

ЗМІСТ	
ПЕРЕЛІК УМОВНИХ ПОЗНАЧЕНЬ.....	8
ВСТУП .....	9
РОЗДІЛ 1. АНАЛІЗ ПРЕДМЕТНОЇ ОБЛАСТІ І ПОСТАНОВКА ЗАДАЧІ ...	10
1.1. Загальні відомості про системи адміністрування .....	10
1.1.1. Система адміністрування та її функції.....	10
1.1.2. Властивості систем адміністрування .....	10
1.2. Огляд існуючих систем адміністрування.....	11
1.3. Проблеми що виникають при створенні системи адміністрування ....	14
1.4. Постановка задачі розробки системи адміністрування спортивними залами .....	20
Висновки по розділу .....	21
РОЗДІЛ 2. АНАЛІЗ ІНФОРМАЦІЙНОГО ЗАБЕЗПЕЧЕННЯ .....	22
2.1. Засоби представлення даних .....	22
2.1.1. Технологія JSP .....	22
2.1.2. Мова розмітки веб-сторінок.....	23
2.1.3. Фреймворк Bootstrap .....	23
2.1.4. Технологія JavaServer Faces .....	25
2.2. Засоби збереження даних.....	27
2.2.1. База даних MySQL .....	27
2.2.2. База даних PostgreSQL.....	28
2.2.3. Фреймворк Hibernate.....	29

					<b>ІК-21.12 1448.01 ПЗ</b>		
Зм.	Лист	№ докум.	Підп.	Дата			
Розроб.	Клейменов				Система адміністрування спортивними залами		
Перев.	Ліхоузова						
Н. контр.	Пасько						
Затв.	Ткач				<div>Лім.</div> <div>Лист</div> <div>Листів</div> <div>6</div> <div>НТУУ «КПІ»</div> <div>Каф. ТК</div> <div>Гр. ІК-21</div>		

2.3.	Засоби створення business-logic (business-layer) .....	31
2.3.1.	Технологія Servlets .....	31
2.3.2.	Архітектурний шаблон MVC .....	33
2.4.	Обґрунтування вибору засобів розробки .....	35
	Висновки по розділу .....	36
РОЗДІЛ 3. РОЗРОБКА АЛГОРИТМІЧНОГО ТА ПРОГРАМНОГО ЗАБЕЗПЕЧЕННЯ .....		37
3.1.	Архітектура програмного забезпечення.....	37
3.2.	Опис структури бази даних .....	38
3.3.	Вимоги до технічного забезпечення.....	41
3.4.	Керівництво користувача.....	41
	Висновки по розділу .....	46
РОЗДІЛ 4. ОХОРОНА ПРАЦІ .....		47
4.1.	Аналіз умов праці .....	47
4.2.	Аналіз шкідливих і небезпечних факторів .....	49
4.2.1.	Аналіз рівня освітлення .....	49
4.2.2.	Аналіз мікроклімату приміщення .....	51
4.2.3.	Аналіз рівня шуму .....	52
4.2.4.	Аналіз рівня випромінювання .....	53
4.3.	Електробезпека .....	53
4.4.	Пожежна безпека .....	54
	Висновки по розділу .....	55
ЗАГАЛЬНІ ВИСНОВКИ .....		57
ПЕРЕЛІК ПОСИЛАНЬ .....		58
ДОДАТКИ.....		59

## ПЕРЕЛІК УМОВНИХ ПОЗНАЧЕНЬ

*JSP (Java Server Pages)* – технологія, що дозволяє веб-розробникам динамічно генерувати HTML, XML та інші веб-сторінки;

*HTML (HyperText Markup Language)* – стандартна мова розмітки веб-сторінок в Інтернеті;

*БД* – база даних;

*API (application programming interface)* – програмний інтерфейс ;

*IT* – Інформаційні технології;

*JS (JavaScript)* – мова програмування;

*WEB* – інтернет-простір;

*NoSQL (not only SQL)* – ряд підходів, спрямованих на реалізацію сховищ баз даних, що мають суттєві відмінності від моделей, які використовуються в традиційних реляційних СУБД;

*CMS (Content Management System)* – програмне забезпечення для організації веб-сайтів чи інших інформаційних ресурсів в Інтернеті чи окремих комп'ютерних мережах.

## ВСТУП

В новітній час інформаційних технологій кожен з нас намагається день за днем знайти та пізнати для себе якнайбільше нової та важливої інформації або отримати певні послуги що нас цікавлять за найкоротший проміжок часу, адже, як всім відомо, кожна секунда нашого єдиного життя являється безцінною.

Ми намагаємося піклуватися про наше здоров'я і різними способами підтримувати його в належному стані, задля цього люди обирають здорове харчування, здоровий спосіб життя, спеціальні медичні препарати, відпочинки на курортах та фізичні вправи.

Фізичні вправи завжди вважались одним з основних способів підтримання належного стану здоров'я, саме тому ходячи до спортивної зали кожен з нас намагається витратити якнайменше часу на підписання різноманітних угод та оплату цього спортивного комплексу, а більше часу приділити самому процесу тренування, зміцнюючи цим самим своє фізичне самопочуття та здоров'я в цілому.

Більшість студентів мають досить обмежений час для своїх забаганок, саме тому було вирішено зробити сервіс з адміністрування спортивними залами гуртожитків, що дозволить кожному студенту витрачати більше часу саме на тренування, а адміністратору – швидше та краще вдосконалювати спортивний інвентар та зал в цілому.

## РОЗДІЛ 1. АНАЛІЗ ПРЕДМЕТНОЇ ОБЛАСТІ І ПОСТАНОВКА ЗАДАЧІ

### 1.1. Загальні відомості про системи адміністрування

#### 1.1.1. Система адміністрування та її функції

Система адміністрування – інформаційна система або комп'ютерна програма, яка використовується для забезпечення і організації спільного процесу створення, редагування і управління контентом (тобто вмістом).

Основні функції системи:

- Надання інструментів для створення вмісту, організація спільної роботи над вмістом;
- Управління вмістом: зберігання, контроль версій, дотримання режиму доступу, управління потоком документів і т. п.;
- Публікація вмісту;
- Подання інформації у вигляді, зручному для навігації та пошуку.

В системі адміністрування можуть перебувати найрізноманітніші дані: документи, фільми, фотографії, номери телефонів, наукові дані і так далі. Така система часто використовується для зберігання, управління, перегляду і публікації документації. Контроль версій є одним з основних її переваг, коли вміст змінюється групою осіб.

#### 1.1.2. Властивості систем адміністрування

Система адміністрування – програма, що надає інструменти для додавання, редагування, видалення інформації у веб-додатку.

Більшість сучасних систем мають модульну архітектуру, що дозволяє адміністратору самому вибирати і налаштовувати ті дані та компоненти, які необхідні та актуальні в конкретний проміжок часу.

Типові модулі:

- динамічне або статичне меню,

Зм	Лист	№ докум.	Підп.	Дата

**ІК-21.12 1448.01 ПЗ**

Аркуш

10

- відгуки користувачів,
- новини,
- опитування,
- пошук по сайту,
- статистика,
- гостьова книга і т. д.

Веб-додатки, організовані як системи управління контентом (системи адміністрування), засновані на наступних технологіях: веб-сервер, сховище даних (найчастіше СУБД, наприклад такі як MySQL або PostgreSQL, проте існують і NoSQL CMS), веб-додаток для забезпечення роботи самої системи, редактор сторінок, файловий менеджер з веб-інтерфейсом для управління файлами сайту і система управління правами користувачів.

## 1.2. Огляд існуючих систем адміністрування

Систему адміністрування спортивною залою можна порівняти з багатьма інтернет-магазинами, кожен з яких має свою адміністративну частину та частину користувача. Відмінність між магазинами в основному лежить у способі оплати (готівковий та безготівковий) за певною платіжною системою (у разі оплати послуги \ товару за допомогою електронних коштів) та в послугах, що надаються. Сервіс адміністрування спортивною залою надає можливість користувачеві (в нашому випадку це студент гуртожитку) придбати абонемент на певну кількість місяців за допомогою інтернет-додатку, витрачаючи на це мінімум часу та зусиль. З іншої сторони, адміністратору надається можливість приймати як електронні платежі так і готівкові, це залежить від побажання користувача, який саме шлях для отримання абонементу на відвідування залою він обрав. Також адміністратору надається ще ряд інших можливостей, що дозволяють якісніше обслуговувати спортивну залу.

Зм	Лист	№ докум.	Підп.	Дата

**ІК-21.12 1448.01 ПЗ**

Аркуш

11

Одним з найяскравіших прикладів системи адміністрування є інтернет-магазин *rozetka.com.ua*. Даний сервіс дозволяє користувачам придбати різноманітні типи товарів та послуг, використовуючи при цьому зручні для клієнта, що користується їхніми послугами, способи оплати (готівкова та безготівкова оплата). Оглядаючи клієнтську частину даного сервісу (адміністративна частина для звичайних користувачів недоступна) можна побачити що даний сервіс орієнтований на кінцевого користувача, інтерфейс взаємодії з користувачем розроблений враховуючи принципи «Дружнього інтерфейсу», оглядаючи сторінки сервісу інтуїтивно видно де і яка інформація знаходиться, виділені найважливіші деталі що сприяють покупці певного товару або послуги в один клік, створені сервіси допомоги клієнтам, які функціонують двадцять чотири години на добу. Для системи адміністрування спортивною залою сервіс *rozetka.com.ua* є досить гарним прикладом, але даний магазин надає досить багато функціоналу у клієнтській частині, що не потрібен у системі адміністрування спортивною залою. При оплаті абонементів не потрібний «кошик» для товарів, адже дана система надає товар тільки одного типу – абонемент до залу, тобто немає необхідності накопичувати однотипні товари для придбання в один клік. Виходячи з цієї ж причини не потрібно організовувати у клієнтській частині складні багат шарові меню, адже весь функціонал клієнта можна зручно і наочно розташувати на декількох сторінках, створивши зручну та інтуїтивно зрозумілу систему переходів між сторінками, що дозволить клієнту ефективно та швидко придбати абонемент на необхідні місяці відвідування залу. Важливою складовою клієнтської частини сервісу *rozetka.com.ua* є особистий кабінет клієнта що дозволяє слідкувати за історією придбання товарів, аналогічний функціонал реалізується й системою адміністрування спортивною залою, адже облік грошей, що студент витрачає, в тому числі й на відвідування залу є досить важливою складовою студентського життя.

Зм	Лист	№ докум.	Підп.	Дата

**ІК-21.12 1448.01 ПЗ**

Аркуш

12

Найбільш приближеним до необхідної нам системи адміністрування спортивною залом для гуртожитків є сервіс *www.sportlife.ua*. Даний сервіс надає величезну кількість послуг що пов'язані на пряму із здоров'ям людини, наприклад:

- абонементи на відвідування басейну;
- абонементи на відвідування тренажерного залу;
- абонементи на групові заняття спортом;
- абонементи на дитячий фітнес;
- абонементи на настільний теніс;
- абонементи на сквош;
- абонементи на ігровий зал;
- і т. д.

