

МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ
ОДЕСЬКИЙ ДЕРЖАВНИЙ ЕКОЛОГІЧНИЙ УНІВЕРСИТЕТ

Факультет Комп'ютерних наук
Кафедра Інформаційних технологій

ДИПЛОМНА РОБОТА

Рівень вищої освіти бакалавр

На тему: Дослідження технологій адміністрування та контролю в
комп'ютерних мережах

Виконав студент 5курсу групи КН
Напрямок підготовки 6.0501101
Комп'ютерні науки
Ількін Віктор Олегович

Керівник к.т.н., доцент
Онищенко Сергій Михайлович

Консультант _____

Рецензент к.геогр.н.
Лужбін Анатолій Михайлович

Одеса 2018

ЗМІСТ

ВСТУП.....	5
1 АНАЛІЗ ПРЕДМЕТНОЇ ОБЛАСТІ	
1.1 Поняття динамічних інструментів для створення послуг в режимі реального часу.....	7
1.2 Огляд наявних програмних продуктів.....	15
2 ТЕХНІЧНЕ ЗАВДАННЯ ДЛЯ РОЗРОБКИ веб-сторінки	
2.1 Визначення функціональних і не функціональних вимог та структури веб-торінки.....	24
2.2 Вибір та обґрунтування програмних засобів.....	29
3 РОЗРОБКА веб-сторінки для ELIT TENNIS CLUB	
3.1 Розробка архітектури.....	38
3.2 Розробка дизайну додатку.....	41
3.3 Розробка функціональних можливостей.....	43
3.4 Апробація роботи.....	46
3.5 Інструкція для користувача.....	49
ВИСНОВКИ.....	54
ПЕРЕЛІК ПОСИЛАНЬ.....	55

ВСТУП

Актуальність теми. Велика кількість спортивних центрів мають власні веб-сайти, але там немає реалізованого функціоналу, який би давав можливість записатися на послугу онлайн. Зазвичай щоб записатися, потрібно йти в приймальню центру. Тому розробка веб-сервісу, який надаватиме можливість користувачу записуватися на послугу в режимі онлайн є актуальною. Такий програмний продукт пришвидшить та спростить процес отримання послуги в центрі.

На сьогодні будь-яка організація здійснює рекламу послуг, окремо від цього працює і система планування графіку надання цих послуг. Розроблений продукт дозволить оптимізувати час клієнта, а також працівників, які працюють у різних спортивних центрах, що знаходяться у різних місцях, а всі необхідні функції – зібрані в одному додатку.

Метою роботи є дослідження та аналіз процесу надання клієнту отримання необхідної інформації та замовлення послуги, використовуючи графічний інтерфейс веб-сайту. Відштовхуючись від мети, роботу можна поділили на такі задачі:

- 1) Дослідити веб-сайти, які надають користувачам отримання інформації, запис на послугу. На основі проведеного аналізу розробити функціональні та нефункціональні вимоги до майбутнього програмного продукту;
- 2) Розробити архітектуру програмної системи та спроектувати структуру бази даних;
- 3) Зробити вибір мов програмування та технологій для програмної реалізації описаного продукту;
- 4) Провести тестування розробленого програмного продукту.

В процесі розробки використовуватимуться наступні мови програмування та технології:

- PHP – інтерпретована мова програмування для виконання скриптів та генерування HTML сторінок на стороні сервера;
- JavaScript – об'єктна мова програмування, що виконується на стороні клієнта в браузері;
- MySQL – система управління реляційними базами даних;
- SQL – мова запитів для взаємодії з сервером баз даних;

Розроблений веб-сайт дозволить використовувати його в майбутньому для здійснення планування графіку тренувань за допомогою мережі Інтернет.

Об'єктом дослідження є процеси які дозволяють працювати з послугами, забезпечать перегляд інформації, зв'язок з адміністратором. Предметом – методи які дозволяють реалізувати вище перераховані процеси.

1 АНАЛІЗ ПРЕДМЕТНОЇ ОБЛАСТІ

1.1 Поняття динамічних інструментів для створення послуг в режимі реального часу.

Для створення динамічних web-сайтів використовують спеціальні мови веб-програмування. Найбільш використовувані: ASP.NET (Microsoft), Java (Sun Microsystems), PHP, Python, Perl і Ruby. Мови web-програмування дозволяють автоматично, за запитом користувача, генерувати статичні html-сторінки і передавати їх користувачеві.

ASP (Active Server Pages) – технологія розроблена компанією Microsoft. Працює на платформі Windows і веб-сервері IIS (Internet Information Server). Для роботи потрібно база даних Microsoft SQL і операційна система на основі Windows, які є комерційними продуктами (MS SQL 2014 стандартний пакет - 3.700 \$ ~ 200.000 руб.,

Windows Server 2012 R2 стандартний набір – 800 \$ ~ 45.000 руб.), Що робить дану технологію недоступним для малого і середнього бізнесу.

Python – застосовується як інтерпретована мова для скриптів різного призначення. Синтаксис програми Python приблизно однаковий з описує задачу псевдокодом, що дозволяє зменшити обсяг програми. Була реалізована Гвідо ван Россум. Його можна використовувати, як в великих проектах, так і в малих. Інтенсивно застосовується такими корпораціями як Google і Yandex.

Java – повністю об'єктно-орієнтована мова програмування, розроблений компанією Sun Microsystems. За допомогою Java можна реалізувати проект будь-якої складності. Java здатна працювати на будь-якому комп'ютері, завдяки використанню віртуальної Java-платформи.

Perl – високорівнева інтерпретується динамічний мова програмування загального призначення, створений Ларрі Уоллом. Основна

особливість цієї мови – це широкі можливості при роботі з текстом, з регулярними виразами.

Ruby – динамічний об'єктно-орієнтована мова програмування, розроблений Юкіхіро Матсумото. Характеризується динамічною типізацією і автоматичним управлінням пам'яттю. Мова Ruby використовується в веб-розробці в складі відкритого веб-фреймворку Rails, частіше званого Ruby on Rails.

PHP – мова сценаріїв, що виконується на стороні веб-сервера, розроблений Расмусом Лердорфом як інструмент створення динамічних та інтерактивних веб-сайтів.

PHP досить гнучкий і потужний, тому даний мова використовується в проектах будь-якого масштабу: від простого блогу до найбільших веб-додатків в Інтернеті. В даний час підтримується більшістю хостинг-провайдерів і є одним з лідерів серед мов програмування, що застосовуються для створення динамічних веб-сайтів. Код PHP інтерпретується Веб-сервером і генерує HTML або інший документ, що спостерігається відвідувачем сторінки.

Ця група мов відрізняється від алгоритмічних мов, насамперед решаемими завданнями. База даних – це файл (або група файлів), що представляє собою упорядкований набір записів, що мають однакову структуру і організовані за єдиним шаблоном (як правило, в табличному вигляді). База даних може складатися з декількох таблиць. Зручно зберігати в базах даних різні відомості з довідників, картотек, журналів бухгалтерського обліку і т. Д.

При роботі з базами даних найчастіше потрібно виконувати наступні операції:

- 1) створення / модифікація властивостей / видалення таблиць в базі даних;
- 2) пошук, відбір, сортування інформації за запитам користувачів;
- 3) додавання нових записів;
- 4) модифікація існуючих записів;
- 5) видалення існуючих записів.

Перші бази даних з'явилися дуже давно, як тільки з'явилася потреба в обробці великих масивів інформації і вибірки груп записів по визначеним ознаками. Для цього був створений структурована мова запитів SQL (Structured Query Language), Він заснований на потужній математичної теорії і дозволяє виконувати ефективну обробку баз даних, маніпулюючи не окремими записами, а групами записів.

Для управління великими базами даних і їх ефективної обробки розроблені СУБД (Системи Управління Базами Даних).

Практично в кожній СУБД крім підтримки мови SQL є також свій унікальний мову, орієнтований на особливості цієї СУБД і не переносимо на інші системи.

Сьогодні в світі налічується п'ять провідних виробників СУБД: Microsoft (SQL Server), IBM (DB2), Oracle, Software AG (Adabas), Informix і Sybase. Їх продукти націлені на підтримку одночасної роботи тисяч користувачів в мережі, а бази даних можуть зберігатися в розподіленому вигляді на декількох серверах. У Oracle є встроєний мову PL / SQL, в Informix – INFORMIX 4GL, в Adabas – Natural і т. Д.

З появою персональних комп'ютерів були створені так звані настільні СУБД. Родоначальником сучасних мов програмування баз даних для ПК прийнято вважати СУБД dBase II, мова якої був інтерпретується. Потім для нього були створені компілятори, з'явилися СУБД FoxPro і Clipper, що підтримують діалект цієї мови. Сьогодні схожі, але несумісні версії мов сімейства dBase реалізовані в продуктах Visual FoxPro фірми Microsoft і Visual dBase фірми Inprise.

Графіка і файлова система:

Технологія файлової системи являє собою набір папок-каталогів, виконавчих файлів (html, css, php, javascript документи) і графіки.

Будь-який сучасний браузер підтримує безліч форматів графіки. Найпопулярніші: jpg, png, gif, flash, а також з'явилася можливість виведення векторної графіки svg.

Програми та інструменти

У сучасному світі з'явилася велика кількість програм та інструментів, які допомагають створювати якісні веб-ресурси. Я перерахую тільки найпопулярніші і необхідні.

Програми, що дозволяють спростити технологію створення сайту:

Dreamweaver – зручна програма, що дозволяє швидко створювати сайти.

Photoshop – дозволяє редагувати картинки і зберігати їх у форматі для web.

Notepad ++ - технологічний редактор з підсвічуванням коду.

Filezilla – FTP-клієнт, який дозволяє швидко і без проблем завантажувати / вивантажувати документи на хостингу.

WordPress – написана на PHP с використання СУБД MySQL, застосовується для створення блогів, новинних сайтів, інтернет-магазинів.

Joomla – написана на PHP і JavaScript с використання СУБД MySQL, можливості дуже широкі від невеликого сайту до великого корпоративного проекту.

CMS (технологія – система управління сайтом)

CMS це система управління сайтом, завдяки якій ви можете запустити веб-проект за лічені хвилини. У наш час технологія CMS містить в собі більшість необхідних зв'язків, модулів і плагінів «прямо з коробки». Що б ви не хотіли зробити - інтернет магазин, корпоративний сайт або блог – як правило, в будь-якої CMS це вже реалізовано. Їх є безліч, як платних, так і безкоштовних, наприклад, bitritx, modx, wordpress, joomla, opencart та інші.

Основні особливості CMS системи:

- 1)Надання інструментів для створення, редагування, видалення вмісту сайту;
- 2)Управління вмістом: зберігання, контроль версій, дотримання прав доступу і т. п. ;
- 3)Налаштування призначеного для користувача інтерфейсу;
- 4)Подання інформації у вигляді, зручному для навігації і пошуку.

В системі управління контентом можуть перебувати найрізноманітніші дані: документи, відео, фотографії, контакти, наукові дані і т.д.

Така система часто використовується для зберігання, управління, перегляду і публікації документації. До появи CMS систем, для того, щоб створити сайт, необхідно було знати масу технологій і мов веб-програмування, наприклад, таких як: HTML, CSS, PHP; володіти деякими навичками дизайнера і мати загальне уявлення про процес розробки веб-проекту.

Тепер будь-якій людині стало доступне створення інтернет-сайтів, по дизайну і функціональності нічим не поступають продуктам професійних веб-студій.

Однак стартові набори таких CMS систем не завжди можуть відповідати всім поставленим вимогам. Якщо спочатку існують специфічні запити до функціоналу майбутнього сайту, наявність нестандартних рішень, краще відразу звернутися до хорошого фахівця-розробнику, або до людини, яка розбирається в специфіці CMS системи, який допоможе довести стандартний функціонал до необхідних вимог.

CMS Joomla

Joomla (Джумла) – система, розроблена на PHP і використовує базу даних MySQL. Дана система підходить під будь-які завдання. Має зрозумілу для початківця користувача панель адміністрування. Створено безліч модулів, плагінів і розширень для Джумли, що дозволяють робити на її основі різні сайти: від простого сайту-візитки до інтернет-магазину.

CMS Wordpress

Wordpress (Вордпресс) – система розроблена на PHP, в якості сервера бази даних використовує MySQL. Переваги системи - це її простота адміністрування, підтримка RSS, безліч розширень і доповнень. Система підійде для введення блогу на сайті.

CMS 1C Bitrix

Bitrix (Бітрікс) – комерційна система розроблена компанією 1С, орієнтована на корпоративні сайти, інформаційні та довідкові портали, соціальні мережі, інтернет магазини, сайти ЗМІ. Найпоширеніша комерційна система в російській сегменті Інтернету. Підтримує наступні СУБД: MySQL, Oracle, MS SQL.

CMS Drupal

Drupal (Друпал) – дана CMS написана на PHP і використовує як сховище даних реляційну базу даних (підтримуються MySQL, PostgreSQL та інші). Ядро Друпал має дуже добре опрацьовану структуру, де передбачена можливість переробити всю систему на інші цілі, написавши додаткові модулі, що дозволяє з її допомогою створювати великі портали і специфічні веб-додатки. Це робить Друпал найбільш популярним серед професійних розробників.

