 – Форма перегляду деталей та редагування даних про учня

Додати нового учня

Ім'я

По батькові

Прізвище

Дата народження



Номер телефону

Email

Створити

Рисунок В4 – Форма створення нового учня

ДОДАТОК Г – КОД ПРОГРАМИ ТА КОД СТВОРЕННЯ ТАБЛИЦЬ БАЗИ ДАНИХ

Створення таблиці student:

```
create table student
(
    student_id    bigint generated by default as identity
        constraint student_pkey
            primary key,
    active         boolean          not null,
    birth_date     date,
    email_address  varchar(255),
    first_name     varchar(255) not null,
    last_name      varchar(255) not null,
    middle_name    varchar(255),
    phone_number   varchar(255)
);
```

Створення таблиці teacher:

```
create table teacher
(
    teacher_id    bigint generated by default as identity
        constraint teacher_pkey
            primary key,
    active         boolean          not null,
    email_address  varchar(255),
    first_name     varchar(255) not null,
    last_name      varchar(255) not null,
    middle_name    varchar(255),
    phone_number   varchar(255),
    teacher_level  bigint
);
```

Створення таблиці subject:

```
create table subject
(
    subject_id    bigint generated by default as identity
        constraint subject_pkey
            primary key,
    subject_name   varchar(255) not null
);
```

Створення таблиці status:

```
create table status
(
```

```

status_id    bigint generated by default as identity
            constraint status_pkey
                primary key,
status_name varchar(255) not null
);

```

Створення таблиці student_group:

```

create table student_group
(
    group_id    bigint generated by default as identity
                constraint student_group_pkey
                    primary key,
    group_name  varchar(255) not null,
    subject_id  bigint
                constraint fkkglhmaukq7ll0igf88en7u0d6
                    references subject,
    teacher_id  bigint
                constraint fkj4u3hnu8uahrpcepw5bwovv9k
                    references teacher
);

```

Створення таблиці report_request:

```

create table report_request
(
    request_id    bigint generated by default as identity
                  constraint report_request_pkey
                      primary key,
    description    varchar(255),
    request_name    varchar(255) not null,
    request_status integer,
    from_teacher_id bigint
                  constraint fkr30p7ka30xsbgwio8t4crqgp6
                      references teacher,
    to_teacher_id  bigint
                  constraint fk9tnyjmcx177wqe09tolgpjfq
                      references teacher
);

```

Створення таблиці student_request:

```

create table student_request
(
    request_id    bigint generated by default as identity
                  constraint student_request_pkey
                      primary key,
    request_status integer,
    student_id     bigint
);

```

```

        constraint fk1pxdaeqgtjh5gmqqksl59565n
            references student,
teacher_id      bigint
        constraint fk9vm59584hvidagf6chmumb5qi
            references teacher
);

```

Створення таблиці student_status:

```

create table student_status
(
    student_id bigint not null
        constraint fk9vm59584hvidagf6chmumb5qi
            references student,
    status_id  bigint not null
        constraint fkiv6702k8t75t1cn9r7d562mub
            references status,
    constraint student_status_pkey
        primary key (student_id, status_id)
);

```

Створення таблиці student_student_group:

```

create table student_student_group
(
    student_id bigint not null
        constraint fktclobxmlap2y5v8hhusml491a
            references student,
    group_id   bigint not null
        constraint fkew25af9wxjxok5yqpgh36i84i
            references student_group,
    constraint student_student_group_pkey
        primary key (student_id, group_id)
);

```

Модуль для роботи з групами гуртка:

```

package com.registration.controller;

import com.registration.dto.GroupDto;
import com.registration.service.StudentGroupService;
import org.springframework.http.ResponseEntity;
import org.springframework.web.bind.annotation.*;

import javax.validation.Valid;
import java.util.List;

@RestController
@RequestMapping("/groups")

```

```

public class GroupController {

    private final StudentGroupService studentGroupService;

    public GroupController(StudentGroupService studentGroupService)
    {
        this.studentGroupService = studentGroupService;
    }

    @GetMapping("/{groupId}")
    public GroupDto getGroup(@PathVariable Long groupId) {
        return studentGroupService.getGroup(groupId);
    }

    @GetMapping
    public List<GroupDto> getAllGroups() {
        return studentGroupService.getAllGroups();
    }

    @PostMapping
    public GroupDto saveGroup(@RequestBody @Valid GroupDto groupDto)
    {
        return studentGroupService.saveGroup(groupDto);
    }

    @PutMapping
    public ResponseEntity<Void> updateGroup(@RequestBody @Valid
GroupDto groupDto) {
        studentGroupService.updateGroup(groupDto);
        return ResponseEntity.ok().build();
    }

    @DeleteMapping
    public ResponseEntity<Void> deleteGroup(@RequestParam Long
groupId) {
        studentGroupService.deleteGroup(groupId);
        return ResponseEntity.ok().build();
    }
}

package com.registration.service;

import com.registration.dto.*;
import com.registration.entity.StudentGroup;
import com.registration.entity.Subject;
import com.registration.entity.Teacher;
import com.registration.repository.StudentGroupRepository;
import com.registration.repository.SubjectRepository;