Також даний сервіс надає додаткову різносторонню інформацію, таку як:

- інформація про компанію;
- місцезнаходження залів, де можна отримати вказані послуги;
- служба підтримки;
- і т. д;

Враховуючи потреби і можливості студмістечка (в нашому випадку студмістечка КПП) сервіс адміністрування спортивною залом буде мати схожий функціонал з сервісом *www.sportlife.ua*, за виключенням того, що наш сервіс фізично не може надати деякі послуги, що присутні у *www.sportlife.ua*. Як і у попередньому випадку з *rozetka.com.ua* ми не маємо можливості оглянути та опрацювати всі плюси та мінуси адміністративної частини сервісу *www.sportlife.ua*.

Розглянувши декілька найбільш популярних систем адміністрування, інтернет-магазин та інтернет-сервіс з надання послуг у спортивній сфері, можна зробити висновок що кожен з цих сервісів має функціонал який

Зм	Лист	№ докум.	Підп.	Дата



необхідно втілити в систему адміністрування спортивною залом, та функціонал непотрібний для нашої системи. Виходячи з цього стає зрозуміло чому потрібно створити та впровадити окремий додаток щодо адміністрування спортивними залами гуртожитків, адже необхідна система, яка буде відповідати всім вимогам та підходити до середовища студентського містечка, включаючи необхідний функціонал, що дозволяє швидко і ефективно управляти спортивною залом, правильно розпоряджатись своїм часом, не витрачаючи його на непотрібні дрібниці.

### 1.3. Проблеми що виникають при створенні системи адміністрування

При вирішенні даної задачі виникають проблеми, що пов'язані зі створенням зручної та ефективної клієнтської частини та частини адміністрування.

Однією з проблем створення вдалої та ефективної системи адміністрування спортивною залом є дотримання правил проектування дружнього інтерфейсу, що дозволяють ефективно застосовувати додаток, спрощують його розуміння користувачем та підвищують ефективність роботи користувача з даною системою без зайвого навантаження на мозок людини.

При розробці як клієнтської частини так і частини адміністрування користуються наступними принципами:

- всі сторінки та елементи інтерфейсу, які використовуються в системі, повинні мати властивості наочності, тобто користувач повинен витратити мінімум часу та зусиль для розпізнавання того, що йому потрібно зробити, або ідентифікувати для ухвалення рішення. Кольори, шрифти, фон, контрастність, піктограми й інші елементи інтерфейсу повинні сприяти максимальному сприйняттю інформації, розташованої перед

користувачем і спонукати його зробити всі дії, в тому числі придбання абонементу у спортивну залу, за один клік.

- користувач не повинен витрачати зайвого часу на розпізнавання елементів інтерфейсу, що раніше не використовувались. Користувач має знаходити очікувану інформацію в областях полів чи екрану або форм, в яких ця інформація з'являлась раніше, тобто при переході від форми до форми інформація повинна знаходитись користувачем в одних і тих же місцях;
- якщо дії користувача можуть привести до втрати інформації, порушення взаємозв'язку з середовищем, то додаток повинний видавати користувачу інформацію, яка попереджає його про можливі неприємності;
- якщо система виконує якісь дії тривалий час, то обов'язково повинна бути використана якась компонента (типу Progress bar) з повідомленням про тривалість процесу;
- необхідно поліпшувати вигляд екрану завдяки введенню різноманітних оздоблюючих елементів;
- увесь додаток повинен бути прозорим і не мати обтяжливих елементів, що ускладнюють роботу з додатком. Використовується підхід поступового зменшення підказок, додаткової інформації для того користувача, що вже вивчив правила користування додатком;

Враховуючи те що створюється система адміністрування спортивною залою, можна сказати, що для створення клієнтської частини додатку вказаних принципів буде достатньо, що не можна сказати за адміністративну частину сервісу, адже треба враховувати те, що адміністратор має набагато більший функціонал по управлінню спортивною залою та додатком в цілому. Необхідно створити ефективний та, водночас, зручний інтерфейс для моніторингу обліку оплат абонементів, функціонал з ефективного

планування часу роботи спортивної зали та її завантаженням клієнтами (студентами). Для цього адміністратор повинен мати доступ до деякої статистичної інформації, що показує яка кількість студентів займається кожен місяць, як варіюється це число, має статистичні дані з витрат на дану спортивну залу, та кількість грошей що доступні для удосконалення спортивного обладнання зали. Вся ця інформація допомагає адміністратору приймати рішення щодо подальших дій з управління та оснащення спортивною залю певним необхідним в даний момент часу спортивним інвентарем.

Наступною проблемою, що виникає під час створення майже всіх сервісів даного типу, є підключення платіжної системи з оплати різноманітних послуг (використовуючи електронні гроші).

*Електронні гроші* — означення грошей чи фінансових зобов'язань, обмін та взаєморозрахунки з яких проводяться за допомогою інформаційних технологій. Відповідно до законодавства України, електронні гроші – одиниці вартості, які зберігаються на електронному пристрої, приймаються як засіб платежу іншими особами, ніж особа, яка їх випускає, і є грошовим зобов'язанням цієї особи, що виконується в готівковій або безготівковій формі.

Ця проблема є досить ваговою, оскільки без підключення платіжної системи не буде фактичної оплати абонементів та, як результат, вдосконалення спортивної зали буде відбуватися менш ефективно.

Існує два способи вирішення проблеми, пов'язаної з оплатою послуг в безготівковій формі: перший – підключати та налаштовувати оплату за послуги для кожної платіжної системи окремо (а це дуже клопітна робота з документами, на яку уходить дуже багато часу та сил) і другий спосіб – скористатися послугами компанії, що є платіжним агрегатором.

Зм	Лист	№ докум.	Підп.	Дата

Другий спосіб дозволяє скоротити час для запуску сервісу, в нашому випадку – системи адміністрування, адже для отримання послуг від даної компанії необхідно лише підписати декілька договорів, і тоді користувачі нашої системи самостійно зможуть вибрати саме той спосіб оплати, який для них є найзручнішим. В ці способи входять: електронні гаманці, оф-лайн методи оплати та, звичайно ж, оплата банківською карткою.

Кожна з існуючих платіжних систем, що використовуються у різноманітних інтернет – магазинах та інших сервісах, мають свої недоліки та переваги.

#### Платіжна система *LiqPay*

*LiqPay* – система електронних платежів, розроблена в ПриватБанку. Дана платіжна система надає основні види оплати:

1. Переказ грошей на карту ПриватБанку;
2. Оплата кредитними картками VISA / MASTERCARD в системі LiqPay;
3. Переказ грошей в системі Приват24.

Дана система дає можливість перевести в готівку чеки від Google, завантажуючи на сайт відскановані зображення лицьового та зворотного боку чека, можна продивитись свої транзакції, можливий переказ грошей між пластиковою карткою і внутрішнім рахунком і навпаки, а також між пластиковими картами і між внутрішніми рахунками, причому одержувачем може бути як VISA, так і будь-яка інша карта ПриватБанку. На сайті можна поповнити рахунок мобільного телефону (винесена в окремий пункт меню) і оплатити інші послуги, такі як:

- Інтернет-провайдери;
- Послуги зв'язку;
- Охоронні системи;
- Платіжні та обмінні сервіси;
- Телебачення і радіо;

Зм	Лист	№ докум.	Підп.	Дата

**ІК-21.12 1448.01 ПЗ**

Аркуш

17

- Хостинг і домени;
- Комунальні послуги.

З недоліків даної системи можна навести те що на головній сторінці зображення-посилання перекриває новини і веде на сторінку, доступну для перегляду тільки зареєстрованим користувачам; наявна незрозуміла логіка відкриття посилань в різних вікнах; на сторінці зі списком послуг було б непогано збільшити шрифт; виключені чекбокси виглядають як неактивні.

#### Платіжна система *Portmone.com*

Міжбанківська система доставки та оплати рахунків *Portmone.com* – один з найбільш популярних способів оплати рахунків в Уанеті. Користувачі системи не обов'язково повинні бути клієнтами якогось певного банку – досить бути власником платіжної картки міжнародних систем VISA або MasterCard.

Як і більшість компаній забезпечуючих моментальні і регулярні платежі, *Portmone.com* дозволяє оплачувати послуги мобільного та фіксованого зв'язку, доступу в інтернет, комунальні послуги (в Києві), а так само цілий ряд інших послуг. Однак даний сегмент є всього лише роздрібним бізнесом і не вичерпує всіх напрямків діяльності даної системи.

Найголовнішою характеристикою даної системи є безпека , оскільки одне з основних питань, яке турбує користувачів і партнерів, полягає в безпеці: система отримує доступ до даних платіжної картки користувача. Це доведено відгуками користувачів та безперервною роботою даної системи на протязі багатьох років.

Може скластися враження, що *Portmone.com* здійснює прийом платежів і переказ коштів на користь продавців послуг. Як відомо, для цього потрібна спеціальна ліцензія. Однак технічно (і юридично) це не так. Платежі в системі проходять від банків до компаній-одержувачів безпосередньо.

Зм	Лист	№ докум.	Підп.	Дата

**ІК-21.12 1448.01 ПЗ**

Аркуш

18

*Portmone.com* збирає мінімум статистики необхідної для ідентифікації клієнта та питань безпеки. Ці дані є закритими і не передаються ніякої третій стороні, включаючи партнерів.

Питання сплати податків лягає на плечі учасників платіжного процесу, в число яких, як ми вже з'ясували, *Portmone.com* не входить. Податки відраховуються з надходжень за підключення провайдерів послуг, підтримку і абонентську плату.

*Portmone.com* є продуктом, орієнтованим на сегменти P2B (моментальні і регулярні платежі) і B2B (платформа для впровадження сервісу по прийому платежів). Це дозволяє йому охопити максимально можливу аудиторію платників.

На сьогоднішній день система має ще не досить міцні позиції в регіонах, однак це показово, і характеризує ступінь проникнення інтернету в різних областях України.

#### Платіжна система *Western Union*

Компанія *Вестерн Юніон* (повна назва *Western Union Financial Services*) – визнаний світовий лідер у сфері термінових грошових переказів. У створеній нею системі діє понад 285 000 пунктів в 200 країнах і територіях, здійснюючи найшвидші перекази грошей. Послуги *Western Union* доступні, практично, на всій земній кулі.

#### Які послуги надає *Вестерн Юніон*?

В Україні *Western Union* пропонує клієнтам два види грошових переказів:

##### а) Грошові перекази до запитання (*Western Union Will Call Transfers*).

Це переклади між фізичними особами, які здійснюються без відкриття банківських рахунків. Вони можуть бути отримані адресатом вже через кілька хвилин після відправлення в будь-якому пункті обслуговування системи *Western Union*. Відправляючи даний переклад, можна скористатися

Зм	Лист	№ докум.	Підп.	Дата

**ІК-21.12 1448.01 ПЗ**

Аркуш

19

такими додатковими послугами: доставка переказу адресату, додаткове повідомлення, повідомлення одержувача по телефону про грошовий переказ та інші.

б) Прискорені платежі (Western UnionQuick Pay).

Така послуга дозволяє фізичним особам оплачувати готівкою рахунки різних організацій, які для цього повинні мати спеціальну угоду з *Western Union*. Даний вид перекладу отримав спочатку широке поширення в США і деяких західноєвропейських країнах, а в останні роки – і в Україні, Білорусі, Молдові, Литві, Латвії.

Враховуючі всі аспекти роботи розглянутих платіжних систем можна сказати, що для системи адміністрування спортивною залом гуртожитку найбільш прийнятною та підходящою за функціоналом є платіжна система *Portmone.com*, оскільки її надійність та функціонал дозволяють забезпечити всі необхідні задачі нашого сервісу.