Аналіз переваг і недоліків системи управління сайтом CMS.

Система управління сайтом (CMS) – система управління вмістом (контентом) – комп'ютерна програма або інформаційна система, яка використовується для організації та забезпечення процесу щодо спільного створення, управління і редагування вмісту сайту. Сьогодні існує безліч CMS: від популярних, за якими стоять величезні компанії-розробники, до самописних, нікому не відомих.

CMS бувають абсолютно різні, деякі системи орієнтовані тільки на вирішення конкретних завдань (ведення блогів, інтернет магазини, форуми), інші є універсальними і надають розробником зручне середовище проектування і програмування для розробки чого завгодно. Частина CMS складатися з безлічі функціональних блоків і модулів, інші монолітні, неподільні, та ще й зашифровані. Одні системи поставляються безкоштовно і з можливістю внесення своїх доробок, а деякі надаються за гроші і не допускають можливість редагування ядра.

В основному сьогоднішні масові CMS використовують генерацію HTML / XML-документів під керуванням двох скриптових мов: PHP – на стороні сервера, JavaScript – на стороні клієнта (браузера). При цьому можливості стандартного web-сервера, що складається з трьох основних

частин: мережевого сервера, транслятора PHP, системи управління базами даних (зазвичай MySQL), можуть використовуватися в різного ступеня.

Існують різноманітні системи управління сайтом, серед яких зустрічаються платні і безкоштовні, популярні і непопулярні, побудовані за різними технологіями. Кожен сайт має панель управління, яка є тільки частиною всієї програми, але достатня для управління ним.

Комерційні коробкові CMS – продукти, створені комерційними організаціями з метою отримання прибутку від продажу ліцензії і / або технічної підтримки. Серед відомих платних систем управління можна виділити: 1С -Бітрікс, UMI.CMS, NetCat, HostCMS і інші.

Безкоштовні CMS з'явилися, як альтернативне бізнес-рішення, з тією лише різницею, що в першому випадку компанія заробляє на продажу ліцензій, а в другому – на підтримку та впровадженні будь-якого Web - ресурсу. За своєю функціональністю безкоштовні CMS не поступаються платним аналогам і також популярні серед розробників. До таких систем відносяться: Joomla, WordPress, Drupal, MODx, OpenCart та інші.

Приватні розробки невеликих студій, вузькоспеціалізовані CMS і інші рішення можна віднести до непопулярних CMS.

Підключення CMS.

плюси:

- можливість самостійного наповнення;
- можливість розподіленого адміністрування;
- структурованість інформації;
- розширення можливостей за допомогою готових модулів;
- автономність (клієнт «не буде залежати від Web-студії»).

мінуси:

- подорожчання розробки сайту, за рахунок більш складного; програмування;
- наступні доопрацювання сайту також пов'язані з правкою CMS;
- плата за ліцензію (у разі платної CMS);
- підвищена оплата хостинг -майданчик;
- вимогливість до ресурсів (сайт довше завантажується);
- наявність вразливостей;
- низький рівень безпеки (частіше для безкоштовних CMS).

Як видно, основним аспектом установки CMS є отримання можливості самостійного редагування Web-ресурсу.

Системи управління контентом з'явилися на ринку інформаційних технологій системи управління контентом (Content Management System – CMS) дозволили знайти компроміс між низькою гнучкістю статичних і високою вартістю динамічних сайтів і в результаті підвищити керованість і знизити витрати на адміністрування ресурсу.

Система управління контентом – це автоматизований редакторський комплекс, службовець для управління змістом і структурою Інтернет-ресурсу в онлайн-режимі. На практиці це означає, що будь-який, навіть незнайомий з інформаційними технологіями користувач може абсолютно самостійно управляти змістом сайту, не вдаючись при цьому до послуг професіоналів. За рахунок обмежень, що накладаються на логічну структуру контенту, на дизайн і функціональні можливості створюваних динамічних сайтів, при використанні таких систем радикально знижується трудомісткість розробки і підтримки.

Ринок систем управління контентом порівняно молодий, але вже встиг зайняти свою нішу в сфері сайтопостроєння. Розробники контент-систем пропонують своїм клієнтам продукцію різного рівня складності в залежності від обсягів фірми, роду діяльності, кількості інформації, обслуговуючого персоналу, максимальної пропускної здатності і т. Д. Відрізняється і структура систем.

Варто зауважити, що якою б складною не була система, розробники орієнтуються на деякі загальні принципи, в основі яких лежить уявлення про життєвий цикл контенту.

Підтримка розробки контенту по суті справи є одним з ключових компонентів таких програмних продуктів. Саме з розробки починається життєвий цикл будь-якого матеріалу, опублікованого на сайті: ведуться роботи по його створенню, редагуванню та утвердження контенту, а роль системи полягає в автоматизації цих процесів, підтримці спільної роботи авторів, редакторів, програмістів і менеджерів.

Досягнення поставленої мети здійснюється завдяки поділу контенту і дизайну. Всі компоненти сайту, включаючи шаблони і наповнення, зберігаються в певних місцях сховища даних. Система ж автоматично звертається в потрібні місця сховища, дозволяючи безлічі користувачів, навіть не є технічними фахівцями, працювати над підготовкою контенту до публікації, включаючи перевірку його достовірності.

Управління діловими процесами включає етапи розробки структури ресурсу, попереднього перегляду і публікації підготовленого контенту.

Визначається зовнішній вигляд, готуються шаблони, розподіляються ролі користувачів і класифікується необхідна бізнес-інформація (товари, ціни, обсяги та т. Д.). Важливими компонентами тут є служби, які підтримують своєчасність надходження необхідного контенту.

При цьому ведеться аудит подій і зберігаються версії документів, причому при необхідності користувачі можуть звертатися до попередніх версій.

Коли сайт повністю підготовлений до публікації, в роботу вступають засоби динамічного формування Web-сторінок в залежності від прямої і непрямой інформації про конкретний споживача. Пряма інформація збирається з особистих карток, непряма – переважно на основі поведінки користувача.

Є три основні форми динамічного поширення контенту – персоналізація, локалізація і глобалізація. Безумовно, контент може бути статично або динамічно згенерований і надано відвідувачам безпосередньо, без застосування будь-якої було поведінкової логіки.

Персоналізація в тій чи іншій мірі використовується тепер на більшості Web-сайтів. Тут вступають в хід особисті вітання, рекламні оголошення і особливим чином скомпоновані фрагменти контенту, все це покликане сприяти залученню відвідувачів на сервер.

Існує два методи персоналізації: на основі правил і за допомогою фільтрів.

Персоналізація на базі правил – це надання контенту певним користувачам або їх групам із застосуванням умовної бізнес-логіки. Для розробки правил зазвичай служить вводиться користувачем в реєстраційну картку інформація про себе і своїх перевагах.

При персоналізації за допомогою фільтрів застосовуються складні алгоритми категоризації і надання контенту на основі аналізу поведінки користувача - до якої інформації він звертається, які сайти відвідує і т. Д.

Класифікація систем управління контентом

Чіткої класифікації систем управління контентом в даний час не існує, оскільки такий вид програмних продуктів ще новий, не надто поширений і має невисокий рейтинг на ринку. За типом розробки їх можна поділити наступним чином.

– Модульні системи. Являють собою порівняно недорогі рішення, орієнтовані на представницькі сайти для середнього бізнесу і корпоративні Web-ресурси, що розробляються як "первинні" проекти. Вони призначені для

створення і підтримки корпоративних або тематичних Web-сайтів. Дуже багато виробників контент-систем будують свої продукти за модульним принципом. У цьому підході є переваги. Виробник може розраховувати на велику аудиторію в залежності від потреб компаній – потенційних покупців.

У свою чергу, компанії-покупці можуть поступово набувати модулі і підвищувати функціональність і гнучкість свого електронного ресурсу. Варто зауважити, що купити відразу повний пакет програм за кілька тисяч доларів може дозволити собі далеко не кожна фірма.

– Спеціалізовані системи. Орієнтовані на певний вид діяльності і відображення її на електронному ресурсі. Служать для управління Інтернет-спільнотами, реалізації рішень для електронної комерції, ЗМІ і т. Д. Базовою функціональністю цих систем досить для обслуговування найскладніших процесів певного типу, чого немає в універсальних продуктах. З іншого боку, їх впровадження набагато дешевше, ніж замовлена розробка.

– Замовні системи. Розробка системи ведеться з урахуванням цілей, завдань і побажань замовника.

Існує також класифікація систем управління контентом за призначенням.

– Системи управління Web-контентом. Призначені для управління Web-контентом на сайті (додавання, видалення, редагування контенту), більш складні розробки забезпечують функції адміністрування існуючих Web-ресурсів (зміна структури Web-сайту, розмежування прав доступу), а також можуть бути інструментом для розробки нових Web-ресурсів. За допомогою цих систем власник може самостійно управляти вмістом свого ресурсу, не вдаючись до послуг розробників. Дані продукти створюються, як правило, дрібними Web-студіями і окремими Web-розробниками. З цієї причини їх не можна розглядати як серйозних конкурентів.

– Системи електронного документообігу підприємства, що використовують Web-інтерфейс. До даної категорії систем управління контентом відносяться системи документообігу масштабу підприємства, в яких є функції роботи з Web-документами через Web-інтерфейс. Системи документообігу часто використовуються для оптимізації бізнес-процесів середніх і великих підприємств. Вони інтегровані з корпоративною інформаційною системою підприємства і дозволяють створювати портальні інтранет-рішення, системи взаємодії з постачальниками і збутовою мережею.

Системи, побудовані за модульною схемою, дозволяють нарощувати свою функціональність шляхом введення додаткових модулів.

1.2 Огляд наявних програмних продуктів

На етапі проектування розглянемо існуючі програмні продукти, які виконують надання послуг спортивного центру, а саме:

- Prairie stone sport & wellness center;
- Peterborough Sport & Wellness Centre;
- Sports & Wellness.

Prairie stone sport & wellness center – програмний продукт який ознайомлює нас з спортивним центром «Prairie stone». Даний веб сервіс надає доступну інформацію про центр, послуги, які він надає. Присутній зручний пошук по сайту. Також ресурс дозволяє відправляти повідомлення до адміністратора.

Головна сторінка сайту містить такі елементи:

Навігаційне меню сайту (рис. 1.1):

- Welcome;
- programs;
- amenities;
- group fitness;
- membership;
- kids korner;
- join our elist;
- contact us.



Рисунок 1.1 – Навігаційне меню сайту

Сторінка для зв'язку з адміністратором на рис.1.2.

Contact Us

Thank you for visiting praiestonesports.com. If you have any questions or comments please contact by filling out the form below. For immediate attention call 847.285.5405 for more information.

First Name *

Last Name *

E-mail Address *

Phone

Street Address

City

State

Zip Code

Comments/Questions *

txtetxtetxtetxtetxtetx

tetxtetxtetxtetxtetx

txtetxtetxtetxtetxtetx

tetxtetxtetxtetxtetx

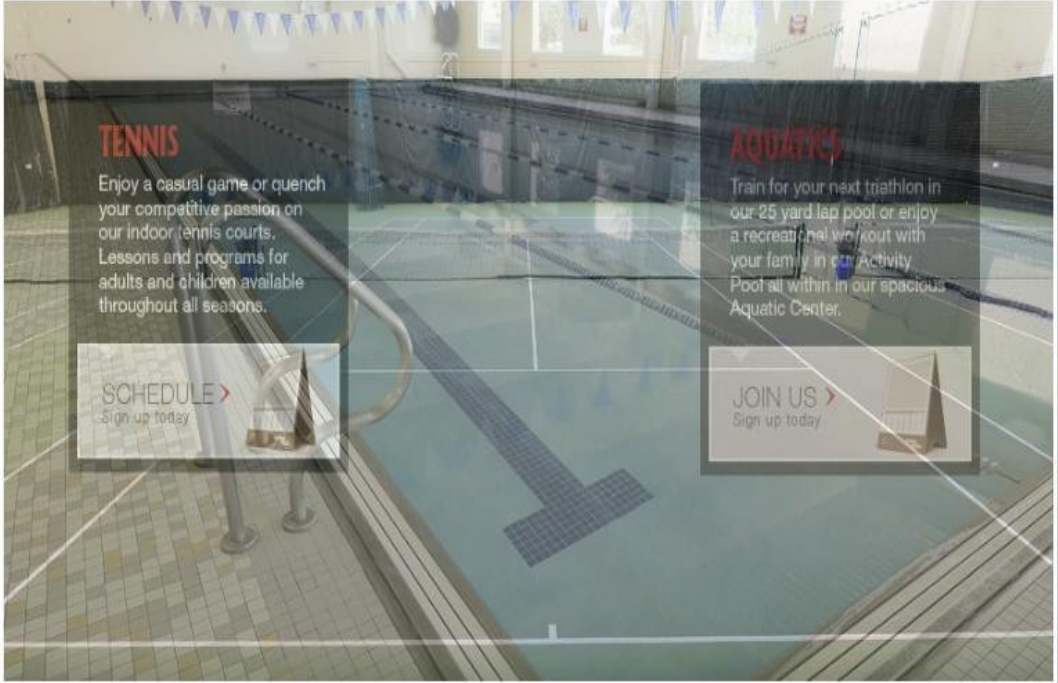
txtetxtetxtetxtetxtetx

tetxtetxtetxtetxtetx

txtetx

Рисунок 1.2 – сторінка Contact Us

Слайдер з фотографіями даного центру, календар на рис.1.3.