```

```

import com.registration.repository.TeacherRepository;
import com.registration.util.mapper.MapStructMapper;
import org.springframework.stereotype.Service;

import javax.transaction.Transactional;
import java.util.List;
import java.util.stream.Collectors;

@Service
public class StudentGroupServiceImpl implements StudentGroupService
{

    private final StudentGroupRepository studentGroupRepository;
    private final SubjectRepository subjectRepository;
    private final TeacherRepository teacherRepository;
    private final MapStructMapper mapper;

    public StudentGroupServiceImpl(StudentGroupRepository
studentGroupRepository, SubjectRepository subjectRepository,
TeacherRepository teacherRepository, MapStructMapper mapper) {
        this.studentGroupRepository = studentGroupRepository;
        this.subjectRepository = subjectRepository;
        this.teacherRepository = teacherRepository;
        this.mapper = mapper;
    }

    @Override
    @Transactional
    public GroupDto saveGroup(GroupDto groupDto) {
        StudentGroup group =
mapper.groupDtoToStudentGroup(groupDto);
        SubjectDto subjectDto = groupDto.getSubjectDto();
        TeacherDto teacherDto = groupDto.getTeacherDto();
        if (subjectDto != null) {
            if (subjectDto.getSubjectId() != null) {
                Subject subject =
subjectRepository.getById(subjectDto.getSubjectId());
                group.setSubject(subject);
            } else if (subjectDto.getSubjectName() != null) {
                Subject subject =
subjectRepository.save(mapper.subjectDtoToSubject(subjectDto));
                group.setSubject(subject);
            }
        }
        if (teacherDto != null) {
            Teacher teacher = teacherDto.getTeacherId() != null ?
teacherRepository.getById(teacherDto.getTeacherId()) :
createNewTeacher(teacherDto);
            group.setTeacher(teacher);
        }
    }
}

```

```

        }
        return
mapper.studentGroupToGroupDto(studentGroupRepository.save(group));
    }

    private Teacher createNewTeacher(TeacherDto teacherDto) {
        return
teacherRepository.save(mapper.teacherDtoToTeacher(teacherDto));
    }

    @Override
    public GroupDto getGroup(Long groupId) {
        StudentGroup group =
studentGroupRepository.findById(groupId);
        ignoreStudents(group);
        return mapper.studentGroupToGroupDto(group);
    }

    @Override
    @Transactional
    public void updateGroup(GroupDto groupDto) {
        StudentGroup group =
studentGroupRepository.findById(groupDto.getId());
        if (groupDto.getName() != null) {
            group.setName(groupDto.getName());
        }

        if (groupDto.getSubjectDto() != null) {
group.setSubject(subjectRepository.findById(groupDto.getSubjectDto().
getSubjectId()));
        }
    }

    @Override
    public void deleteGroup(Long groupId) {
        studentGroupRepository.deleteById(groupId);
    }

    @Override
    public List<GroupDto> getAllGroups() {
        List<StudentGroup> groups =
studentGroupRepository.findAll();
        groups.forEach(this::ignoreStudents);
        return
groups.stream().map(mapper::studentGroupToGroupDto).collect(Collectors.toList());
    }

```

```

        private void ignoreStudents(StudentGroup group) {
            group.getStudents().forEach(s -> s.setGroups(null));
        }
    }

package com.registration.repository;

import com.registration.entity.StudentGroup;
import org.springframework.data.jpa.repository.JpaRepository;
import org.springframework.stereotype.Repository;

@Repository
public interface StudentGroupRepository extends
    JpaRepository<StudentGroup, Long> {

}

package com.registration.entity;

import lombok.Data;

import javax.persistence.*;
import java.util.List;

@Entity(name = "student_group")
@Data
public class StudentGroup {
    @Id
    @GeneratedValue(strategy = GenerationType.IDENTITY)
    private Long groupId;

    @Column(nullable = false)
    private String groupName;

    @ManyToOne(fetch = FetchType.EAGER)
    @JoinColumn(name = "subject_id")
    private Subject subject;

    @OneToOne
    @JoinColumn(name = "teacher_id")
    private Teacher teacher;

    @ManyToMany(mappedBy = "groups")
    private List<Student> students;
}

package com.registration.util.mapper;