#### 1.4. Постановка задачі розробки системи адміністрування спортивними залами

Метою роботи є створення зручної системи адміністрування спортивними залами гуртожитків НТУУ «КПІ» яка дозволить економити час студентам-користувачам та адміністратору зали. В результаті проведеного аналізу існуючих аналогів та предметної області було прийнято рішення розробити власну систему адміністрування .

Вимоги, яким має відповідати розроблена система:

- простота у використанні;
- система надає API для збереження даних, що надходять в online-режимі в єдину БД;
- інтуїтивно зрозумілий інтерфейс;
- додаток не залежить від платформи, на якій він використовується.

Процес створення системи адміністрування спортивною залом можна розділити на 3 складових – розробка database-layer для збереження даних якими оперують як клієнт так і адміністратор, створення business-layer для надання даному сервісу бізнес-характеристик, таких як способи нарахування грошей та підключення систем безготівкової оплати та створення presentation-layer для зручного та зрозумілого відображення даних системи. Отже задачі, що вирішуються в роботі:

- створення клієнтської частини для оплати послуг та перегляду статистичних даних;
- створення адміністративної частини для управління спортивною залом та веб-додатком в цілому;
- створення API для збереження даних;
- створення зручних грошових звітів;
- створення платформонезалежного додатку;
- забезпечення зручної та ефективної взаємодії користувача з додатком;
- економія часу клієнта;
- економія часу адміністратора.

#### Висновки по розділу

У першому розділі проведено аналіз існуючих рішень та предметної області, проаналізовані основні сучасні способи створення систем адміністрування різноманітних типів та визначено переваги та недоліки цих рішень. В результаті проведеного аналізу сформульована постановка задачі, наведене призначення, цілі та задачі розробки.



## РОЗДІЛ 2. АНАЛІЗ ІНФОРМАЦІЙНОГО ЗАБЕЗПЕЧЕННЯ

### 2.1. Засоби представлення даних

#### 2.1.1. Технологія JSP

*JSP (Java Server Pages)* – технологія, що дозволяє веб-розробникам створювати вміст, який має як статичні, так і динамічні компоненти [6]. Сторінка JSP містить текст двох типів: статичні вихідні дані, які можуть бути оформлені в одному з текстових форматів HTML [7], SVG, WML, або XML, і JSP-елементи, які конструюють динамічний вміст. Крім цього можуть використовуватися бібліотеки JSP-тегів, а також EL (Expression Language), для впровадження Java-коду в статичний вміст JSP-сторінок.

Код JSP-сторінки транслюється в Java-код сервлету за допомогою компілятора JSP-сторінок Jasper, і потім компілюється в байт-код віртуальної машини java (JVM) [8]. Контейнери сервлетів, здатні виконувати JSP-сторінки, написані на платформонезалежній мові Java. JSP-сторінки завантажуються на сервері і управляються зі структури спеціального Java server packet, який називається Java EE Web Application. Зазвичай сторінки упаковані в файлові архіви .war і .ear.

#### *Переваги:*

- Кросплатформеність;
- Широка поширеність;
- Відкритий вихідний код;
- Велика кількість готових рішень і бібліотек;
- Низька вартість підтримки додатків.

#### *Недоліки:*

- Недостатньо розвинене ООП;
- Неузгоджений синтаксис вбудованих функцій і порядок їх параметрів;
- Відсутність багатопоточності.

Зм	Лист	№ докум.	Підп.	Дата

**ІК-21.12 1448.01 ПЗ**

Аркуш

22

### 2.1.2. Мова розмітки веб-сторінок

*HTML* — стандартна мова розмітки веб-сторінок в Інтернеті. Більшість веб-сторінок створюються за допомогою мови HTML (або XHTML). Документ HTML оброблюється браузером та відтворюється на екрані у звичному для людини вигляді.

HTML є похідною мовою від SGML, успадкувавши від неї визначення типу документу та ідеологію структурної розмітки тексту.

Попри те, що HTML — штучна комп'ютерна мова, вона не є мовою програмування.

HTML разом із каскадними таблицями стилів та вбудованими скриптами — це три основні технології побудови веб-сторінок.

HTML впроваджує засоби для:

- створення структурованого документу шляхом позначення структурного складу тексту: заголовки, абзаци, списки, таблиці, цитати та інше;
- отримання інформації із Всесвітньої мережі через гіперпосилання;
- створення інтерактивних форм;
- включення зображень, звуку, відео, та інших об'єктів до тексту.

### 2.1.3. Фреймворк Bootstrap

*Bootstrap* — це безкоштовний набір інструментів з відкритим кодом, призначений для створення веб-сайтів та веб-застосунків, який містить шаблони CSS та HTML для типографіки, форм, кнопок, навігації та інших компонентів інтерфейсу, а також додаткові розширення JavaScript. Він спрощує розробку динамічних веб-сайтів і веб-застосунків.

*Bootstrap* — це клієнтський фреймворк, тобто інтерфейс для користувача, на відміну від коду серверної сторони, який знаходиться на сервері. Репозиторій з даним фреймворком є одним з найбільш популярних на GitHub. Серед інших, його використовують NASA і MSNBC.

Bootstrap має модульну структуру і складається переважно з наборів таблиць стилів LESS, які реалізують різні компоненти цього набору інструментів. Розробники можуть самостійно налаштовувати файли Bootstrap, обираючи компоненти для свого проекту.

Основні інструменти Bootstrap:

- *Сітки* (grid) — наперед задані, готові до використання колонки
- *Шаблони* (template) — фіксовані чи адаптивні шаблони сторінок
- *Типографіка* (typography) — опис та визначення класів для шрифтів, таких як шрифти для коду, цитат тощо
- *Мультимедіа* (media) — засоби управління зображеннями та відео
- *Таблиці* (table) — засоби оформлення таблиць, які зокрема забезпечують сортування
- *Форми* (form) — класи для оформлення як форм, так і деяких подій
- *Навігація* (nav, navbar) — класи для оформлення вкладок, сторінок, меню і панелей навігації
- *Сповіщення* (alert) — класи для оформлення діалогових вікон, підказок і спливаючих вікон
- *Іконочний шрифт* (icon font) — набір іконок у вигляді шрифту, складається майже з 500 компонентів.

*Переваги:*

- Однакові кнопки, однакові поля форм, вид таблиць і т.д.;
- Забезпечує сумісність сторонніх розширень з шаблоном;
- Верстка сторінок з використанням bootstrap є хорошим рішенням для back-end розробників і тих, хто хоче мати акуратний інтерфейс, не докладаючи до цього зайвих зусиль;
- Ефективна JavaScript складова.

### Недоліки:

- Всі сайти, які використовують Bootstrap, схожі один на одного, тобто відсутня унікальність. Подібні сайти-близнюки просто не запам'ятовуються.
- Негнучкість. Якщо потрібно щось більш-менш відмінне від стандартного Bootstrap, то доводиться з найперших кроків боротися зі стилями за замовчуванням. На практиці виконується подвійна робота:
  - верстаємо те, що потрібно;
  - змушуємо це працювати поверх стандартних стилів Bootstrap.
- Надмірний код. Те, що реально зробити двома вкладеними блоками, часто робиться п'ятьма. Будь-які зміни тягнуть за собою інші зміни.
- Зважаючи на свою простоту, деякі розробники використовують компоненти Bootstrap не за прямим призначенням.

#### 2.1.4. Технологія JavaServer Faces

*JavaServer Faces* (JSF) – це каркас програмування, технологія для веб-застосунків, що написані на Java. Він служить для того, щоб полегшувати розробку користувацьких інтерфейсів для Java EE застосунків. На відміну від більшості MVC фреймворків, які керуються запитами, підхід JSF ґрунтується на використанні компонентів. Стан компонентів користувацького інтерфейсу зберігається, коли користувач запитує нову сторінку й потім відновлюється, якщо запит повторюється. Для відображення даних звичайно використовується JSP, але JSF можна пристосувати й під інші технології, наприклад XUL.

Користь технології JSF обумовлена, в першу чергу, наявністю специфікації JSF. Специфікація дозволяє розробляти JSF фреймворки з різним призначенням та різною внутрішньою структурою. Вона лиш гарантує, що фреймворк буде підпорядкований певній структурі. Але з іншого боку специфікація дуже обмежує еталонну реалізацію в освоєнні

Зм	Лист	№ докум.	Підп.	Дата

нових можливостей. Тобто наприклад аїах-технології такі як Ajax4JSF включають дуже багато інтеграційного коду який виникає через потребу в узгодженні еталонної реалізації з основними вимогами специфікації. Можна відмітити, що сама специфікація розроблена досить неоднорідно. Плюсами специфікації є: дерево компонентів, підтримка різних технологій представлення(JSP, Facelets), підтримка різноманітних рендерерів — класів, що відповідають за відображення компоненту, підтримка обробки подій і перевіркою інформації, що вводиться, визначення навігації, а також підтримку інтернаціоналізації (i18n) і доступності (accessibility). Але є й недоліки: дуже великий обсяг коду для реалізації ітераційних компонентів, відсутності обробки повідомлень в середині ітераційного компоненту, великий обсяг шаблонного коду, який можна було б опустити, при реалізації власних компонентів(custom component), непродуманість певних архітектурних рішень в специфікації, щодо реалізації аїах, управління станом дерева компонентів, пошуку по дереву компонентів. Специфікація JSF 1.0 та 1.1 була розроблена завдяки Java Community Process як JSR 127, а JSF 1.2 як JSR 252. Майбутня JSF 2.0 буде розроблена як JSR 314. Сама специфікація не належить до жодної компанії і розроблюється групою експертів з таких відомих компаній як Sun, Oracle, IBM, Novell, Macromedia, BEA Systems, Hewlett-Packard, Siemens AG. Таким чином технологію JSF можна віднести до відкритих стандартів.

### *Переваги*

- генерація серверної частини інтерфейса користувача;
- базується на компонентах(ніякого HTML);
- наявна обробка подій(event) та станів(states);
- різноманітні view-технології — не тільки HTML та JavaScript;
- розробка з урахуванням доступного інструментарію;
- наявний стандарт і він включений до Java EE;
- рольова модель розробки веб.

Зм	Лист	№ докум.	Підп.	Дата

**IK-21.12 1448.01 ПЗ**

Аркуш

26

### *Недоліки*

- потрібно багато часу для вивчення та освоєння технології;
- потребуються потужні обчислювальні можливості серверу.

## 2.2. Засоби збереження даних

### 2.2.1. База даних MySQL

*MySQL* – вільна система керування реляційними базами даних.

MySQL була розроблена компанією «ТсХ» для підвищення швидкодії обробки великих баз даних. Ця система керування базами даних (СКБД) з відкритим кодом була створена як альтернатива комерційним системам. MySQL з самого початку була дуже схожою на mSQL, проте з часом вона все розширювалася і зараз MySQL — одна з найпоширеніших систем керування базами даних. Вона використовується, в першу чергу, для створення динамічних веб-сторінок, оскільки має чудову підтримку з боку різноманітних мов програмування.