TENNIS

Enjoy a casual game or quench your competitive passion on our indoor tennis courts. Lessons and programs for adults and children available throughout all seasons.

SCHEDULE >
Sign up today

AQUATICS

Train for your next triathlon in our 25 yard lap pool or enjoy a recreational workout with your family in our Activity Pool all within in our spacious Aquatic Center.

JOIN US >
Sign up today

Meet Your Goals. Your Way.

As a member of Prairie Stone™ Sports & Wellness Center you will find more than just a place to exercise. You will find a family of dedicated employees committed to helping you achieve your wellness goals. It is because of that dedication that you will find state-of-the-art fitness equipment; innovative group exercise classes, superior cleanliness, and the desire to continuously develop new and exciting ways help you improve your health.

FITNESS•CALENDAR						
May 2016						
S	M	T	W	T	F	S
1	2	3	4	5	6	7
8	9	10	11	12	13	14
15	16	17	18	19	20	21
22	23	24	25	26	27	28
29	30	31				

Рисунок 1.3 – Головна сторінка сайту з інформацією

Поле для пошуку розміщено на рис. 1.4.



Рисунок 1.4 – Поле для пошуку

На рисунку 1.5 зображено загальний інтерфейс програмного продукту.

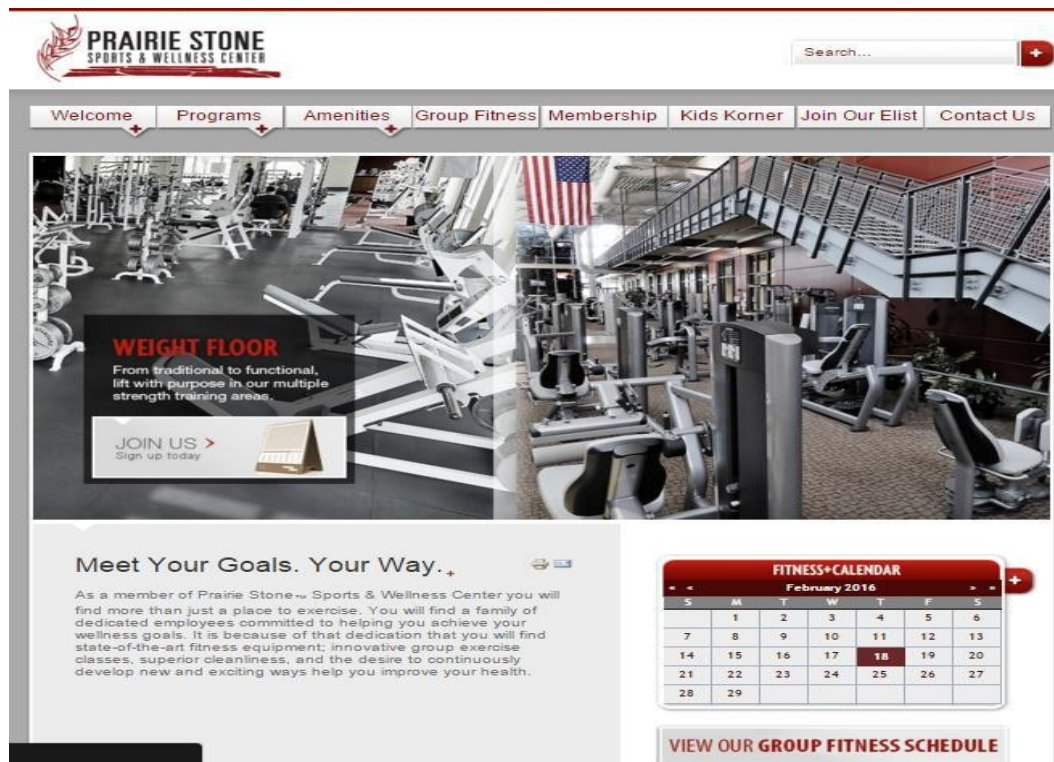


Рисунок 1.5 – загальний інтерфейс програмного продукту «Prairie stone»

Peterborough Sport & Wellness Center– програмний продукт який ознайомлює з спортивним центром «Peterborough Sport». Даний web-сайт надає доступ до інформації, яка дає повідомляє користувачу про центр та

його послуги. Web-сайт є зручним у користуванні, інформативним і інтуїтивно зрозумілим. Головна сторінка сайту містить наступні елементи:

Навігаційне меню сайту (рисунок 1.6).

- Home;
- membership;
- aquatics;
- fitness;
- personal training;
- child;
- about us.

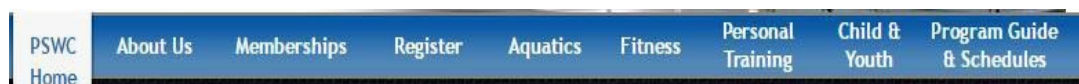


Рисунок 1.6 – меню сайту

На головній сторінці розміщено інформацію про робочий час центру та інша необхідна користувачу інформація.



Рисунок 1.7 – Інформація для користувача

необхідну інформацію про себе. Користуватися ним зручно і легко, все розміщено по відповідних меню тому зорієнтуватися легко. Головна сторінка містить наступні елементи:

- Навігаційне меню;
- Fitness & wellness;
- Amenities;
- Family;
- Membership;
- Club community;
 - Розклад занять;
- Фото з тренувань.

Інформація подається коротко, зрозуміло і по темі. Даний сервіс має свою спільноту, що дозволяє краще співпрацювати з клієнтами.

На рисунках 1.10 зображено головну сторінку web-сайту спортивного центра «Sports & Wellness» всю необхідну інформацію про себе має на інтернет-сторінці.

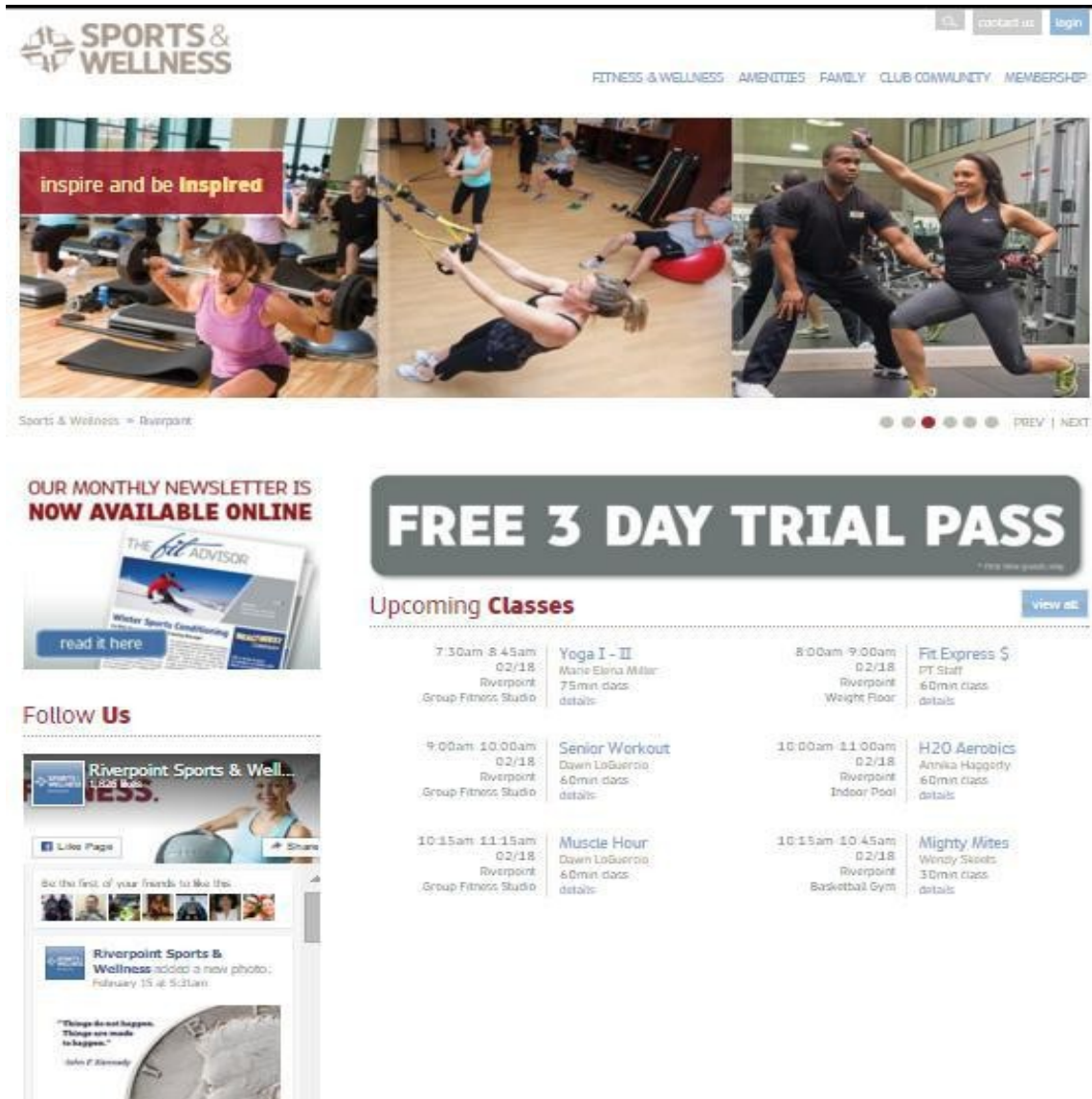


Рисунок 1.10 – інтерфейс веб-сервісу «Sports & Wellness»

На рисунку 1.11 зображено сторінку з навігацією по web-сайту.



Рисунок 1.11 – навігація по веб-сервісу

На рисунку 1.12 зображено сторінку авторизації на сайті.

На рисунку 1.13 зображено сторінку зв'язку з адміністратором.



Sports & Wellness

contact us

Fill out the form below and a Club Representative will contact you shortly.

First Name *

alan

Last Name *

pimp

Email *

alan757575@gmail.com

Phone *

097.....8

Are you a current Member? *

☒ Yes, I am a Member

☐ No, I am not a Member

How can we help you? *

Рисунок 1.13 – Сторінка “contact us”

Порівняльна характеристика програмних продуктів відображена в таблиці 1.1

Таблиця 1.1– Порівняльна характеристика програмних продуктів

Назва програмного продукту	Prairie stone	Peterborough Sport & Wellness Centre	Sports & Wellness
функціональність	Перегляд інформації, зв'язок з адміністратором	Перегляд інформації, зв'язок з адміністратором, пошук по сайту	Перегляд інформації, зв'язок з адміністратором, пошук по сайту, реєстрація авторизація на сайті
Інтерфейс користувача	Зручний у використанні	Зручний використанні інтуїтивно зрозумілий інтерфейс	Зручний у використанні
Допомога користувачу	Присутня	Присутня	Присутня

В результаті аналізу існуючих аналогів, виділено наступні недоліки:

- 1) не адаптовані під об'єкт управління;
- 2) велика кількість інформації;
- 3) скупчення функціоналу;
- 4) відсутність можливості записатися на послугу через веб-сайт.

Кожен з розглянутих продуктів має також свої переваги і недоліки. Проаналізувавши вище переглянуті приклади, можна використати деякі їхні рішення при розробці програмного продукту. Проте потрібно їх адаптувати під обрану предметну область і функціонал розроблюваного продукту.

2 ТЕХНІЧНЕ ЗАВДАННЯ ДЛЯ РОЗРОБКИ ВЕБ-СТОРІНКИ

2.1 Визначення функціональних і не функціональних вимог та структури веб-сторінки

Для належної роботи веб-сервісу мережі спортивно-оздоровчих центрів необхідне виконання таких бізнес процесів(рисунок 2.1):

- додавання послуги;
- редагування послуги;
- видалення послуги;
- перегляду інформації;
- авторизація адміністратора;
- відправка листа адміністратору;
- запис на послугу.