```

```

import com.registration.dto.*;
import com.registration.entity.*;
import org.mapstruct.Mapper;
import org.mapstruct.Mapping;

@Mapper(componentModel = "spring")
public interface MapStructMapper {
    Subject subjectDtoToSubject(SubjectDto subjectDto);
    SubjectDto subjectToSubjectDto(Subject subject);

    Teacher teacherDtoToTeacher(TeacherDto teacherDto);
    TeacherDto teacherToTeacherDto(Teacher teacher);

    Student studentDtoToStudent(StudentDto studentDto);
    StudentDto studentToStudentDto(Student student);

    Status statusDtoToStatus(StatusDto statusDto);
    StatusDto statusToStatusDto(Status status);

    StudentGroup groupDtoToStudentGroup(GroupDto groupDto);
    @Mapping(source = "studentGroup.subject", target = "subjectDto")
    @Mapping(source = "studentGroup.teacher", target = "teacherDto")
    GroupDto studentGroupToGroupDto(StudentGroup studentGroup);
}

```

Файл pom.xml, який містить інформацію про залежності проекту:

```

<?xml version="1.0" encoding="UTF-8"?>
<project xmlns="http://maven.apache.org/POM/4.0.0"
xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
    xsi:schemaLocation="http://maven.apache.org/POM/4.0.0
https://maven.apache.org/xsd/maven-4.0.0.xsd">
    <modelVersion>4.0.0</modelVersion>
    <parent>
        <groupId>org.springframework.boot</groupId>
        <artifactId>spring-boot-starter-parent</artifactId>
        <version>2.6.7</version>
        <relativePath/> <!-- lookup parent from repository -->
    </parent>
    <groupId>com.nuos</groupId>
    <artifactId>oos-edu-registration</artifactId>
    <version>0.0.1-SNAPSHOT</version>
    <name>oos-edu-registration</name>

```



```

    <description>Project for Registration Out-Of-School Educational
Company</description>
    <properties>
        <java.version>11</java.version>
    </properties>
    <dependencies>
        <dependency>
            <groupId>org.springframework.boot</groupId>
            <artifactId>spring-boot-starter-security</artifactId>
        </dependency>
        <dependency>
            <groupId>org.springframework.boot</groupId>
            <artifactId>spring-boot-starter-validation</artifactId>
        </dependency>
        <dependency>
            <groupId>org.springframework.boot</groupId>
            <artifactId>spring-boot-starter-web</artifactId>
        </dependency>
        <dependency>
            <groupId>org.springframework.boot</groupId>
            <artifactId>spring-boot-starter-data-jpa</artifactId>
        </dependency>

        <dependency>
            <groupId>org.postgresql</groupId>
            <artifactId>postgresql</artifactId>
            <scope>runtime</scope>
        </dependency>
        <dependency>
            <groupId>org.projectlombok</groupId>
            <artifactId>lombok</artifactId>
            <optional>true</optional>
            <version>1.18.24</version>
        </dependency>
        <dependency>
            <groupId>org.springframework.boot</groupId>
            <artifactId>spring-boot-starter-test</artifactId>
            <scope>test</scope>
        </dependency>
        <dependency>
            <groupId>org.mapstruct</groupId>
            <artifactId>mapstruct</artifactId>
            <version>1.4.2.Final</version>
        </dependency>
        <dependency>
            <groupId>org.hibernate</groupId>
            <artifactId>hibernate-core</artifactId>
            <version>5.4.30.Final</version>

```

```

</dependency>
<dependency>
    <groupId>org.hibernate</groupId>
    <artifactId>hibernate-jpamodelgen</artifactId>
    <version>5.4.30.Final</version>
</dependency>
<dependency>
    <groupId>io.springfox</groupId>
    <artifactId>springfox-boot-starter</artifactId>
    <version>3.0.0</version>
</dependency>
</dependencies>

<build>
    <plugins>
        <plugin>
            <groupId>org.apache.maven.plugins</groupId>
            <artifactId>maven-compiler-plugin</artifactId>
            <version>3.5.1</version>
            <configuration>
                <source>11</source>
                <target>11</target>
                <annotationProcessorPaths>
                    <path>
                        <groupId>org.mapstruct</groupId>
                        <artifactId>mapstruct-
processor</artifactId>
                        <version>1.4.2.Final</version>
                    </path>
                    <path>
                        <groupId>org.projectlombok</groupId>
                        <artifactId>lombok</artifactId>
                        <version>1.18.24</version>
                    </path>
                    <path>
                        <groupId>org.projectlombok</groupId>
                        <artifactId>lombok-mapstruct-
binding</artifactId>
                        <version>0.2.0</version>
                    </path>
                </annotationProcessorPaths>
            </configuration>
        </plugin>
    </plugins>
</build>
</project>

```