#### *Відомі користувачі:*

- *Apple* — для динамічних сторінок на Apple.com, для багатьох внутрішніх проектів
- *Amazon.com* — для багатьох внутрішніх проектів
- *Cox Communications* — четвертий за розміром провайдер кабельного телебачення в США, має понад 3,600 таблиць та 2 мільярди рядків даних у базах і виконує приблизно 2 мільйони вставок на годину.
- *Digg*.
- *Flickr*
- *Google* — для програми AdWords пошукового рушія
- *LiveJournal* — приблизно 300 мільйонів переглядів сторінок на день
- *MediaWiki i Wikinedia*

- *NASA* — конвертувала базовану на Oracle систему закупок на MySQL
- *Nokia*
- *Yahoo!*

*Переваги:*

- простота у встановленні та використанні;
- підтримується необмежена кількість користувачів, що одночасно працюють із БД;
- кількість рядків у таблицях може досягати 50 млн;
- висока швидкість виконання команд;
- наявність простої і ефективної системи безпеки.

*Недоліки:*

- MySQL версії 5.1 має 20 відомих серйозних помилок в додаток до 35 дефектів версії 5.0. Критичні помилки іноді не виправляються протягом тривалих періодів часу. Одним із прикладів є критична помилка, що відома з 2003 року.
- MySQL показує низьку ефективність при використанні її як сховища даних, це частково пов'язано з нездатністю використовувати декілька процесорів для обробки одного запиту. До того ж, MySQL часто критикують за те, що ця СКБД має розходження зі стандартом SQL щодо трактування NULL значень і значення за замовчуванням.

### 2.2.2. База даних PostgreSQL

*PostgreSQL* — об'єктно-реляційна система керування базами даних (СКБД). Є альтернативою як комерційним СКБД (Oracle Database, Microsoft SQL Server, IBM DB2 та інші), так і СКБД з відкритим кодом (MySQL, Firebird, SQLite).

Порівняно з іншими проектами з відкритим кодом, такими як Apache, FreeBSD або MySQL, PostgreSQL не контролюється якоюсь однією

Зм	Лист	№ докум.	Підп.	Дата

**IK-21.12 1448.01 ПЗ**

Аркуш

28

компанією, її розробка можлива завдяки співпраці багатьох людей та компаній, які хочуть використовувати цю СКБД та впроваджувати у неї найновіші досягнення.

*Переваги:*

- Відкрите ПЗ відповідає стандарту SQL-PostgreSQL – безкоштовне ПЗ з відкритим вихідним кодом.
- Велике співтовариство – існує досить велика спільнота в якому ви за простою знайдете відповіді на свої питання
- Велика кількість доповнень – незважаючи на величезну кількість вбудованих функцій, існує дуже багато доповнень, що дозволяють розробляти дані для цієї СУБД і управляти ними.
- Розширення – існує можливість розширення функціоналу за рахунок збереження своїх процедур.

*Недоліки:*

- Продуктивність – при простих операціях читання PostgreSQL може значно уповільнити сервер і бути повільніше своїх конкурентів, таких як MySQL
- Популярність – за своєю природою, популярністю ця СУБД похвалитися не може, хоча і є досить велика спільнота.
- Хостинг – в силу вище перерахованих факторів іноді досить складно знайти хостинг з підтримкою цієї СУБД.

### 2.2.3. Фреймворк Hibernate

*Hibernate* — засіб відображення між об'єктами та реляційними структурами (object-relational mapping, ORM) для платформи Java. Hibernate є вільним програмним забезпеченням, яке поширюється на умовах GNU Lesser General Public License [3]. Hibernate надає легкий для використання каркас (фреймворк) для відображення між об'єктно-орієнтованою моделлю даних і традиційною реляційною базою даних.



Метою Hibernate є звільнення розробника від значних типових завдань із програмування взаємодії з базою даних. Розробник може використовувати Hibernate як при розробці з нуля, так і для вже існуючої бази даних.

Hibernate піклується про зв'язок класів з таблицями бази даних (і типів даних мови програмування із типами даних SQL), і надає засоби автоматичної побудови SQL запитів й зчитування/запису даних, і може значно зменшити час розробки, який зазвичай витрачається на ручне написання типового SQL і JDBC коду. Hibernate генерує SQL виклики і звільняє розробника від ручної обробки результуючого набору даних, конвертації об'єктів і забезпечення сумісності із різними базами даних.

Hibernate забезпечує прозору підтримку збереження даних, тобто їхньої персистентності (англ. *persistence*) для «POJO»-об'єктів, себто для звичайних Java-об'єктів; єдина сувора вимога до класу, що зберігається — конструктор за замовчанням [2].

Hibernate забезпечує використання SQL-подібної мови Hibernate Query Language (HQL), яка дозволяє виконувати SQL-подібні запити, записані поряд з об'єктами даних Hibernate. *Запити критеріїв* надаються як об'єктно-орієнтована альтернатива до HQL.

Hibernate може використовуватись як у самостійних програмах Java, так і в програмах Java EE, що виконуються на сервері (наприклад, сер влети чи EJB session beans). Також він може включатись як додаткова можливість до інших мов програмування. Наприклад, Adobe інтегрував Hibernate у дев'яту версію ColdFusion (що запускається на серверах з підтримкою додатків J2EE) з рівнем абстракції нових функцій і синтаксису, доданих до CFML.

#### *Переваги:*

- Немає необхідності писати рутинні insert / update / delete / select для CRUD операцій;
- Умови зв'язку між об'єктами (рядками таблиць) вказуються декларативно в одному місці;

Зм	Лист	№ докум.	Підп.	Дата

**ІК-21.12 1448.01 ПЗ**

Аркуш

30

- Можливість використовувати поліморфні запити для ієрархій класів;
- Високий ступінь незалежності від конкретної СУБД.

*Недоліки:*

- Можливі проблеми з продуктивністю для складних запитів на об'єктному SQL;
- Ускладнює використання специфічних конструкцій мови SQL конкретної СУБД;
- The object-relational impedance mismatch.

### 2.3. Засоби створення business-logic (business-layer)

*Бізнес-логіка* – в розробці інформаційних систем – сукупність правил, принципів, залежностей поведінки об'єктів предметної області (сфери людської діяльності, яку підтримує система). Інакше можна сказати, що бізнес-логіка – це реалізація правил і обмежень поведінки системи. Є синонімом терміна «логіка предметної області» (англ. Domain logic).

Простіше кажучи, бізнес-логіка – це реалізація предметної області в інформаційній системі. До неї відносяться, наприклад, формули розрахунку щомісячних виплат по позиках (у фінансовій індустрії), автоматизоване надсилання електронного листа керівнику проекту після закінчення виконання частин завдання всіма підлеглими (в системах управління проектами), відмова від готелю при скасуванні рейсу авіакомпанією (в туристичному бізнесі) і т.д.

#### 2.3.1. Технологія Servlets

*Java Servlet API* – стандартизований API для створення динамічного контенту до веб-сервера, використовуючи платформу Java. Сервлети – аналог технологій PHP, CGI і ASP.NET. Сервлет може зберігати інформацію між багатьма транзакціями, використовуючи HTTP куки, сесії або через редагування URL [4].

Зм	Лист	№ докум.	Підп.	Дата

**ІК-21.12 1448.01 ПЗ**

Аркуш

31

Servlet API, що міститься в пакеті javax.servlet, описує взаємодію веб-контейнера і сервлета. Веб-контейнер — це компонент веб-сервера, що створений для взаємодії з сервлетами. Він відповідає за управління життєвим циклом сервлетів, перетворення URL у певний сервлет та забезпечення того, щоб клієнт, який зробив URL запит, мав відповідні права доступу.

Сервлети, інтерфейси та базові класи, протоколи роботи з ними, робоче оточення, описуються у відповідних специфікаціях компанії Sun Microsystems.

Для полегшення розробки HTTP сервлетів, у специфікації описано абстрактний клас HttpServlet, від якого розробникам пропонується успадковувати свої сервлети.

#### Схема роботи

1. Клієнт (наприклад, веб-оглядач), відвідує веб-сторінку та надсилає HTTP запит на сервер.
2. Web-сервер отримує запит та передає його контейнеру сервлетів. Контейнер сервлетів може виконуватись в тому ж самому процесі, що і веб-сервер, в окремому процесі на тій же системі, що і веб-сервер, або взагалі в окремому процесі на іншій системі.
3. Контейнер сервлетів з'ясовує який сервлет слід викликати, виходячи з інформації про конфігурацію утримуваних сервлетів, та викликає його, передаючи в якості параметрів об'єктні представлення запиту та відповіді.
4. Сервлет використовує об'єкт запиту для отримання інформації про віддаленого користувача, параметри HTTP запиту тощо. Сервлет виконує запрограмовані в ньому дії та надсилає результати роботи через об'єкт відповіді.
5. Після того, як сервлет припиняє обробку запиту, контейнер сервлетів перевіряє коректність відправки відповіді, та повертає управління до головного веб-сервера.

Сервлети, також, використовуються в технології JSP. Шаблони сторінок транслуються у вихідні тексти Java-класів успадкованих від стандартних класів сервлетів. Java-компілятор компілює ці вихідні тексти в Java-байт коди. Отримані скомпільовані класи можуть використовуватись в сервлет-контейнері. Як правило, сервлет-контейнери виконують усі ці допоміжні дії автоматично [9].

*Переваги:*

- Виконуються швидше, ніж CGI-сценарії.
- Хороша масштабованість.
- Надійність і безпека (реалізовані на JAVA)
- Не залежать від платформи
- Безліч інструментів моніторингу та налагодження

*Недоліки:*

- Слабке поділ рівня уявлення і бізнес-логіки.
- Можливі конфлікти при паралельній обробці запитів.

### 2.3.2. Архітектурний шаблон MVC

*Model-view-controller*, MVC – архітектурний шаблон, який використовується під час проектування та розробки програмного забезпечення [1].

Мета шаблону — гнучкий дизайн програмного забезпечення, який повинен полегшувати подальші зміни чи розширення програм, а також надавати можливість повторного використання окремих компонентів програми. Крім того використання цього шаблону у великих системах призводить до певної впорядкованості їх структури і робить їх зрозумілішими завдяки зменшенню складності.

Архітектурний шаблон Модель-Вид-Контролер (MVC) поділяє програму на три частини. У тріаді до обов'язків компоненту Модель (Model) входить зберігання даних і забезпечення інтерфейсу до них. Вигляд (View)

Зм	Лист	№ докум.	Підп.	Дата

відповідальний за представлення цих даних користувачеві [5]. Контролер (Controller) керує компонентами, отримує сигнали у вигляді реакції на дії користувача, і повідомляє про зміни компоненту Модель. Така внутрішня структура в цілому поділяє систему на самостійні частини і розподіляє відповідальність між різними компонентами.

MVC поділяє цю частину системи на три самостійні частини: введення даних, компонент обробки даних і виведення інформації. Модель, як вже було відмічено, інкапсулює ядро даних і основний функціонал з їх обробки. Також компонент Модель не залежить від процесу введення або виведення даних. Компонент виводу Вигляд може мати декілька взаємопов'язаних областей, наприклад, різні таблиці і поля форм, в яких відображається інформація. У функції Контролера входить моніторинг за подіями, що виникають в результаті дій користувача (зміна положення курсора миші, натиснення кнопки або введення даних в текстове поле).