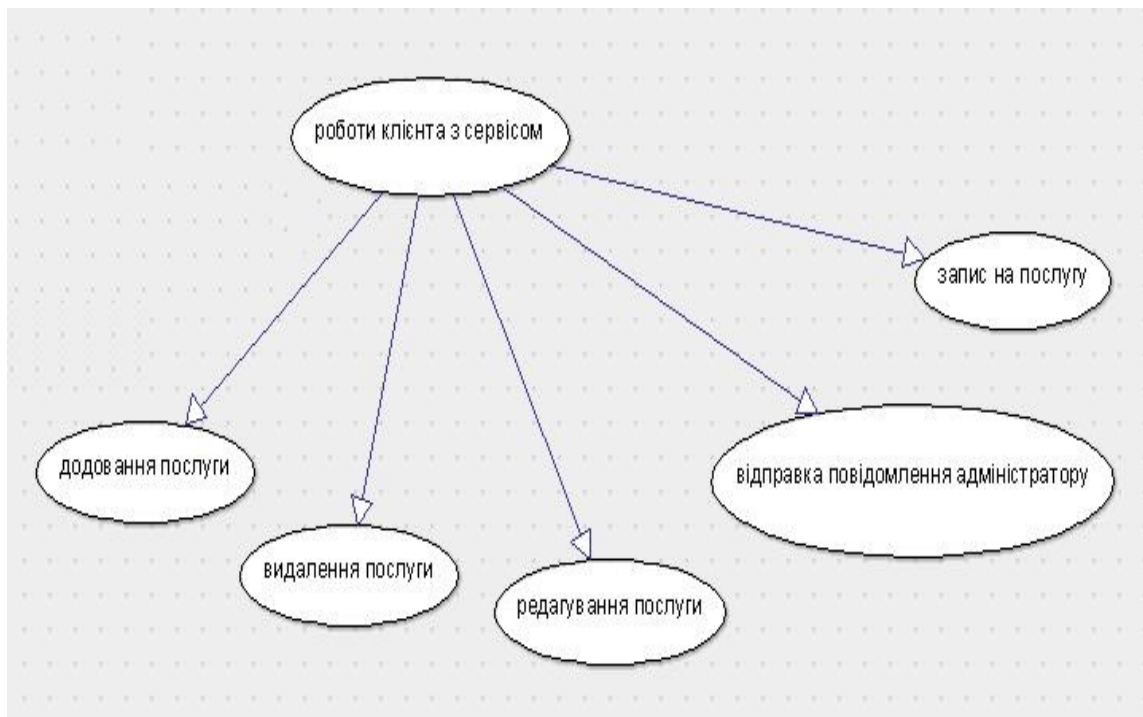


Рисунок 2.1 – Діаграма бізнес-процесів розроблюваного програмного продукту

Розглянемо детальніше кожен з представлених на рисунку 2.1 бізнес-процесів.

На рисунку 2.2 бачимо діаграму функції процесу додавання послуги.

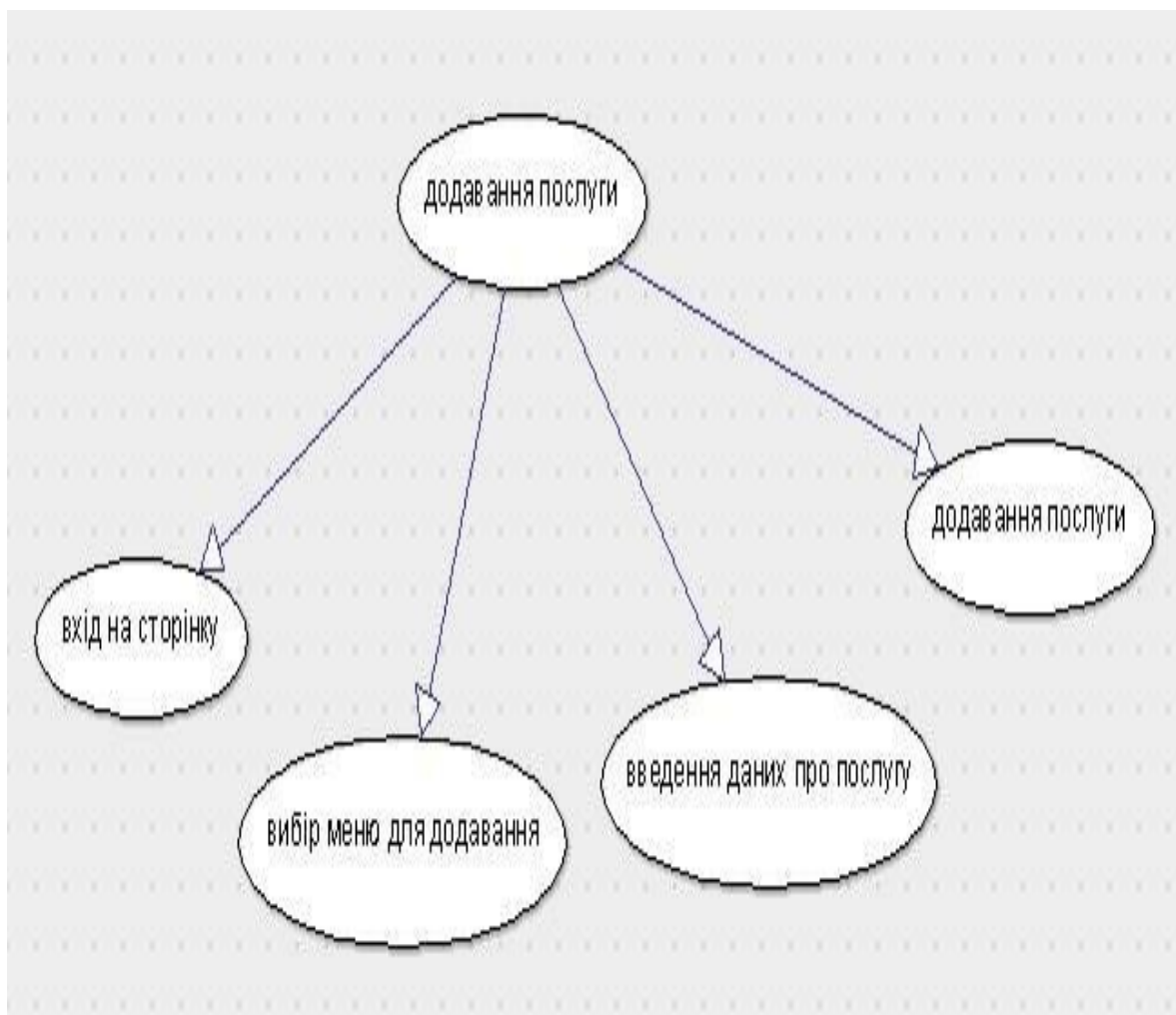


Рисунок 2.2 – Діаграма функції процесу додавання послуги

Додавання послуги в веб-сРисуноеервісі складається з наступних кроків:

- 1) Вхід на сторінку адміністратора;
- 2) Вибір меню для подавання;
- 3) Введення даних про послугу;
- 4) Додавання послуги;
- 3) Підтвердження додавання.

Характеристику бізнес-процесу додавання послуги , а також характеристики послуг,запити на подавання послуг, додавання послуги на веб-ресурсі , розпорядження директора про внесення нової послуги в перелік, значення характеристик , різноманітні повідомлення від клієнтів спортивного закладу наведено в таблиці 2.1.

Таблиця 2.1– Характеристика бізнес-процесу додавання послуги

Назва характеристики	Значення характеристики
Ім'я бізнес-процесу	Додавання послуги на веб-ресурсі
Основні учасники	Адміністратор
Вхідна подія	Запит на додавання послуги
Вхідні документи	Розпорядження директора про внесення нової послуги в перелік.
Вихідна подія	Дані про додавання
Вихідні документи	Повідомлення
Клієнт бізнес-процесу	Адміністратор

У системі є два актори: адміністратор і клієнт. Згідно їхніх ролей система буде надавати їм відповідний функціонал. Функціональні можливості продукту будуть відображені на діаграмі варіантів використання зображений на рис. 2.3.

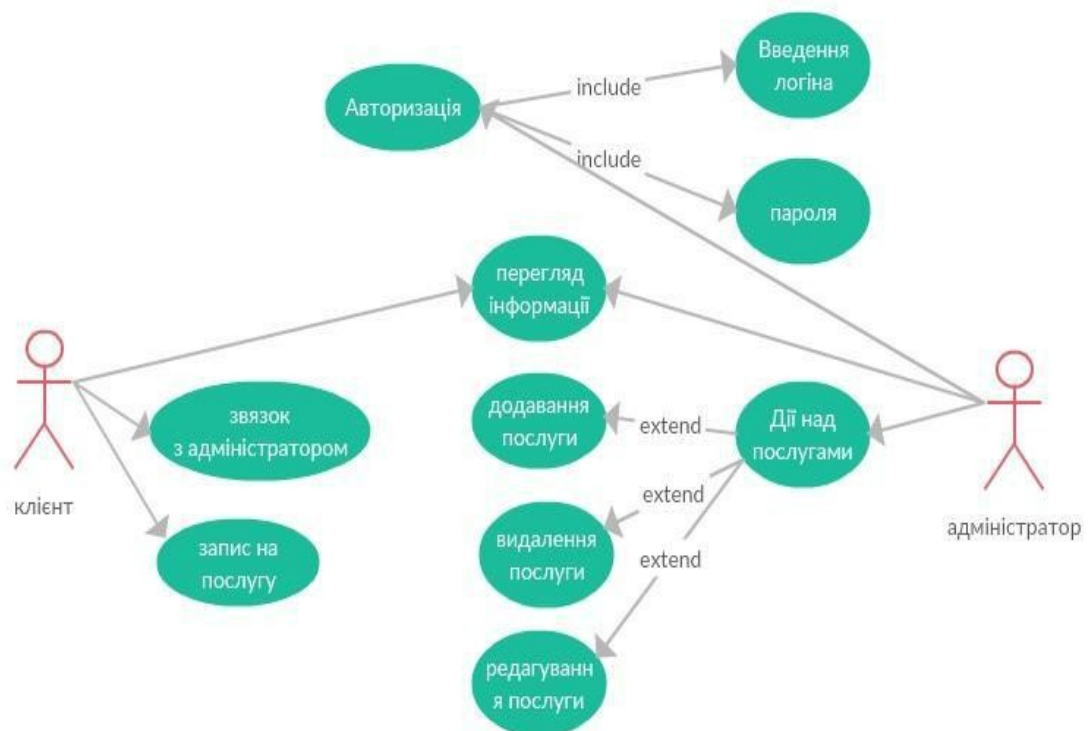


Рисунок 2.3 – Діаграма варіантів використання

Опис варіантів використання подано у таблиці 2.2. Розглянемо такі варіанти використання.

Таблиця 2.2 – Варіант використання «Перегляд інформації»

Контекст використання	дозволяє переглядати інформацію яку надає ресурс		
Дійові особи	Адміністратор, користувач		
Передумова			
Тригер	Перший запуск додатку		
Сценарій	Вхід користувача на сторінку		
Постумова			

Специфікація функціональних вимог наведена у таблиці 2.3

Таблиця 2.3 – Специфікація функціональних вимог

Ідентифікатор вимог	Назва вимоги	Атрибути вимог		
		Пріоритет	Складність	Контакт
1	Авторизація користувача	Обов'язкове	Середня	-
3	Додавання послуги	Обов'язкове	Середня	-
4	Редагування послуги	Обов'язкове	Середня	-
5	Видалення послуги	Обов'язкове	Середня	-
6	Запис на послугу	Обов'язкове	Висока	-
7	Зворотній зв'язок	Обов'язкове	Висока	-

Специфікація нефункціональних вимог наведена у таблиці 2.5.

Таблиця 2.4 – Специфікація нефункціональних вимог

	Назва вимоги	Характеристики
1	Застосовність	
1.1	Час для навчання користувачів	Не більше 45 хв
1.2	Вимірюваний час відгуку для типових завдань	Не більше 30 с
2	Надійність	
2.1	Середній час безвідмовної роботи	100 год
2.2	Середнє напрацювання до ремонту	1.5 року
2.3	Максимальна норма помилок або дефектів	50
3	Робочі характеристики	
3.1	Місткість (максимальне значення)	2000 користувачів
4	Проектні обмеження	
4.1	Мова програмування	PHP, JavaScript
5	Вимоги до документації, призначеної для користувача, і до системи допомоги	Наявність інструкції користувача
6	Інтерфейси	
6.1	Інтерфейс користувача	Веб-додаток
6.2	Програмні інтерфейси	Бібліотека jQuery RDBMS MySql
6.3	Комунікаційні інтерфейси	PhpMyAdmin Доступ до мережі інтернет

2.2 Вибір та обґрунтування програмних засобів

Мова всесвітньої павутини постійно розвивається. Так, важливий мова передачі даних в інтернеті html з 1990 року вийшов вже на п'ятий «рівень». У підсумку світ інтернету стає все привабливішим і функціональнішим, а розробники веб-ресурсів змагаються в інноваціях та зручності для користувача. Html 5 - це вже не просто реальність. Сайти на html 5 з'являються все частіше, незважаючи на проблеми зі стандартизацією.

«HyperText Markup Language 5», таким чином, стає новою ерою в створенні сайтів. З цією мовою прийшли нові можливості. Головне – це збільшення інтерактивності сайтів, розробка інтернет-додатків, спрощена взаємодія інтернет-сторінок з мультимедійним контентом, взаємодія з формами, робота з програмним інтерфейсом. Таким чином, сайти на html 5 стають більш інтерактивними і структурованими. Звідси, поліпшення поведінкових факторів і теоретично більш високе ранжування ресурсів пошукачами.

З'являються такі елементи, як «header», «section», «article», «nav», «aside» і «footer». Це робить розмітку більш впорядкованою. Крім цього, завдяки html 5 зручніше використовувати медіа – аудіо, фото і відео. Простіше завантажувати фотографії, медіа тепер можна переглядати в браузері без будь-яких додаткових програм, наприклад, Adobe Flash плеєра. Html 5 також дозволяє простіше вводити дані. У реальному часі ви дізнаєтеся про коректність введення даних або ж про помилки.

Що стосується дизайну сайтів на html 5, то він, безсумнівно, виграє у конкурентів на html 4. Можна досить награтися варіантами поєднання фото з текстом і так далі. Краса та стиль, як то кажуть – це html 5.

Якщо говорити про графіку, то основна перевага html 5 – це нова функція «Canvas» безпосередньо на сторінках веб-ресурсу. З її допомогою без зайвих рухів тіла можна впровадити графіку, а також простенькі ігри. Інтерактивність зміни контенту, звичайно, приваблює відвідувачів.

Аналіз можливостей каскадних таблиць стилів CSS

Точне розташування об'єктів на сторінці відносно один одного є, мабуть, днією з найскладніших завдань для веб-майстра. Написати HTML-код сторінки так, щоб всі графічні зображення та текстові блоки були на своїх місцях не тільки на комп'ютері веб-майстри, а й у всіх відвідувачів сайту, – справжнє мистецтво. Максимально великий кегль, виставлений користувачем у своєму браузері, може чудове творіння перетворити на безладне нагромодження тексту і картинок. Каскадні таблиці стилів (Cascading Style Sheets), декларовані як засіб повного контролю над HTML-розміткою,

подаються до світлі розглянутої проблеми, як хороший інструмент для організації точного розташування об'єктів на сторінці. CSS дозволяють перевизначити всі властивості будь-якого тега, призначаються за умовчанням. Стає можливим виконувати вирівнювання текстового блоку щодо сторінки і інших текстових блоків.

Використання CSS відкриває нові специфічні можливості, аналогів яким немає з стандартному HTML. Наприклад, можливо встановити межбуквенне і міжрядкові відстані для тексту, точно вказати положення малюнка на задньому плані без його мозаїчного викладання по всьому документа. Взагалі, Каскадні таблиці стилів дозволяють задуматися про дизайн веб-сторінки, про естетичну сторону сприйняття інформації користувачем, а не тільки про утилітарною. Із застосуванням CSS зростає швидкість створення нової сторінки. Стили, визначені один раз, можуть бути використані необмежену число раз в будь-якому місці документа. Важливою деталлю є можливість задати стилі в окремому файлі у вигляді присвоєння різних властивостей тегам і створенні користувальницьких стилів. Перевизначення стилів в такому файлі викличе автоматичну зміну стилю відображення всіх об'єктів, для яких застосовувався змінений стиль.

Зупинимося докладніше на способах застосування CSS. Існує чотири способи застосування стилів: Перевизначення стилю в елементі розмітки Розміщення опису стилю в заголовку документа з використанням тега `style` Розміщення посилання на зовнішній опис через тег `Link` Імпорт стилю в документ Перевизначення стилю в елементі розмітки – це застосування деякого доступного в CSS стилю до даного елементу розмітки. При такому способі перевизначення зміни торкнуться тільки того елемента, за який відповідає даний тег і не будуть відображатися на інших елементах, також виводяться цим тегом в іншому місці сторінки. Дозволяє застосувати потрібний стиль до конкретної ділянки документа.

Перевизначити стандартний елемент розмітки або створити власний з необхідними властивостями відображення можна за допомогою парного тега `style`, що знаходиться в заголовку документа перед тегом `body`. Також за допомогою тега `style` створюються довільні класи. Їх використання допускає одноразово задати однакові властивості відображення для різних тегів. Зручно, наприклад, задавати однаковий шрифт для тексту і знаходяться в ньому гіперпосилань. Якщо при цьому не заданий колір і стиль підкреслення, то ці параметри залишаться для гіперпосилань колишніми і вони залишаться виділеними серед звичайного тексту. Коли одні й ті ж стилі

використовуються на кількох або всіх сторінках сайту, не обов'язково визначати їх у кожному html-документі. Досить зберегти їх в окремому файлі і посилатися на нього в кожному документі за допомогою тега `link`, що стоїть в елементі `head` (до тега `body`). Атрибут `rel` при цьому повинен мати значення `"stylesheet"`. Можливо також зробити імпорт стилю в документ. Відмінність від посилання на зовнішнє опис в тому, що імпорт стилю може бути зроблений всередину елемента `style` або всередину зовнішнього файлу з описом стилів, що задається в теге `link`. Таким чином можна створити зовнішній файл для зовнішнього файлу. Інструкція імпорту стилю ставиться перед всіма іншими описувача стилів, що робить можливим легко перевизначити імпортований стиль. Але у будь-якій технології є свої недоліки. Є вони й у технології CSS. Точніше, найбільше недоліків бачиться в застосуванні CSS. Це зумовлено неповною сумісністю CSS з різними браузерами навіть останніх версій і виникає через це відсутність пропагованої в CSS точності відображення сторінки в різних браузерах. Імпорт стилю, наприклад, підтримується далеко не всіма браузерами. З цієї причини багато веб-майстри поки що побоюються застосування всіх можливостей CSS використовуючи тільки деякі. Блокування кегля шрифту при відображенні тексту в браузері дозволяє набути впевненості у точному розташуванні всіх елементів сторінки на комп'ютері користувача, але позбавляє користувача можливості користувача вибрати зручний для читання розмір шрифту самостійно. Такий підхід є, по суті, неповагою до відвідувачеві сайту. Часто про CSS кажуть, що з їх допомогою легко міняти дизайн всього сайту відразу.

Для цього досить перевизначити стилі у файлі стилів. Але справжній дизайнер, що зважився на зміну дизайну сайту, ніколи не зупиниться лише на переробці відображення шрифтів. Каскадні таблиці стилів - перспективна технологія. Але багато веб-майстри поки що обмежуються тільки застосуванням CSS для зміни кольору гіперпосилання при підведенні до ній покажчика миші.

Аналіз можливостей клієнтської мови програмування JavaScript

Мова JavaScript являє собою мову сценаріїв для Web. На сьогодні підтримується майже всіма браузерами. JavaScript є підмножиною мови Java, але не є частиною мови Java. Корпорація Sun не підтримує мову JavaScript. Через свою обмеженість JavaScript не може бути використана для програмування багатьох складних функцій. Вона призначена для

програмістів та користувачів, щоб розширити функціональні можливості мови HTML.

На відміну від аплетів Java, які принципово відрізняються від ресурсів HTML і повинні динамічно завантажуватися при звертанні до сторінки web, сценарій JavaScript включається в сторінку HTML за допомогою дескриптора та інтерпретування браузером в режимі реального часу. Крім того, JavaScript можливо застосовувати для створення серверних додатків.

В якості попередників JavaScript можливо розглянути ряд невеликих проблемно-орієнтованих мов, наприклад, HyperTalk, dBase та LiveScript. На таких мовах, завдяки їх синтаксичній простоті, вмонтованій функціональній простоті створення об'єктів може програмувати навіть початківець. JavaScript дозволяє вносити в сторінки web інтерактивність, забезпечує взаємодію з користувачем, підтримує заповнення форм введення та переміщення по web-документу. Деякі потужні типи систем інтерактивної взаємодії вдається реалізувати за рахунок комбінації можливостей JavaScript з іншими властивостями сторінок web, наприклад, роботи з фреймами та вмонтованими додатками. JavaScript стала новим відкритим стандартом мови сценаріїв Internet, яка підтримується багатьма компаніями.

Для того, щоб повідомити браузеру, що в документі HTML вмонтований сценарій JavaScript, використовують дескриптор. В початковому коді HTML ця команда повинна бути вказана між дескрипторами мови.

Сучасні гіпертекстові інформаційні системи умовно можна уявити у вигляді сукупності декількох комбінацій:

- системи збереження гіпертекстових об'єктів;
- системи відображення гіпертекстових об'єктів;
- системи підготовки гіпертекстових об'єктів;
- системи програмування переглядом сукупності гіпертекстових об'єктів.

Першими були розроблені системи збереження та відображення (1989-1991рр.), які продовжують розвиватись і далі. Після 1991р. стали з'являтися і перші системи підготовки документів. Лише після 1995р. були запропоновані і перші мови управління сценаріями перегляду.

Програми перегляду гіпертекстових сторінок традиційно називають скриптами (scripts). Як це було раніше і в локальних системах, в програмуванні перегляду гіпертекстових документів web існує два методи:

При першому методі для розробки гіпертекстової сторінки потрібний тільки звичайний текстовий редактор і сам і сам гіпертекстовий документ повинен легко читатися людиною-оператором (користувачем). Другий підхід дозволяє збільшити ефективність виконання програм та захист кодів від несанкціонованих модифікацій. Байткоди або мобільні коди забезпечують технологію програмування на Java. Технологія мови JavaScript дуже проста та зручна. Всі операції, які можливо виконувати програмою JavaScript, описують дії над добре відомими та зрозумілими об'єктами, які є елементами робочої області програми браузера Mozilla Firefox і контейнерами мови HTML. Тобто існують об'єкти з множиною властивостей та множиною функцій над об'єктами, які ще називають методами. Крім атрибутів об'єкт та метод, існує ще і третій - подія. Події орієнтовані на роботу в web-додатках, наприклад, завантаження сторінки в робочу область браузера, вибір гіпертекстового зв'язку і т.д. Використовуючи параметр подія можна організувати перегляд динамічних об'єктів: управління багато віконним інтерфейсом, рядок, який біжить і т.д. Кожний із класу об'єктів має функції управління об'єктами класу - метод. Найголовнішими методами є ті, які дозволяють переназначати занесення об'єктів. Робиться це звичайно по операції присвоєння. Всі типи операторів, котрі підтримують відомі мови програмування, реалізовані JavaScript (+, -, *, /, %, ", ", +=, -=, ...). Крім операцій над числами та стандартних класів в JavaScript є команди управління потоком обчислень, наприклад:

- break -примусовий вихід із циклу;
- continue - перехід в кінець циклу;
- for - цикл;
- if ... else - умовний оператор;
- will - умовний цикл;
- var - оператор змінної.

Важливим елементом мови є події. Програміст використовує події для виконання різних частин програмного коду скрипкової мови. До подій можна віднести наступні оператори:

- on Load - виконання коду або функції при завантаженні;
- on Submit - при натисненні на кнопку Submit;
- on Unload - при переході на іншу сторінку;
- on Change - при зміні значення елемента форми;
- on Click - при виборці об'єкта (button, checkbox і т.д.)

В новій версіях мови JavaScript значно розширені її можливості, за рахунок:

- можливість взаємодії Java та JavaScript;
- використання plug-ins;
- введення нових об'єктів (Area, Function, Image і т.д.)

На думку розробників цієї версії, це значно підвищить потужність програми на JavaScript. jQuery – бібліотека JavaScript, що фокусується на взаємодії JavaScript і HTML. Бібліотека jQuery допомагає легко отримувати доступ до будь-якого елементу DOM, звертатися до атрибутів і вмісту елементів DOM, маніпулювати ними. Також бібліотека jQuery надає зручний API по роботі з AJAX .

Для роботи з цією бібліотекою необхідно розмістити її в корінному каталозі системи і підключити її. Це можна зробити локально до певної сторінки або глобально до сторінки від якої унаслідуються інші.

Ця бібліотека має свою історію створення. HTML була однією з перших речей, яку Джон Резіг освоїв, коли він тільки почав займатися програмуванням. Резіг програмував на QBasic, коли один його знайомий показав йому, як створити веб-сторінку (використовуючи Angelfire), а також основи HTML. Батько подарував йому на Різдво дві книги по HTML. Саме тоді, коли він тільки почав програмувати на Visual Basic, HTML і веб-дизайн дуже зацікавили його.

Але пристрасть до JavaScript прийшла значно пізніше, приблизно в 2004 році. Тоді Резіг отримував ступінь в галузі комп'ютерних наук і працював на півставки в місцевій фірмі Brand Logic. Він займався дизайном сайту, в якому створювався користувальницький скролінг. Джон був розчарований і засмучений, особливо тому, що використовував код інших розробників, після чого вирішив серйозно вивчити JavaScript. Вивчивши, дійшов висновків, що JavaScript – це простий, але витончений мову, яка є неймовірно потужним для вирішення багатьох завдань. Протягом найближчого пари років Джон створив безліч різних JavaScript-додатків, перш ніж закінчити створення jQuery. Основною метою створення jQuery Резіг бачив можливість закодувати багаторазові шматки коду, які дозволять спростити JavaScript і використовувати їх так, щоб не турбуватися про крос-браузерних питаннях. Бібліотека була представлена громадськості на комп'ютерній конференції «BarCamp» у Нью-Йорку в 2006 році [7].

Web-технологія ASP.NET

ASP.NET з моменту свого створення середовище була принципово новою серверною технологією. Звичайно, в деяких випадках можна було використовувати ASP.NET для створення клієнтських сценаріїв Javascript, в основному це стосувалося елементів управління перевірки, а останнім часом – інфраструктури web-частин. Але в більшості випадків це було всього лише перетворенням серверних властивостей в клієнтську поведінку, і розробникові не доводилося піклуватися про взаємодію з клієнтом до отримання наступного POST-запиту. Для створення більш інтерактивних сторінок з клієнтськими сценаріями Javascript і DHTML розробникам доводилося діяти самостійно, використовуючи тільки функцію у відповідь викликів сценаріїв ASP.NET.