Зареєстровані події транслуються в різні запити, що спрямовуються компонентам Моделі або об'єктам, відповідальним за відображення даних. Відокремлення моделі від вигляду даних дозволяє незалежно використовувати різні компоненти для відображення інформації. Таким чином, якщо користувач через Контролер внесе зміни до Моделі даних, то інформація, подана одним або декількома візуальними компонентами, буде автоматично відкоригована відповідно до змін, що відбулися.

*Переваги:*

- Відділення бізнес-логіки від призначеного для користувача інтерфейсу і роботи з базою даних;
- Дозволяє розташовувати різні частини в різних місцях, що сприятливо позначається на підтримці і обслуговуванні;
- Тестованість;
- Інтуїтивна зрозумілість;
- Підтримка.

Зм	Лист	№ докум.	Підп.	Дата

*Недоліки:*

- Складність розробки.

#### 2.4. Обґрунтування вибору засобів розробки

Дана система адміністрування спортивними залами розроблялась для студмістечка НТУУ «КПІ», тому засоби розробки обиралися відповідно до вимог адміністраторів спортивних залів НТУУ «КПІ», актуальності сучасних технологій, цінових політик сучасних фреймворків та ПО в цілому.

Мовою написання програмного продукту було вибрано Java з огляду на зручність використання єдиної мови програмування на всіх рівнях програмного продукту. На стороні back-end використовується фреймворк hibernate для взаємодії з БД, даний фреймворк написаний для мови програмування Java та має багато плагінів для прискорення розробки продукту, на стороні front-end використовується технологія jsp що взаємодіючи з проміжними слоєм продукту побудованим за допомогою servlets показує гарні показники надійності та продуктивності.

Фреймворк hibernate був обраний з огляду на те що система адміністрування спортивними залами має не складну базу даних, з чого випливає те що продуктивність запитів до даної бази мають гарні показники ефективності та надійності. Використовуючи даний фреймворк немає необхідності писати рутинні запити до бази даних, всі залежності між таблицями та їх обмеження легко переносяться на java код, даний фреймворк забезпечує високий ступінь незалежності від конкретної СУБД. Також даний фреймворк дуже зручний у взаємодії з MySQL, яку було вибрано як сховище даних через простоту у встановленні та використанні, підтримку необмеженої кількості користувачів, що одночасно працюють із БД, безкоштовність, високу швидкість виконання різноманітних запитів та команд, наявність простої та ефективної системи безпеки.

Для створення front-end частини проекту була задіяна технологія jsp з огляду на те що вона має низьку вартість підтримки проектів, розроблених за

Зм	Лист	№ докум.	Підп.	Дата

**ІК-21.12 1448.01 ПЗ**

Аркуш

35

даною технологією, являється кросплатформеною технологією, дуже широко поширена на багатьох існуючих проектах, має відкритий вихідний код, за даною технологією можна знайти велику кількість готових рішень та бібліотек.

Дуже важливою складовою розробки даного додатку є бізнес логіка, що реалізована за допомогою Java Servlet. Сервлет є інтерфейсом Java, реалізація якого розширює функціональні можливості сервера. Сервлет взаємодіє з клієнтами за допомогою принципу запит-відповідь. Хоча сервлети можуть обслуговувати будь-які запити, вони зазвичай використовуються для розширення веб-серверів. Для таких додатків технологія Java Servlet визначає HTTP-специфічні сервлет класи.

Перша специфікація сервлетів була створена в Sun Microsystems (версія 1.0 була закінчена в червні 1997). Починаючи з версії 2.3, специфікація сервлетів розроблялася під керівництвом Java Community Process. Стандарт JSR 53 визначав як Servlet 2.3, так і специфікацію JavaServer Page 1.2. JSR 154 включає в себе специфікації Servlet 2.4 і 2.5. Поточна специфікація на 13 червня 2013 року - Servlet 3.1 (описана в JSR-340). Великою перевагою в контексті нашого додатку технології Java Servlet є швидкість роботи, гарна масштабованість, надійність і безпека реалізовані саме на Java, незалежність від платформи, безліч інструментів моніторингу та налагодження і легка інтегрованість back-end and front-end частини.

Також для створення даного проекту було застосовано фреймворк Bootstrap, основні переваги якого в даному контексті є простота, безкоштовність, інтегрованість.

### Висновки по розділу

В даному розділі розглянуто і проаналізовано способи та технології створення різних частин додатку, наведено опис і характеристики мови програмування та фреймворків для створення системи адміністрування спортивними залами, їх основні переваги та недоліки.

## РОЗДІЛ 3. РОЗРОБКА АЛГОРИТМІЧНОГО ТА ПРОГРАМНОГО ЗАБЕЗПЕЧЕННЯ

### 3.1. Архітектура програмного забезпечення

Архітектура системи адміністрування спортивними залами показана на рис. 3.1.

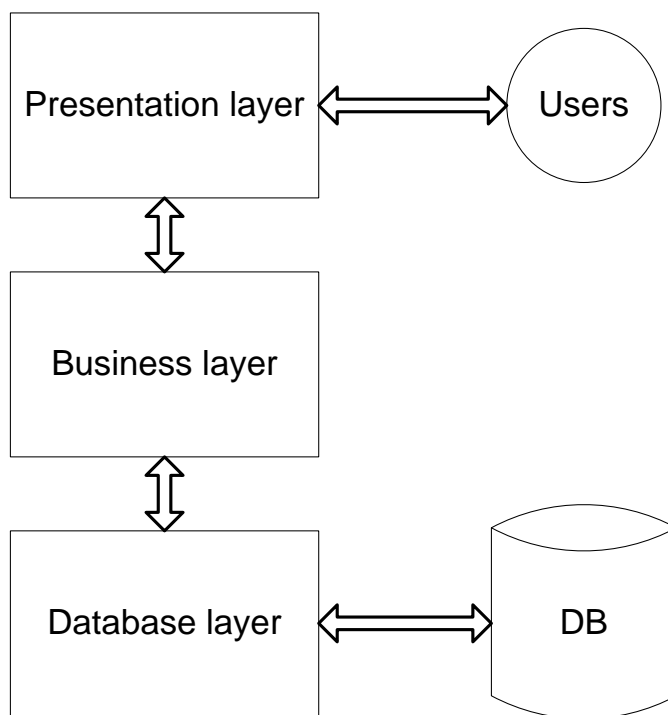


Рисунок 3.1. Архітектура програмного забезпечення додатку

Система адміністрування спортивною залом побудована на основі трьохрівневої архітектури (3-tier): database-layer, business layer, presentation layer;

Рівень даних (database-layer) включає в себе механізми збереження даних (сервера баз-даних, СУБД, файлові системи) та надає доступ до цієї інформації (API для business-layer, що логічно зв'язує db-layer and presentation-layer). Даний рівень створений на основі БД та СУБД MySQL, а також найпоширенішого фреймворку для взаємодії з БД – hibernate, що створений саме для мови програмування Java (на якій створено сам додаток) та має багато переваг, що описані в попередньому розділі.



Рівень бізнес-логіки задає поведінку об'єктів та способи взаємодії даних об'єктів один з одним, надає API для оплати послуг, слідкує за консистентністю даних та їх актуальністю, задає послідовність дій користувача та адміністратора.

Схема послідовності взаємодії юзерів із системою адміністрування спортивною залом наведена у додатку графічного матеріалу. На ній зображена типова послідовність дій, які необхідні для адміністрування спортивною залом, оплати послуг як готівковим так і безготівковим способом, перегляду статистичних даних та реєстрування нових юзерів.

Рівень відображення даних (presentation-layer) відповідає за надання зручного інтерфейсу для взаємодії з додатком. Інтерфейс користувача відображається за допомогою мови розмітки HTML та таблиць стилів CSS, також використовується сучасний фреймворк Bootstrap.

Безпосередньо за відображення даного інтерфейсу відповідає конкретний браузер користувача. Використання такого типу взаємодії з додатком надає можливість керування системою з будь-якого пристрою який має доступ до мережі інтернет та який має встановлений браузер. Це можуть бути: персональний комп'ютер (ПК), смартфон, планшет і т.д.

### 3.2. Опис структури бази даних

У табл. 3.1-3.4 наведена структура таблиць бази даних, схема бази даних наведена у додатку графічного матеріалу.

Таблиця 3.1. Опис таблиці billinformation

Назва стовпця	Опис	Тип даних	Довжина	Первинний	Зовнішній
<i>id</i>	ID рахунку	<i>INT</i>	11	X	

Закінчення таблиці 3.1.

<i>cardNumber</i>	Номер рахунку	<i>VACRCHAR</i>	19		
<i>price</i>	Ціна послуги	<i>DECIMAL</i>	10,2		
<i>isSelected</i>	Вибрана карта	<i>TINYINT</i>	1		

Таблиця 3.2. Опис таблиці costs

Назва стовпця	Опис	Тип даних	Довжина	Первинний	Зовнішній
<i>id</i>	ID витрат	<i>INT</i>	11	X	
<i>userId</i>	ID адміністратора	<i>INT</i>	11		X
<i>costDate</i>	Дата витрати	<i>DATE</i>			
<i>name</i>	Назва витрати(опис)	<i>VARCHAR</i>	400		
<i>price</i>	Ціна витрати	<i>DECIMAL</i>	10,2		

Таблиця 3.3. Опис таблиці payments

Назва стовпця	Опис	Тип даних	Довжина	Первинний	Зовнішній
<i>id</i>	ID платежів	<i>INT</i>	11	X	
<i>userId</i>	ID користувачів	<i>INT</i>	11		X
<i>expense</i>	Ціна послуги	<i>DECIMAL</i>	10,2		
<i>createDate</i>	Дата платежу	<i>DATETIME</i>			
<i>paymentCard</i>	Картка платежу	<i>VARCHAR</i>	19		
<i>paymentMonth</i>	Місяць дії абон-ту	<i>VARCHAR</i>	15		
<i>paymentYear</i>	Рік дії абон-ту	<i>INT</i>	11		
<i>paymentType</i>	Тип платежу	<i>ENUM('CASH', 'BANK')</i>			

Таблиця 3.4. Опис таблиці users

Назва стовпця	Опис	Тип даних	Довжина	Первинний	Зовнішній
<i>id</i>	ID користувача	<i>INT</i>	11	X	
<i>name</i>	Ім'я юзера	<i>VARCHAR</i>	35		
<i>surname</i>	Прізвище	<i>VARCHAR</i>	35		
<i>patronymic</i>	По-батькові	<i>VARCHAR</i>	35		
<i>email</i>	Пошта юзера	<i>VARCHAR</i>	100		
<i>password</i>	Пароль юзера	<i>VARCHAR</i>	50		
<i>dormitoryNumber</i>	Номер гуртоку	<i>INT</i>	11		
<i>roomNumber</i>	Номер кімнати	<i>INT</i>	11		
<i>registrationDate</i>	Дата реєстрації	<i>DATETIME</i>			
<i>sex</i>	Стать	<i>enum('MALE', 'FEMALE')</i>			
<i>roleId</i>	id Ролі	<i>INT</i>	11		X

Таблиця 3.5. Опис таблиці roles

Назва стовпця	Опис	Тип даних	Довжина	Первинний	Зовнішній
<i>id</i>	ID ролі	<i>INT</i>	11	X	
<i>description</i>	Опис ролі (тип)	<i>VARCHAR</i>	5		

### 3.3. Вимоги до технічного забезпечення

#### Вимоги до програмного забезпечення

*Вимоги до програмного забезпечення серверної частини додатку:*

- БД MySQL 5.6 або вище;
- Веб-сервер Apache tomcat 7;
- Операційна система Windows XP, Vista, Linux, Mac OS, and others;

*Вимоги до програмного забезпечення клієнта:*

- Браузер IE 10 або вище, Firefox, Chrome, Opera, Safari;
- Включений JS у браузері.