Технологія ASP.NET представляє собою active server pages (активні серверні сторінки) є основним механізмом розробки web-орієнтованих застосувань для серверів IIS (Internet Information Services). ASP були доповнені можливостями, які роблять більш легким застосування технології для розробників сценаріїв і web-додатків. Крім того, в архітектурі ASP відбулися ряд істотних змін, що дозволили поліпшити безпеку і продуктивність компонентів IIS (Internet Information Services) .

Дана технологія є універсальною платформою для розробки web-додатків корпоративного рівня. ASP.NET пропонує нову модель програмування і інфраструктуру, які дозволяють розробляти захищені і масштабовані рішення. У ASP.NET слід зазначити наступні функціональні можливості:

- можливості адміністрування. Параметри, що визначають конфігурацію ASP.NET, зберігаються в текстовому файлі. Це дозволяє виконувати зміну конфігурації ASP.NET за допомогою будь-якого текстового редактора. Всі проведені зміни набувають чинності негайно;
- засоби безпеки. ASP.NET пропонує розробникові web-додатків декілька типових схем авторизації користувачів. Розробник може використовувати в своєму застосуванні будь-яку з пропонованих схем або замінити їх іншими;
- простота розгортання. Розгортання ASP.NET-додатків виконується шляхом копіювання файлів додатку в спеціальну папку на web-сервері. Перезапуск web-сервера при цьому не потрібний;
- висока продуктивність. ASP.NET має справу з скомпільованим кодом. Завдяки цьому ASP.NET дістає можливість ефективно

використовувати різні механізми оптимізації коду (наприклад, механізми раннього скріплення або оптимізація під конкретну платформу);

- гнучке кешування. ASP.NET може виконувати кешування сторінок даних (як сторінку цілком, так і її частину) відповідно до потреб додатку;

- підтримка національних мов. Оскільки ASP.NET використовує Unicode, розробники мають широкі можливості для застосування в своїх застосуваннях національних алфавітів;

- підтримка мобільних пристроїв. ASP.NET підтримується будь-яким браузером, запущеним на будь-якому пристрої;

- доступність і масштабованість. ASP.NET розроблялася з розрахунку на використання в великих корпоративних системах. Реалізовані в рамках серверних механізмів дозволяють гарантувати високий ступінь доступності додатків. Якщо з процесом, в рамках якого виконується додаток, виникнуть проблеми, система самостійно запустить новий процес, який візьме на себе завдання обслуговування запитів користувачів;

- можливості відладки. ASP.NET забезпечує можливість трасування і відладки кодів додатків. При цьому можлива як локальна, так і віддалене налаштування за допомогою спеціальних інструментів відладки .NET Framework;

- інтеграція з .NET Framework. ASP.NET є частиною платформи .NET Framework. Розробники можуть використовувати можливості, що надаються цією платформою при створенні додатків;

- сумісність з існуючими ASP-додатками. Розгортання ASP.NET в рамках IIS-серверів не впливає на функціонування вже працюючих ASP-додатків. ASP-додатки можуть співіснувати на одному сервері, не заважаючи один одному.

ASP.NET спирається на багатомовні можливості .NET, що дозволяє писати код сторінок на VB.NET, Delphi.NET, Visual C#, J# і так далі .

Разом з ASP.NET існують і інші провідні технології і платформи розробки web-додатків: Java Server Pages (JSP) і web-платформа з відкритим кодом відома під назвою LAMP (Linux, плюс Apache, плюс MySQL, плюс Perl, Python або PHP як мова програмування).

Проте саме технологія ASP.NET дає змогу використовувати усі переваги мови програмування C# в поєднанні з зручним середовищем розробки Microsoft Visual Studio 2010, яка в собі має вже встановлені додатки для створення локального веб-серверу .

Microsoft Visual Studio – лінія продуктів компанії Майкрософт, що включають інтегроване середовище розробки програмного забезпечення і ряд інших інструментальних засобів. Дані продукти дозволяють розробляти як консольні додатки, так і додатки з графічним інтерфейсом, в тому числі з підтримкою технології Windows Forms, а також веб-сайти, веб-додатки, веб-служби як у рідному, так і в керованому кодах для всіх платформ, підтримуваних Microsoft Windows, Windows Mobile, Windows CE, .NET Framework, .NET Compact Framework і Microsoft Silverlight.

Це дає змогу швидко і доступно тестувати роботу розроблюваної системи. Мати доступ до інформації про стан змінних а також можливість відображення даних з підключених баз даних. Всі ці можливості разом із зручним користувацьким інтерфейсом виділяють розробку в середовищі ASP.NET з поміж інших середовищ розробки .

База даних MS SQL

Microsoft SQL Server – комерційна система керування базами даних, що розповсюджується корпорацією Microsoft. Мова, що використовується для запитів Transact-SQL, створена спільно Microsoft та Sybase. Transact-SQL є реалізацією стандарту ANSI/ISO щодо структурованої мови запитів (SQL) із розширеннями. Використовується як для невеликих і середніх за розміром баз даних, так і для великих баз даних масштабу підприємства. Багато років вдало конкурує з іншими системами керування базами даних.

Microsoft SQL Server як мову запитів використовує версію SQL, що отримала назву TRANSACT-SQL (скорочено T-SQL), яка є реалізацією SQL-92 (стандарт ISO для SQL) з багатьма розширеннями. T-SQL дозволяє використовувати додатковий синтаксис процедур, що зберігаються і забезпечує підтримку транзакцій (взаємодія бази даних з керуючим застосунком). Microsoft SQL Server та Sybase ASE для взаємодії з мережею використовують протокол рівня застосунка під назвою Tabular Data Stream (TDS, протокол передачі табличних даних).

Microsoft SQL Server також підтримує Open Database Connectivity (ODBC) інтерфейс взаємодії застосунків з СУБД. Версія SQL Server 2005 надає можливість підключення користувачів через веб-сервер-сервіси, що використовують протокол SOAP. Це дозволяє клієнтським програмам, не призначеним для Windows, кроссплатформенно з'єднуватися з SQL Server. Microsoft також випустила сертифікований драйвер JDBC, що дозволяє застосункам під керування Java (таким як BEA і IBM Websphere) з'єднуватися з Microsoft SQL Server 2000 і 2005 .

SQL Server підтримує дзеркалювання та кластеризацію баз даних. Кластер серверу SQL – це сукупність однаково конфігурованих серверів; така схема допомагає розподілити робоче навантаження між декількома серверами. Усі сервера мають одне віртуальне ім'я, а дані розподіляються за IP-адресами машин кластеру протягом робочого циклу. Також у разі відмови або збою на одному з серверів кластеру доступне автоматичне перенесення навантаження на інший сервер.

3 РОЗРОБКА ВЕБ-СТОРІНКИ ДЛЯ ELIT TENNIS CLUB

3.1 Розробка архітектури

Для розробки архітектури інтерактивного web-сервісу «Мережа спортивно оздоровчого клубу» було вирішено використовувати клієнт-серверну архітектуру. Обрана архітектура клієнт – сервер широко застосовується при роботі з базами даних у мережі і передбачає взаємодію та обмін даними між ними. Архітектура клієнт-сервер передбачає такі три основні компоненти:

- набір серверів, які надають інформацію або інші послуги програмам, які звертаються до них;
- набір клієнтів, які використовують сервіси, що надаються серверами;
- мережа, яка забезпечує взаємодію між клієнтами та серверами.

Обробка та збереження даних відбувається на стороні сервера, відображення даних і надсилання запитів на сервер виконується на стороні клієнта. На рисунку 3.1 зображено трирівневу схему архітектури web-додатку

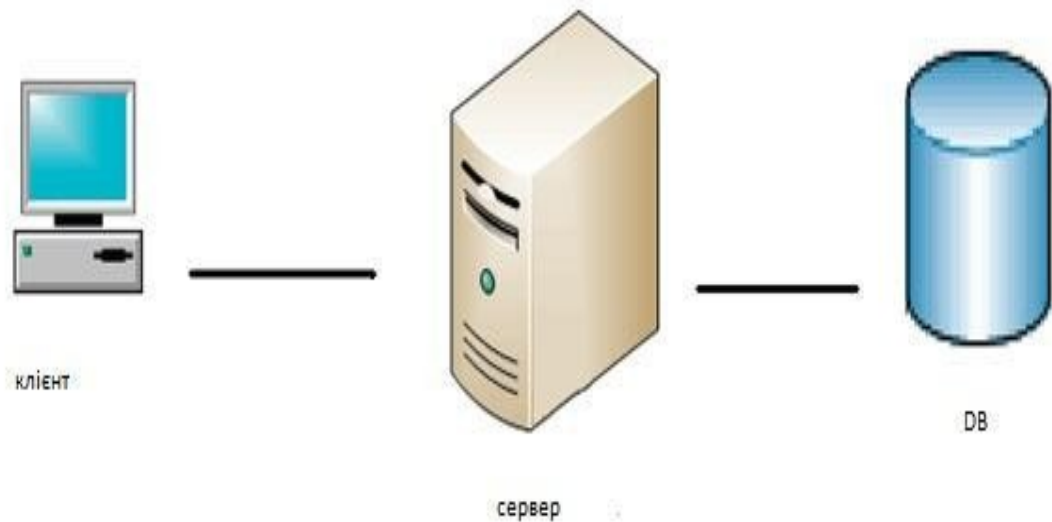


Рисунок 3.1 – Клієнт серверна архітектура web-додатку

Перший рівень – це клієнт, який відсилає запити на сервер та приймає результат обробки запитів. Клієнтом є браузер користувача.

Другий рівень – бізнес-логіка додатку. Це логіка за якою web-сервер обробляє отримані від клієнта запити.

Третій рівень – це сама СУБД, яка отримує запити від сервера і повертає потрібні дані на сервер або зберігає їх. Діаграма класів відображає класи спроектованої системи і відношення між ними. Для відображення основної бізнес-логіки розроблюваної системи побудовано UML діаграму класів, яка зображена на рисунку 3.2.

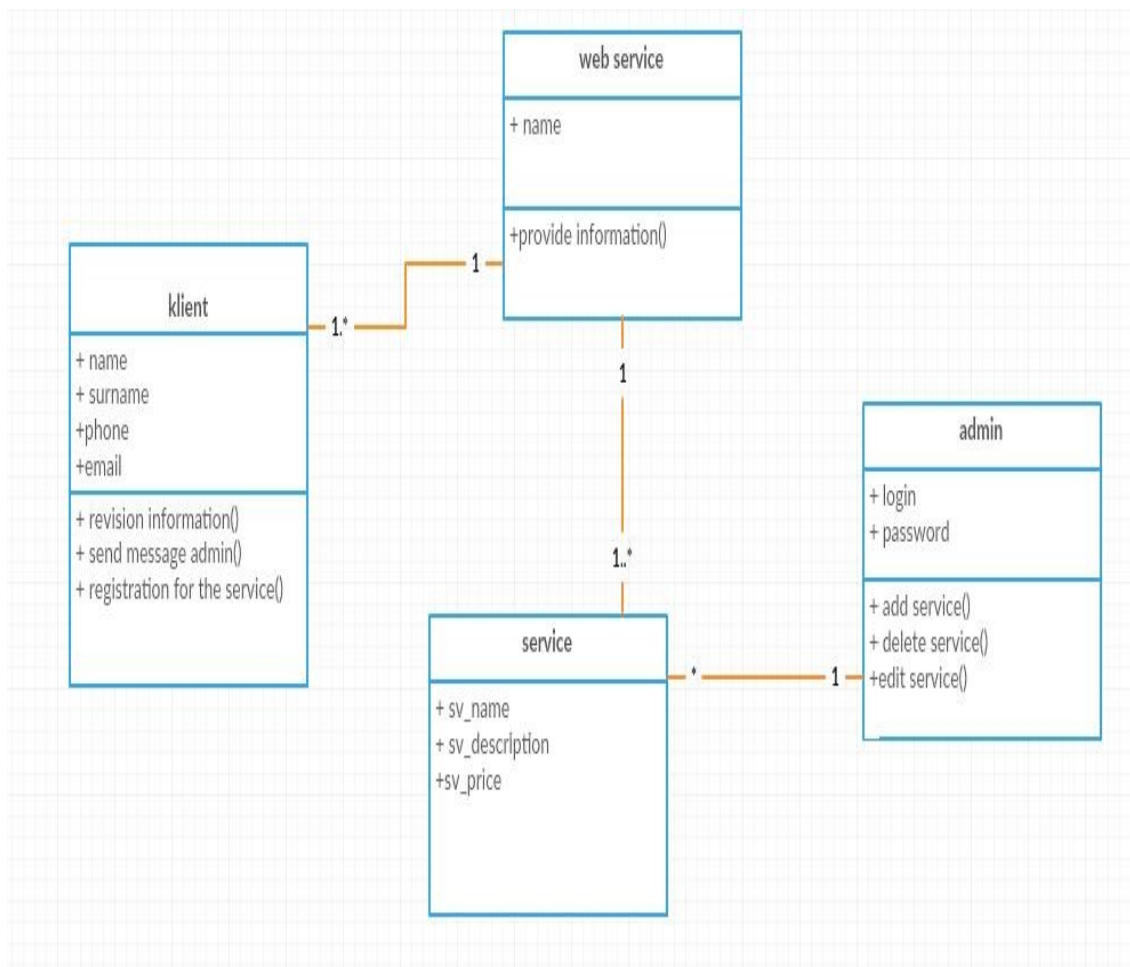


Рисунок 3.2 – Діаграма класів

Для того щоб знати в якому стані перебуває система потрібно використовувати діаграму станів.

Діаграма станів представляє собою те, що кожен об'єкт системи, який володіє певною поведінкою, може знаходитися в певних станах, переходити із стану в стан, здійснюючи певні дії в процесі реалізації сценарію поведінки об'єкту. На діаграмах 3.3-3.5 зображено діаграми станів для основних сценаріїв.

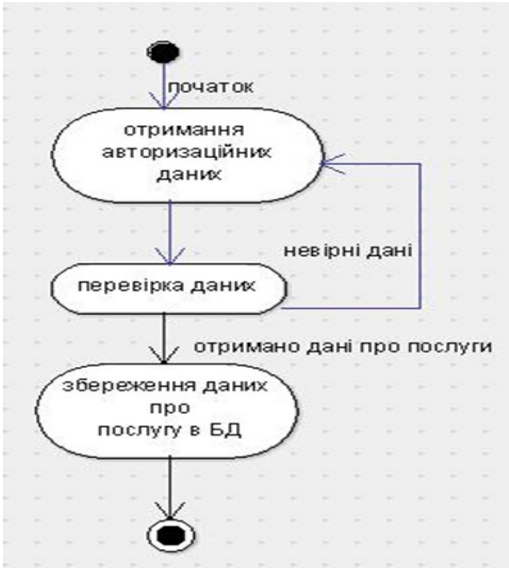


Рисунок 3.3 – Діаграма станів для процесу дії над послугами

Система отримує авторизаційні дані від адміністратора та перевіряє їх. Після успішної авторизації система очікує на дані про послуги, перевіряє їх та зберігає в базі даних.



Рисунок 3.4 – Діаграма станів для процесу запису на послугу

Система отримує введені дані клієнта. Клієнт обирає послугу та відділення і підтверджує надсилання листа.

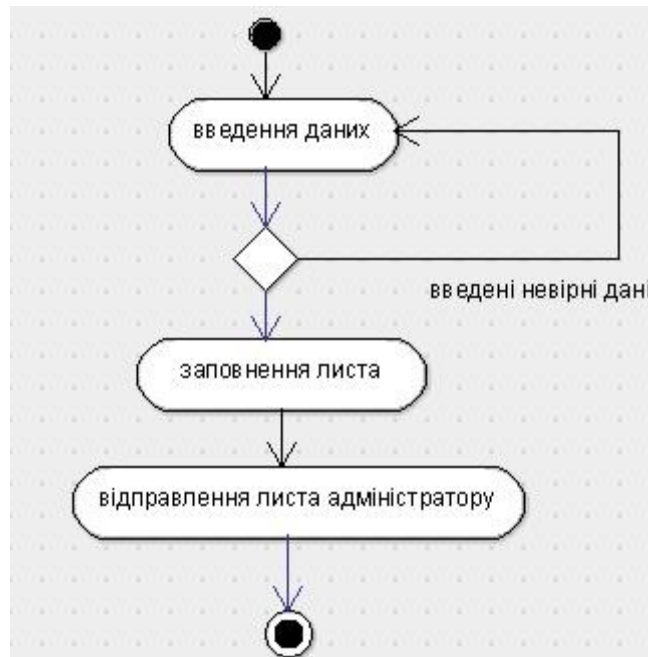


Рисунок 3.5 – Діаграма станів для процесу зв'язку з адміністратором

Клієнт вводить свої дані, система перевіряє чи коректно вони введені. Після введення своїх даних клієнт вводить текст листа. Ввівши текст листа клієнт надсилає його адміністратору.

3.2 Розробка дизайну додатку

Було зроблено головну сторінку сервісу, сторінку портфолію, сторінку з інформацією про послуги.

Приклад розмітки головної сторінки наведено нижче.

Підключення стилів файлу стилів:

```

<link href="layout/styles/layout.css" rel="stylesheet" type="text/css"
media="all">
<li class="one_third">
<article class="overlay"><a href="#"><i class="fa fa-line-chart"></i></a>
<h6 class="heading">Центр на вулиці За Рудкою</h6> <p>Знаходимось:</p>
<p>м.Тернопіль вул. За Рудкою</p>
<p>16/б</p>
<p>+0(98)2548613 </p>
</article>
  
```

```

</li>
</ul>
<!-- ##### -->
</section>
</div>
</div>
<!-- ##### -->
<div class="wrapper row3">
<main class="container clear">
<!-- main body -->
<div class="group">
<div class="one_third first"></
div> <div class="two_third">
<h2 class="heading">Про Нас</h2>
<p>

```

Основні принципи нашої роботи: доброзичливе ставлення до гостей і членів клубу, першокласний сервіс і всебічна підтримка наших клієнтів у досягненні бажаного результату на тренуваннях.

```
</p>
```

```
<p class="btmspace-80">Наш клуб є ідеальним для успішних людей
котрі цінують свій час. Для вас ми пропонуємо оптимальний перелік послуг
для підтримання міцного здоров'я і фізичної форми.
```

```
</p>
```

```
<p>Ми надаємо:</p>
```

Для адміністратора було розроблено адмін панель, результат наведено на рис. 3.4, за допомогою якої адміністратор, зможе оперувати послугами, які є в базі а саме:

- створювати нову послугу;
- редагувати існуючу послугу;
- видаляти послугу.

Клієнт вводить свої дані, система перевіряє чи коректно вони введені. Після введення своїх даних клієнт вводить текст листа. Оформлюється на ту послугу, котра його інтересує.

На рисунку 3.4 зображено дану панель адміністратора.

Послуги

Додати послуги

Показати 10 позицій

Search:

Номер	Назва	Опис	Ціна	Редагувати	Видалити
1	масаж		120-150		
2	Тренажерний зал	для дорослих	150		
3	Басейн	для дорослих	180		

Previous 1 Next

Рисунок 3.4 – Панель адміністратора

За допомогою jquery було додано код в створений файл, який відображає дані з таблиці `service`.

3.3 Розробка функціональних можливостей

Для розробки моделі бази даних обрана реляційна модель даних. Вона найкраще підходить для вирішення цієї задачі, адже вона має ряд наступних переваг:

- незалежність програм від даних. Ідея використання баз даних та систем управління базами даних передбачає використання додаткового рівня між прикладними програмами та власне даними, завдяки чому прикладні програмісти можуть абстрагуватися від реалізації самої бази даних, а зосередити свою увагу на логіці обробки даних;
- простота розробки та моделювання інформаційного ресурсу як плата за деякі обмеження та уніфікацію на рівні реалізації операцій над даними;
- наявність умов керування даними за допомогою операцій над множинами.

В процесі проектування структури бази даних потрібно створити діаграму корпоративної моделі даних.

В діаграмі корпоративної моделі даних графічно представлені зв'язки між об'єктами, з яких складається система. Діаграма корпоративної моделі даних представлена на рисунку 3.6.

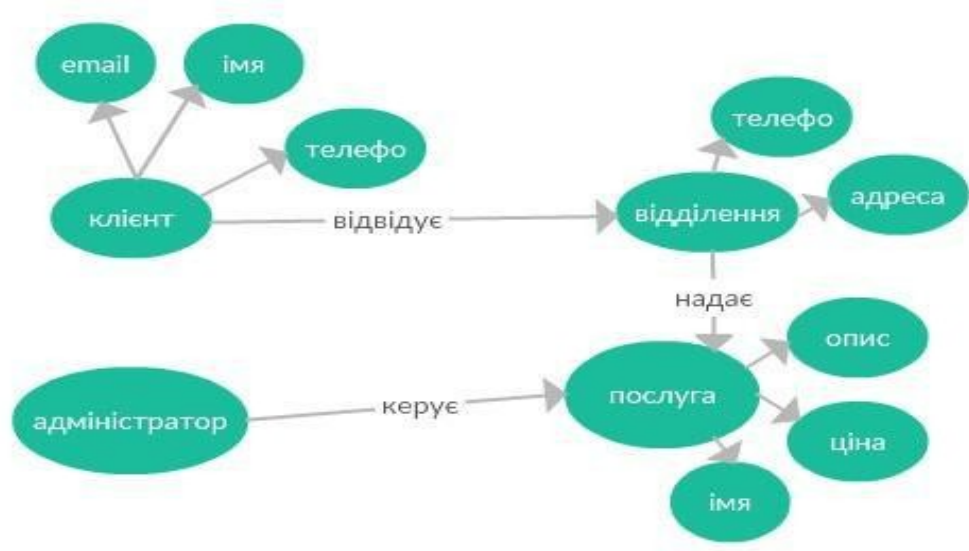


Рисунок 3.6 – Діаграма корпоративної моделі даних

ERD (Діаграма сутність-зв'язок) включає сутність і взаємозв'язки, що відображають основні бізнес-правила предметної області. Така діаграма не дуже деталізована, в неї включаються основні сутності і зв'язки між ними, які задовольняють основним вимогам, що пред'являються до системи. ER – діаграма розроблюваного сервісу зображена на рисунку 3.7.



Рисунок 3.7 – Діаграма розроблюваного сервісу

На ER – діаграмі зображено наступне відношення: service. Функціональна модель предметної області бази даних представлена у вигляді діаграми потоків даних.

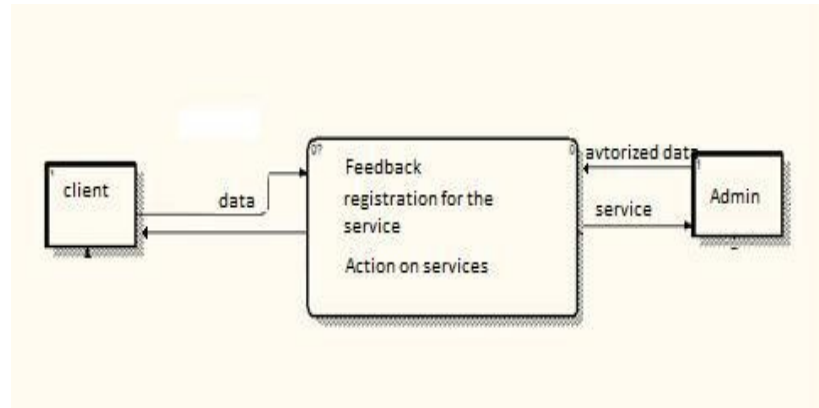


Рисунок 3.8 – Модель DFD

На рисунку 3.9 зображено діаграму декомпозиції процесу роботи з послугою.

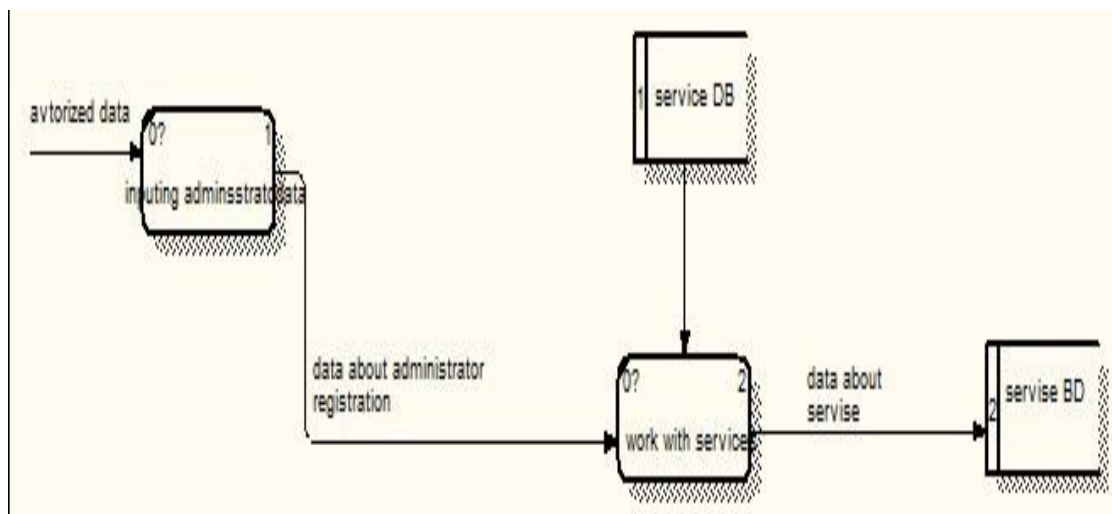


Рисунок 3.9 – Діаграма декомпозиції процесу роботи адміністратора з послугами

На рисунок 3.9 видно що на вхід до моделі подається один потік даних і один на виході. На вхід подаються авторизаційні дані користувача системи. Після проходження авторизації починається процес дій над послугами. У фізичне проектування бази даних входить створення таблиці у відповідній БД, згідно ER - діаграми. Для цього було обрано СУБД MySQL.

Нижче наведено DDL код для створення таблиць в БД.

```
CREATE TABLE service
(serviceID integer(10) NOT NULL AUTO_INCREMENT,
sv_name varchar(20) NOT NULL,
sv_description varchar(55),
sv_price varchar(30),
PRIMARY KEY (serviceID)
);
```

3.4 Апробація роботи

Тестування процес який застосовується для перевірки чи відповідає розроблена система заданим специфікаціям.

Метою проведення функціонального тестування є знаходження невідповідностей між розробленою системою і очікуваним результатом.