Вимоги до технічного забезпечення серверної частини додатку:

- процесор – 1.6 ГГц, 1 ядро ЦП або краще;
- оперативна пам'ять не менш ніж 1024 Мб;
- не менше ніж 35 Гб ПЗУ;
- 10 Мбіт/сек доступу до мережі Інтернет.

### 3.4. Керівництво користувача

Для переходу до системи адміністрування спортивною залою необхідно відкрити браузер, який задовольняє вимогам наведеним у п. 3.3 «Вимоги до технічного забезпечення» та перейти на адресу веб-додатку (він залежить від того, яке доменне ім'я буде надано йому).

1. Сторінка входу. Дана сторінка дозволяє зайти до системи адміністрування та вибрати мову відображення сторінок всього додатку.

Зм	Лист	№ докум.	Підп.	Дата

**ІК-21.12 1448.01 ПЗ**

Аркуш

41

Рисунок 3.1. Логін-форма

2. Сторінка користувача. Дана сторінка дозволяє користувачеві вибрати місяці на які він хоче придбати абонемент на поточний рік, переглянути свою інформацію, переглянути свою статистику витрат за весь час, що включає детальну інформацію, вказати з якої карти він бажає оплатити послуги. При правильній чи не правильній оплаті з'являється відповідне повідомлення.

Ваша інформація		Місяці до оплати:	
#	3	Січень	Лютий
Прізвище	Опашнянський	Березень	Квітень
Ім'я	Роман	Травень	Червень
По батькові	Сергеевич	Липень	Серпень
Поштова адреса	user	Вересень	Жовтень
Гуртожиток №	6	Листопад	Грудень
Кімната №	504	1233 1233 1233 1233	
Зареєстрований	2016-04-19	Заплатити!	
Стать	MALE	Платіж здійснено успішно!	
Ціна			
	30.0		
Платежі користувачів			
Витрати	Дата платежу	Платіжна карта	Рік
30.0	2016-04-19		2016
			April
			CASH

Рисунок 3.2. Форма користувача

3. Головна сторінка адміністратора. Дозволяє вибрати завдяки меню ті дії, які необхідні адміністратору в поточний час

Ваша інформація		Меню адміністратора	
#	1	<a href="#">Платежі користувачів</a>	
Прізвище	Клейменов	<a href="#">Додати користувача</a>	
Ім'я	Роман	<a href="#">Оновити інформацію користувача</a>	
По батькові	Сергеевич	<a href="#">Аналітика</a>	
Поштова адреса	admin	<a href="#">Витрати</a>	
Гуртожиток №	6	<a href="#">Змінити ціну</a>	
Кімната №	504		
Зареєстрований	2016-04-19		
Стать	MALE		

Поточна ціна	
№ карти	1111 2222 3333 4444
Ціна	30.0

Рисунок 3.3. Головна сторінка адміністратора

4. Сторінка пошуку клієнтів. Багато задач адміністратора потребують сторінку пошуку користувачів яка дозволяє проводити пошук по ID, імені, прізвищу чи по імені-прізвищу.

Logout

Користувачі										
#	Ім'я	Прізвище	По батькові	Поштова адреса	Гуртожиток №	Кімната №	Зареєстрований	Роль	Стать	Заплатити
1	Роман	Клейменов	Сергеевич	admin	6	504	2016-04-19	ADMIN	MALE	<input type="button" value="Вибрати"/>
3	Роман	Опашнянский	Сергеевич	user	6	504	2016-04-19	USER	MALE	<input type="button" value="Вибрати"/>

Рисунок 3.4. Сторінка пошуку клієнтів

5. Сторінка оплати послуг готівкою. Дана сторінка адміністратора дозволяє додавати платежі готівкою до єдиної БД. А також видаляти помилкові платежі.

Logout

### Ваша інформація

#	3
Прізвище	Опашнянський
Ім'я	Роман
По батькові	Сергеевич
Поштова адреса	user
Гуртожиток №	6
Кімната №	504
Зареєстрований	2016-04-19
Стать	MALE

### Місяці до оплати

Січень   Лютий   Березень   Квітень   Травень   Червень   Липень   Серпень   Вересень   Жовтень   Листопад   Грудень

☐ ☐ ☐ ☐ ☐ ☐ ☐ ☐ ☐ ☐ ☐ ☐

Заплатити

Платіж здійснено успішно!

### Платежі користувачів

Витрати	Дата платежу	Платіжна карта	Рік	Місяць	Тип	Видалити
30.0	2016-04-19		2016	April	CASH	Видалити
60.0	2016-04-19	1111 1111 1111 1111	2016	December	BANK	Видалити
30.0	2016-04-19		2016	June	CASH	Видалити

Рисунок 3.5. Сторінка оплати послуг готівкою

## 6. Сторінка редагування то додавання юзерів.

Прізвище	Клейменов
Ім'я	Роман
По батькові	Сергеевич
Поштова адреса	rs.kleymenov@gmail.com
Пароль	пароль
Гуртожиток №	5
Кімната №	5
Роль	USER
Стать	MALE
Прийняти	

Рисунок 3.6. Сторінка редагування то додавання юзерів

## 7. Сторінка аналітики. Дана сторінка надає звіти помісячних платежів, платежів за видом грошей та платежів по рокам.

Logout

Рік

2016

Розрахувати статистику!

Платежі помісячно

Витрати	Місяць
90.0	April
90.0	December
30.0	July
30.0	June
30.0	March
60.0	May
30.0	October

Платежі за видом грошей

Витрати	Вид грошей
150.0	CASH

Рисунок 3.7. Сторінка аналітики

8. Сторінка витрат. На даній сторінці адміністратор має можливість додати нову витрату щодо поліпшення устаткування спортивного залу або переглянути статистику щодо минулих витрат.

Додати витрату

Витрати

Додав	Дата	Опис витрати	Ціна	Видалити
Клейменов Роман Сергеевич	2016-04-19	Гриф	3000.0	<input type="button" value="Видалити"/>
Клейменов Роман Сергеевич	2016-04-19	Два блина по 20 кг	600.0	<input type="button" value="Видалити"/>
Клейменов Роман Сергеевич	2015-03-12	Chacha	1000.0	<input type="button" value="Видалити"/>

На сторінку адміністратора

Рисунок 3.8. Сторінка витрат

9. Сторінка рахунку. На даній сторінці адміністратор може змінити поточний рахунок, на який проводяться платежі від користувачів.



### Додати карту

### Картки

№ карти	Ціна	Статус	Видалити	Вибрати
1111 2222 3333 4444	30.0	true	<input type="button" value="Видалити"/>	<input type="button" value="Вибрати"/>
5555 6666 7777 8888	30.0	false	<input type="button" value="Видалити"/>	<input type="button" value="Вибрати"/>
5555 6666 7772 3213	20.0	false	<input type="button" value="Видалити"/>	<input type="button" value="Вибрати"/>

Рисунок 3.9. Сторінка рахунку

### Висновки по розділу

В третьому розділі описано етап розробки алгоритмічного та програмного забезпечення системи адміністрування спортивною залюю. Наведено архітектуру програмного продукту та технічні вимоги, описано керівництво користувача й структуру бази даних.

## РОЗДІЛ 4. ОХОРОНА ПРАЦІ

Охорона праці – це система законодавчих, гігієнічних і організаційних заходів спрямованих на збереження здоров'я і працездатності людини в процесі праці. Основними складовими частинами охорони праці являється трудове законодавство, техніка безпеки і виробнича санітарія, нерозривно зв'язані між собою. Трудове законодавство регламентує питання трудового права, санітарно-гігієнічних та технічних заходів, виробнича санітарія направлена на забезпечення здорових умов праці. У даному розділі проведено аналіз факторів, які впливають на продуктивність праці під час використання розробленої системи. Розглядається приміщення, в якому велась розробка системи розглянутої в даній роботі.

Працюючи з комп'ютером, людина потрапляє під вплив різноманітних факторів: електромагнітних полів, інфрачервоного та іонізуючого випромінювання, шуму і вібрацій, статичної електрики.

В процесі роботи з ПК необхідно дотримуватися правильного режиму праці та відпочинку. В іншому випадку у персоналу з'являється значна напруга зорового апарату з появою скарг на незадоволеність роботою, головний біль, дратівливість, порушення сну, а також втома і хворобливі відчуття в очах, попереку, області шиї і руках.

В розділі розглядаються питання безпеки робочого місця, освітлення, пожежної безпеки, мікроклімату, шуму та електробезпеки.

### 4.1. Аналіз умов праці

Приміщення офісного типу знаходиться на першому поверсі дванадцятиповерхового будинку. Схему приміщення, в якому проводиться дослідження умов праці, зображено на рис. 4.1. Кожне робоче місце обладнане робочим столом кутової форми, кріслом, комп'ютером.

Розміри приміщення:

- довжина (А) - 6,0 м

Зм	Лист	№ докум.	Підп.	Дата

**ІК-21.12 1448.01 ПЗ**

Аркуш

47

- ширина (В) - 4,6 м
- висота (Н) - 3,0 м.

Загальна площа приміщення складає 27,6 м<sup>2</sup>. Об'єм приміщення складає 82,8 м<sup>3</sup>. Загальна кількість робочих місць – 4.

Згідно НПАОП 0.00-1.28-10 на одного працюючого об'єм приміщення повинен складати не менше 20 м<sup>3</sup>, площа – не менше 6 м<sup>2</sup>. Для даного приміщення з урахуванням розмірів на одного працюючого об'єм приміщення складає не менше 20,7 м<sup>3</sup>, площа – не менше 6,9 м<sup>2</sup>, що задовольняє встановленим нормам.

Під вікнами знаходяться батареї системи опалення. Підлога в приміщенні покрита комерційним лінолеумом, стіни обклеєні білими шпалерами, розміщення меблів не заважає природному освітленню.

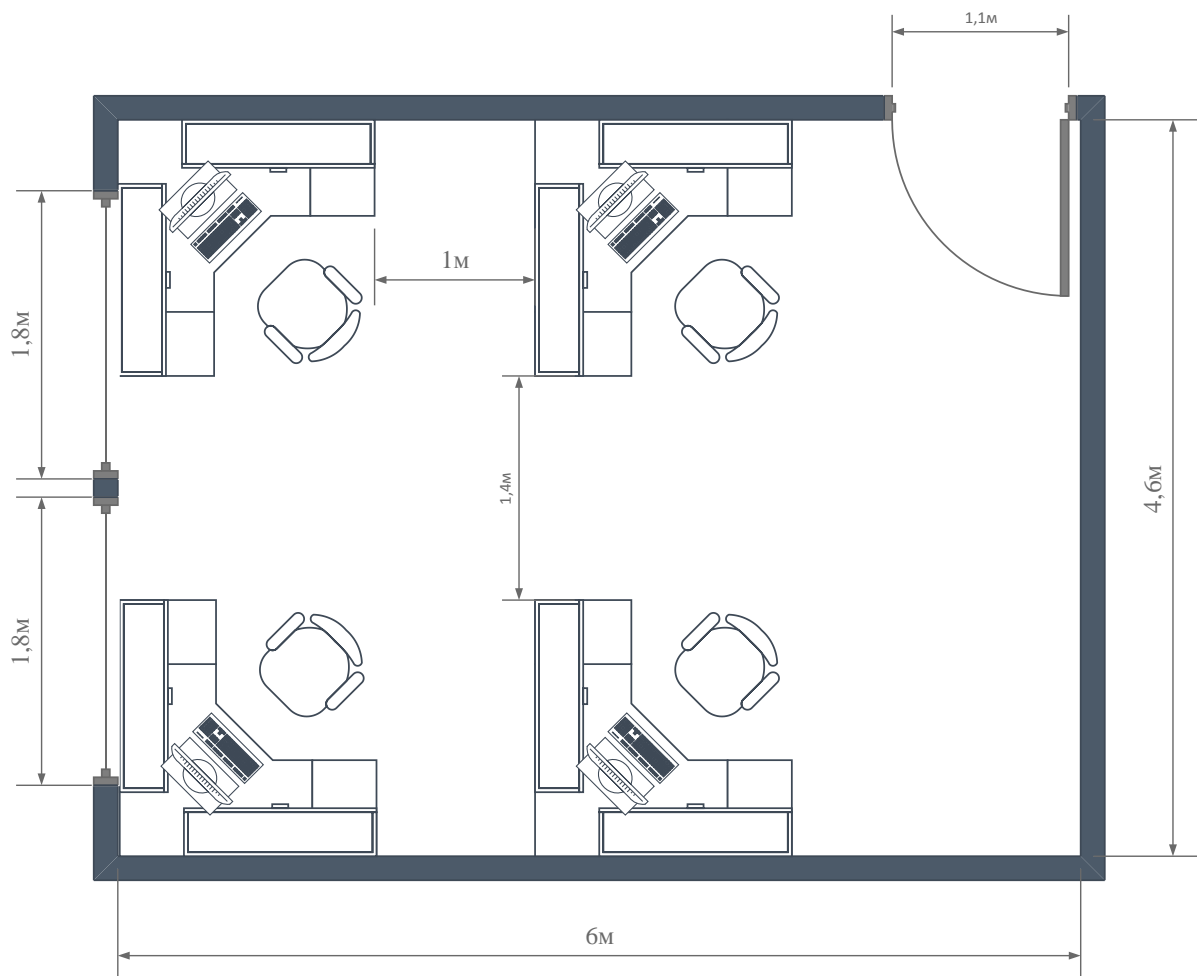


Рис. 4.1. План приміщення

Зм	Лист	№ докум.	Підп.	Дата

**ІК-21.12 1448.01 ПЗ**

Аркуш

48

## 4.2. Аналіз шкідливих і небезпечних факторів

### 4.2.1. Аналіз рівня освітлення

В приміщенні 2 вікна (1,8\*2,0 м) площею 3,6 м<sup>2</sup>, які орієнтовані на східний бік.

Необхідний рівень освітлення на робочому місті визначають в залежності від характеристик зорових робіт. Роботи за ПЕОМ і оформлення документації, відносяться до III розряду зорової роботи (високої точності) для цих робіт згідно ДБН В.2.5-28-2006 нормоване значення освітленості становить 400 лк [12].

Природного освітлення не достатньо для забезпечення нормованого значення освітленості 300-500 лк, тому необхідно використовувати штучне освітлення.

Розрахуємо штучне освітлення в приміщенні. Його розрахунок проводиться за методом світлового потоку, з урахуванням потоків відображених від стелі і стін.

Згідно ДБН В.2.5-28-2006 «Природне і штучне освітлення» для даних робіт встановлена необхідна освітленість робочого місця  $E_n = 400$  лк.

Загальний світловий потік визначається за формулою:

$$\Phi_{\text{заг}} = \frac{E_n \cdot S \cdot k_1 \cdot k_2}{V}$$

де  $E_n$  – нормована освітленість;

$S$  – площа застосування;

$k_1 = 1,5$  – коефіцієнт, що враховує старіння ламп і забруднення світильників;

$k_2 = 1,1$  – коефіцієнт, що враховує нерівномірність освітлення приміщення;

$V$  – коефіцієнт використання світлового потоку, який визначається відповідно до коефіцієнтів віддзеркалення стін, робочих поверхонь, стелі, геометрії приміщення і типів світильників  $V=0,7$ .

Розміри приміщення складають  $A=6$  м,  $B=4.6$  м,  $H=3$  м.

$$S = A \cdot B = 6 \cdot 4.6 = 27.6 \text{ м}^2$$

Оберемо з таблиці коефіцієнти використання світлового потоку:

- коефіцієнт віддзеркалення від побіленої стелі  $R_{\text{стелі}} = 70\%$  ;
- коефіцієнт віддзеркалення від стін білого кольору  $R_{\text{стін}} = 50\%$
- коефіцієнт віддзеркалення від темного лінолеуму  $R_{\text{підлоги}} = 10\%$
- індекс приміщення:

$$i = \frac{A \cdot B}{H \cdot (A + B)} = \frac{A \cdot B}{h_p \cdot (A + B)}$$

$h_p = H - h_n$ , де  $h_n = 0,7$  м – висота робочої поверхні над підлогою.

$$h_p = 3 - 0,7 = 2.3 \text{ м}$$

За таблицею визначимо, що  $\eta = 0.41$ .

Визначимо показник приміщення:

$$i = \frac{6 \cdot 4.6}{2.3 \cdot (6 + 4.6)} \cong 1.132$$

Тепер визначимо значення загального світлового потоку:

$$\Phi_{\text{заг}} = \frac{400 \cdot 27.6 \cdot 1.5 \cdot 1.1}{0.7} = 26022 \text{ лм}$$

Для забезпечення загального штучного освітлення обираємо світильник з двома люмінесцентними лампами типу ЛД-80 .

Світловий потік однієї лампи ЛД-80 складає  $\Phi_{\text{л}} = 4250$

Тепер визначимо кількість ламп, необхідну для освітлення приміщення:

$$N = \frac{\Phi_{\text{заг}}}{\Phi_{\text{л}}} = \frac{26022}{4250} = 7.22 = 8$$

Таким чином, щоб забезпечити світловий потік  $\Phi_{\text{заг}} = 26022$  лм треба використовувати 8 ламп, тобто 4 світильники. Розміщуємо світильники у 2 ряди.

План розміщення світильників зображено на рис 4.2.

Фактична освітленість обчислюється за формулою:

Зм	Лист	№ докум.	Підп.	Дата

**ІК-21.12 1448.01 ПЗ**

Аркуш

50

$$E_{\phi} = \frac{F_{\lambda} \cdot n \cdot N \cdot \eta}{S \cdot k_1 \cdot k_2} = \frac{4250 \cdot 2 \cdot 4 \cdot 0.41}{27.6 \cdot 1.5 \cdot 1.1} = 316.1 \text{ (лк)}$$

Фактична освітленість задовольняє норму у 300-500 лк, отже розрахунки проведено правильно.

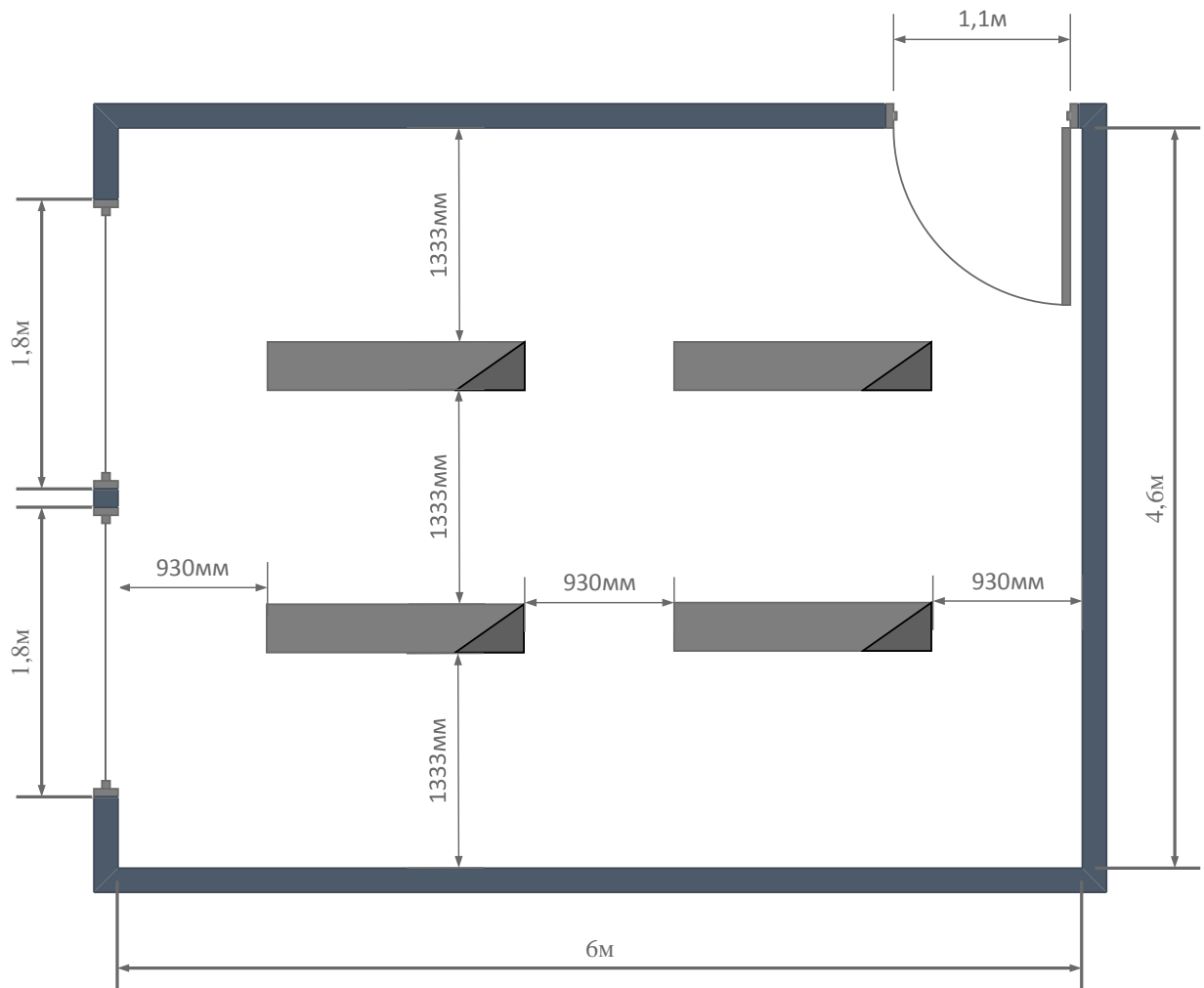


Рис. 4.2. – План розміщення ламп

#### 4.2.2. Аналіз мікроклімату приміщення

Робота, яка проводиться в приміщенні - сидяча, не пов'язана з фізичною напругою. Відповідно до ДСН 3.3.6.042-99 така робота відноситься до категорії «Легка 1а» [11].

Оскільки характер роботи постійний, розглянемо оптимальні значення характеристик мікроклімату, що наведені у табл. 4.1, де приведені також фактичні значення цих параметрів.

Зм	Лист	№ докум.	Підп.	Дата

**ІК-21.12 1448.01 ПЗ**

Аркуш

51

Таблиця 4.1 - Оптимальні і фактичні значення параметрів мікроклімату

Найменування	Фактичне значення	Оптимальне значення
Температура повітря у теплий період року, °C	23 - 24	23 - 25
Температура повітря у холодний період, °C	22 - 23	22 - 24
Відносна вологість повітря, %	40 - 50	40 - 60
Швидкість руху повітря, м/с	0.1	0.1

У приміщенні використовується центральне парове опалення, що включається в холодний період року. У теплий період для підтримки сприятливих умов праці використовується кондиціонер спліт-системи SAMSUNG X23F6H з потужністю 2,6 кВт.

Для нормалізації параметрів повітряного середовища здійснюється регулярні провітрювання приміщення і вологе прибирання.

Всі параметри мікроклімату задовольняють нормам.

#### 4.2.3. Аналіз рівня шуму

Можливими джерелами шуму в даному приміщенні є компютери, кондиціонер, принтер та сканер.

Таблиця 4.2. Значення рівня шуму для типових джерел

Джерелами шуму	Рівень шуму, дБл	Час роботи
Комп'ютер (4 шт)	30	8
Кондиціонер	40	8
Принтер	25	2
Сканер	20	0,5

Згідно з ДСН 3.3.6.037-99 для першого виду діяльності еквівалентний рівень шуму не повинен перевищувати 50 дБА.

Еквівалентний рівень звуку обчислюється за формулою:

$$L_{\text{екв}} = 10 \lg \frac{1}{T} \sum_j^N t_i \cdot 10^{0.1 \cdot L_i}$$

де  $L_i$  – рівень звуку  $i$ -того джерела;  $t_i$  – час роботи  $i$ -того джерела, а  $T$  – час роботи.

Робочий день  $T = 8$  годин.

$$L_{\text{екв}} = 10 \lg \left[ \left( \frac{1}{8} \right) (8000 + 8000 + 8000 + 8000 + 80000 + 50 + 632.456) \right] = 41.48$$

Відповідно еквівалентний рівень за формулою  $L_{\text{екв}} = 41.48$  дБА, що відповідає нормі.

#### 4.2.4. Аналіз рівня випромінювання

На робочому місці мають бути встановлені сучасні монітори, характеристики яких відповідають нормам НПОПО 0.00-1.128-10: припустима інтенсивність потоку енергії - 10Вт/м<sup>2</sup>, напруженість електромагнітного поля по електричній складовій на відстані 50 см від екрана – 10 В/м.

Усе обладнання, що використовується в даному приміщенні, відповідає нормам, щодо електромагнітного випромінювання.

#### 4.3. Електробезпека

Приміщення відноситься до приміщень без підвищеної небезпеки.

Приміщення живиться електричною енергією від централізованого трансформатора, який знаходиться на поверсі поза межами приміщення і понижує напругу до 220 В, змінним струмом частотою 50 Гц. Комп'ютери мають власні блоки живлення, що перетворюють змінний струм мережі у постійний (5 В, 12 В). Потужність даних блоків живлення – 400 – 450 Вт.

ПК у приміщенні відносяться до категорії I захисту від ураження електричним струмом. Усі прилади при підключенні до електричної мережі заземляються завдяки спеціальним контактам у вилці. Корпуса всіх

Зм	Лист	№ докум.	Підп.	Дата

**ІК-21.12 1448.01 ПЗ**

Аркуш

53



електричних пристроїв виготовляються з неструмопровідних матеріалів, а живлення здійснюється спеціальним кабелем.

Таким чином, вимоги ДНАОП 0.00-1.31-99 виконано [14].

#### 4.4. Пожежна безпека

У досліджуваному приміщенні є наявності тільки тверді і волокнисті пальні речовини: дерево, папір, тканина. Згідно НАПБ В.03.002-2007 приміщення відноситься до категорії В по пожежонебезпеці, а робоча зона приміщення згідно з першою категорією надійності (ПУЕ) належить до класу П-Па [12].

Можливими причинами пожежі в приміщенні є несправність електроустаткування і порушення протипожежного режиму (використання побутових нагрівальних приладів, паління).

Відповідно до ДНАОП 0.00-1.31-99 у приміщенні буде встановлено 3 вогнегасники типу ВВ-2 ( по 2 на кожні 20 м<sup>2</sup> ) для тушіння невеликих джерел займання і устаткування під напругою до 1000 В. Відстань між місцями розташування вогнегасників не повинна перевищувати 15 м [15].

Як сигналізатор виникнення пожежі прийнято тепловий сповіщувач типу ІП-105. Сповіщувач встановлюються на стелі. При спрацюванні сповіщувача сигнал надходить до контрольно-приймального приладу, який розташований у приміщенні чергового.

Двері відкриваються всередину, ширина дверей не менше 1,2 м, а висота проходу 2 м. План евакуації буде знаходитися біля дверей.

Таким чином усі фактори пожежної безпеки задовольняють вимогам встановлених норм.

Інструкція з техніки безпеки

*Перед початком роботи*

- Пересвідчитися у цілості корпусів і блоків (обладнання) ПК;

Зм	Лист	№ докум.	Підп.	Дата

**ІК-21.12 1448.01 ПЗ**

Аркуш

54

- Перевірити наявність заземлення, справність і цілість кабелів живлення, місця їх підключення;
- Очистити екран від забруднень.

*Під час роботи за персональним комп'ютером забороняється:*

- Вішати будь-які предмети на дроти.
- Витягувати штепсельну вилку із розетки, тримаючись за шнур.
- Часто вмикати і вимикати персональний комп'ютер без потреби.
- Торкатися екрану та тильної сторони комп'ютера.
- Працювати з мокрими руками.
- Класти на обладнання сторонні предмети.

*По закінченні робочого дня:*

- відключити електроживлення ПК згідно з інструкцією експлуатації, вийнявши вилку кабелю живлення з розетки;
- впорядкувати робоче місце користувача ПК, прибравши використане обладнання та матеріали у відведені місця;
- перевірити стан апаратури, справність електропроводки, штепсельних вилок, розеток.

Залишаючи приміщення після закінчення робочого дня, дотримуючись встановленого режиму огляду приміщення, необхідно:

- зачинити вікна;
- відключити від електромережі всі електроприлади, електрообладнання та вимкнути освітлення;
- зачинити входні двері приміщення на замок і ключ здати черговому охорони.

Висновки по розділу

У ході розробки розділу було вивчено приміщення в якому розроблювалося програмне забезпечення. Було описано та проаналізовано небезпечні й шкідливі виробничі фактори у виробничому приміщенні з

радіоелектронною апаратурою. У результаті проведення робіт встановлено, що об'єм і площа приміщення, які відводяться на одного працівника, відповідають нормативним значенням. Для цих факторів визначено нормативні показники у відповідності з діючою нормативно-технічною документацією та проведено порівняння з фактичними їх значеннями. З виявлених небезпечних та шкідливих факторів виділено найбільш несприятливі для яких було розроблено детальні заходи з охорони праці. Аналіз усіх розрахованих у даному розділі факторів показав результати, які дають всі підстави вважати, що розглянуте виробниче приміщення повністю відповідає всім нормативним документам і вимогам.

## ЗАГАЛЬНІ ВИСНОВКИ

У ході виконання дипломного проекту було розглянуто питання пов'язані з необхідністю та актуальністю створення системи адміністрування спортивними залами. Наведено характеристику предметного середовища та обґрунтовано причину розробки системи. Описано вимоги до сервісу даного виду та технології створення систем адміністрування. Розроблено власну систему адміністрування спортивними залами.

На основі даних, отриманих в процесі аналізу, сформульовано задачу створення системи адміністрування спортивними залами у вигляді веб-додатку.

Для розробки системи використано мову програмування Java, веб-фреймворк Bootstrap, IDE Eclipse Mars, технології JSP та Servlet with MVC – все перераховане є широко вживаними та безкоштовними інструментами, що дозволить легко розширити функціонал системи в разі виникнення такої потреби.

Розроблена модель бази даних, яка дає змогу ефективно та надійно здійснювати доступ до даних, що надходять та використовуються в процесі взаємодії з веб-додатком. Для управління базою даних була обрана СУБД MySQL 5.0 та фреймворк Hibernate.

Розроблена та наведена інструкція користувача по експлуатації системи адміністрування спортивними залами.

Розглянуто з точки зору охорони праці робоче приміщення, де проводилася розробка системи адміністрування спортивними залами.

Результати роботи пройшли апробацію на міжнародній науково-практичній конференції «Summer InfoCom Advanced Solutions 2016» [16].

Зм	Лист	№ докум.	Підп.	Дата

***ІК-21.12 1448.01 ПЗ***

Аркуш

57

## ПЕРЕЛІК ПОСИЛАНЬ

1. Martin Fowler, David Rice, Matthew Foemmel, Edward Hieatt, Robert Mee, Randy Stafford. "Patterns of Enterprise Application Architecture" Addison Wesley (November 05, 2002) – 560 pages
2. Mike Keith, Merric Schincariol. "Pro JPA 2" Apress (2009) – 538 pages
3. Hibernate [Електронний ресурс] // Режим доступу: <http://hibernate.org/>
4. Servlets [Електронний ресурс] // Режим доступу: <http://www.oracle.com/technetwork/java/index-jsp-135475.html>
5. Шаблон проектування MVC [Електронний ресурс] // Режим доступу: <https://ru.wikipedia.org/wiki/Model-View-Controller>
6. JSP сторінки [Електронний ресурс] // Режим доступу: <http://docs.oracle.com/javaee/5/tutorial/doc/bnajo.html>
7. HTML [Електронний ресурс] // Режим доступу: <http://htmlbook.ru/html/!doctype>
8. Мова програмування Java [Електронний ресурс] // Режим доступу: <https://ru.wikipedia.org/wiki/Java>
9. Создание приложений реального времени с помощью Server-Sent Events [Електронний ресурс] // Режим доступу: <http://habrahabr.ru/post/120429/>
10. ДСанПіН 3.3.2.007-98 Державні санітарні правила і норми роботи з візуальними дисплейними терміналами електронно-обчислювальних машин ЕОМ.
11. ДСН 3.3.6.042-99 Санітарні норми мікроклімату виробничих приміщень.
12. ДБН В.2.5-28-2006 Природне і штучне освітлення
13. ПУЕ Правила улаштування електроустановок
14. ГОСТ 12.1.038.82 Электробезопасность. Предельно допустимые значения напряжений прикосновения и токов
15. ГОСТ 12.1.004-91.ССБТ Пожежна безпека
16. Summer InfoCom 2016: Матеріали II Міжнародної науково-практичної конференції, м. Київ, 1-3 червня 2016 р. – К.: Вид-во «Інжиніринг», 2016. – 116 с. – Мови укр., рос., англ.

Зм	Лист	№ докум.	Підп.	Дата

**ІК-21.12 1448.01 ПЗ**

Аркуш

58

## ДОДАТКИ

## ДОДАТОК А

### Відомість дипломного проекту

## ДОДАТОК Б

### Схема бази даних



## ДОДАТОК В

### Алгоритм роботи користувача та адміністратора

## ДОДАТОК Г

### UML діаграма класів