Для того, щоб провести функціональне тестування необхідно розробити тестові випадки, що містить такі дані:

- id – ідентифікатор тестового випадку;
- summary – опис об'єкту тестування;
- expected result – очікуваний результат;
- passed/failed – показує чи тест пройшов чи провалився;
- pre-condition – умова, яка необхідна для виконання, щоб виконати тестовий випадок;
- steps to reproduce – шлях проходження тестового випадку.

Було розроблено 6 тестових випадків які успішно були пройдені.

Для того щоб GUI тестування було більш ефективним, використовують автоматизоване тестування інтерфейсу. Автоматизоване тестування проводилося за допомогою програмного продукту Selenium.

Розроблений програмний продукт є веб-сервісом, то для тестування було встановлено плагін Selenium IDE для браузеру Mozilla Firefox. Усе це відображено у таблиці усі шість випадків. Тести проводились на практичних заняттях, в період проходження переддипломної практики. Практика

проводилась на робочому місці у Компанії Еліт Теніс Клуб. Получені данні були записані в наступну таблицю.

А також будут використовуватися данною компанією на практиці.

Таблиця 3.1 – Тестових випадків які успішно були пройдені

Id	Summary	Steps to reproduce	Pre - condition	Expected result	Passed/Failed
1	2	3	4	5	6
1	Verify ability of user to visit portfolio page.	1. choosing link “Портфоліо і послуги” 2. Choose link and click this linck	1. User visite some page.	1.user click a linck and go to portfolio page	Passed
2	Verify ability of user to visit page with service list.	1. choosing link “Портфоліо і послуги” 2. Choose link and click this linck	1. User visite some page..	1.user click a linck and go to portfolio page	Passed
3	Verify ability of user return to home page	1. choosing link “Домашня” 2. Choose link and click	1. User visited some page	1.user click a linck and go to home page	Passed

		this linck			
Продовження табл. 3.1					
1	2	3	4	5	6
4	Verify ability of administrator visited admin page	1. Enter valid Login. 2. Enter valid Password 3. Click on Sign in button	1. User is located on the Log in page. 2. User has an account.	1. User is successfully logged in to account. 2. System forward admin to the admin page.	Passed
5	Verify ability of administrator add new service	1.Click “add” button 2. enter data about new service	1.visite the administrator page	1. admin add new service	Passed
6	Verify ability of administrator delete service	1.Click “delete” button 2. choos servise 3.delete	1.visite the administrator page	1. admin delete service	Passed

--	--	--	--	--	--

Результат проведеного тестування користувацького інтерфейсу за допомогою Selenium IDE представлено на рисунку 3.5.

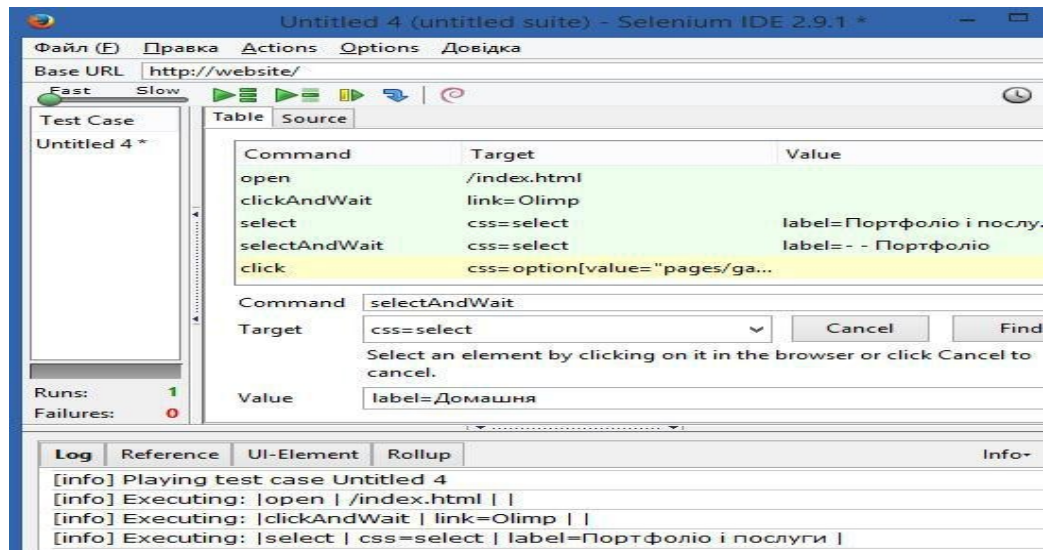


Рисунок 3.5 – Тестування в Selenium IDE

3.5 Інструкція для користувача

Для того щоб користуватися даним веб сервісом користувачу потрібен комп'ютер, підключення до мережі інтернет і веб-браузер. Для того щоб переглянути інформацію про центр його робочий графік користувачу потрібно відвідати веб-сайт, і перейти по силці на необхідну сторінку.

Наші послуги

Тренажерний зал

В клубі представлені різні види професійних тренажерів для кожної групи м'язів. Зона вільних ваг укомплектована широким асортиментом гантелей, грифів і дисків відповідальним, всім сучасним вигодами. Широкий вибір ваги дозволяє займатися як новачку так і досвідченому спортсмену.

Заняття в тренажерному залі допоможуть вам покращити всі фізичні якості, скорудити або набрати вагу, зміцнити здоров'я. А наші висококваліфіковані тренери зроблять процес досягнення ваших мрій швидким та безпечним!

Кордбо

Кордбо – заняття необхідні для тренування серцево-судинної системи, розвитку витривалості, є незамінним для тих хто прагне покращити свою форму. Кордболей нашої залу мають наступні тренери: – Білозір доронки (вертикального дизайну, зручні у користуванні) – Горизонтальний еліпсоїдальний (масову підтримку для сліди).

Масаж

Це комплексний підхід до здоров'я і краси кожної людини. Масаж допомагає вам у досягненні цілей. Доводить пом'якшується, робить здорову і красиву професіоналам і просто розслабляється. Відправляє сонні від краєвих спеціальних масажерів. Приплив сил, м'язовий тонус і неспокійне відчуття легкості знайдеться без напружених зусиль з вашого боку.

Наші послуги

120-150грн (масажи)
100-130грн (Тренажерний зал)

Працюємо

Дні	Час
Понеділок-П'ятниця	9-21
Субота	9-20
Неділя	9-19

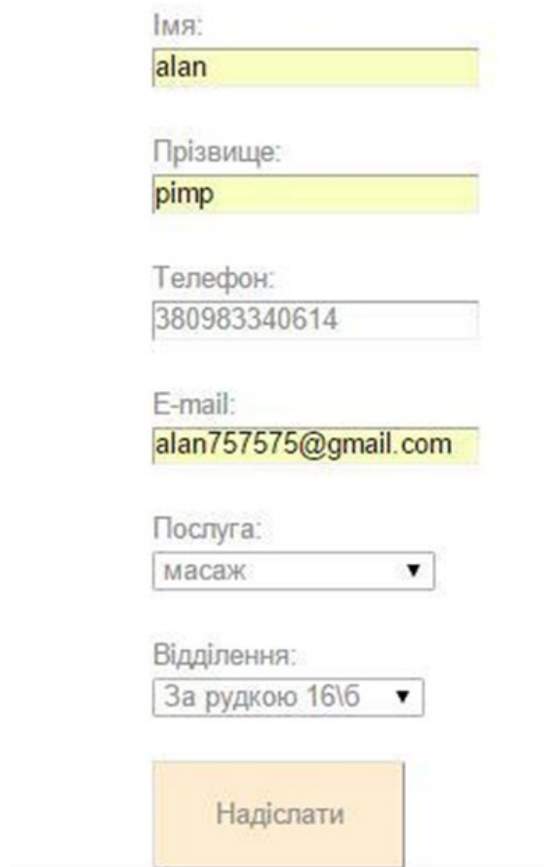
Рисунок 3.6 – Сторінка з послугами і робочим часом

Для того щоб записатися на прийом користувачу необхідно ввести свої дані:

- Ім'я;
- Прізвище;
- Електронну адресу на яку прийде лист.

Вибрати бажану послугу і центр в якому вона буде надаватися.

Даний процес зображено на рисунку 3.7.



Ім'я:
alan

Прізвище:
pimp

Телефон:
380983340614

E-mail:
alan757575@gmail.com

Послуга:
масаж ▼

Відділення:
За рудкою 16\6 ▼

Надіслати

Рисунок 3.7 – Запис на послугу

У випадку якщо користувач хоче зв'язатися з адміністратором він може написати йому лист скориставшись відповідним функціоналом сайту. Для цього потрібно перейти на сторінку «зв'язатися з нами» і заповнити необхідні форми. Результат виконання зображено на рисунку 3.8.

Відвідувач сайту має можливість переглянути фотографії центру. Це надасть йому уявлення про центр, як він виглядає, його послуги, обладнання центру. Портфоліо центру зображено на рисунку 3.10



Ім'я:
alan pimp

E-mail:
alan757575@gmail.com

Телефон:
0983340614

Лист:
hello world

Надіслати

Рисунок 3.9 – Лист адміністратору

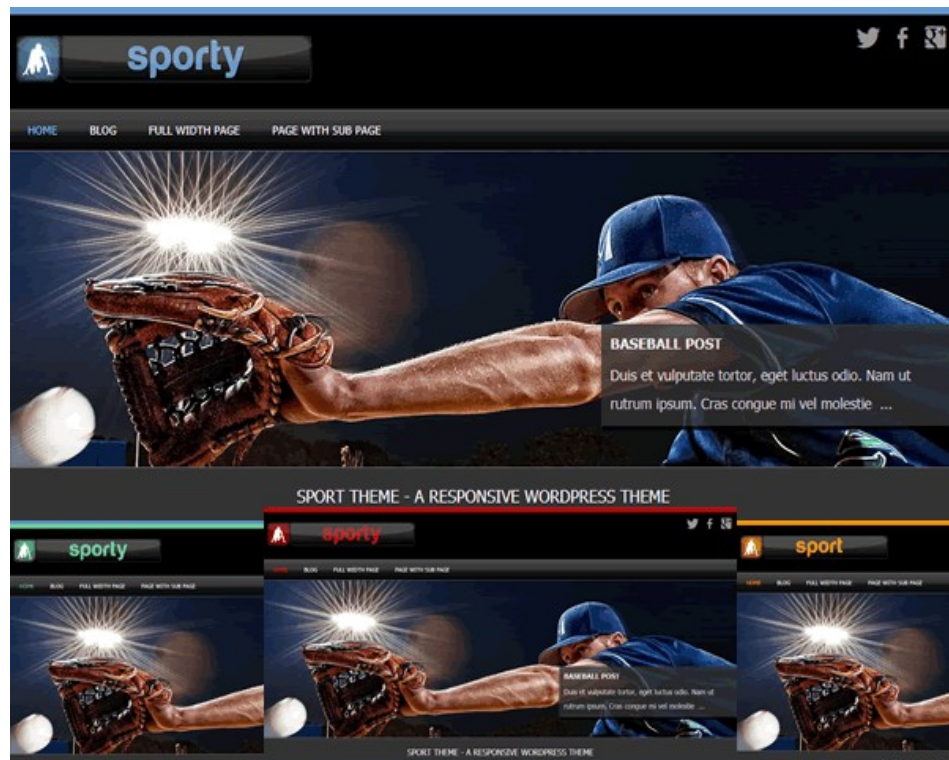


Рисунок 3.10 – Портфоліо сайту

Щоб оперативно додавати, видаляти послуги адміністратор має свою сторінку з реалізованим інтерфейсом. Що дозволяє йому керувати даними таблиці. На рисунку 3.11 зображена сторінка адміністратора з послугами.

Послуги

Додати послуги

Показати 10 позицій

Search:

Номер	Назва	Опис	Ціна	Редагувати	Видалити
1	масаж		120-150		
2	Тренажерний зал	для дорослих	150		
3	Басейн	для дорослих	180		

Previous 1 Next

Рисунок 3.11 – Сторінка адміністратора

Для того щоб мати можливість адміністратору працювати з послугами він повинен авторизуватися в системі ввівши свій логін і пароль. Процес авторизації зображено на рисунку 3.12.

Вхід

User

redbool

Password

Log-In

Рисунок 3.12 – Авторизація

Пройшовши процес авторизації адміністратор може працювати з послугами а саме: додавати послугу, видаляти послугу, редагувати її.

На рисунку 3.13 зображено процес додавання нової послуги.

Послуги

👁️ Переглянути послуги

Назва послуги	масаж
Опис	
Ціна	120-150
+ Зберегти цей запис	

Рисунок 3.13 – Додавання послуги

ВИСНОВКИ

У даному дипломному проєкті було проаналізовано предметну область розроблюваного продукту. Розглянута структура об'єкту дослідження, досліджені актори та їх можливості в системі. Були проаналізовані переваги та недоліки програмних продуктів з аналогічною предметною областю. Інформація отримані при аналізі будуть враховані при розробці програмного продукту.

Була розроблена архітектура web-додатку. Спершу було обрано одну із загальних архітектур розробки програмного забезпечення, яка найкраще підходила для вирішення поставленої задачі. Для кращого розуміння архітектури продукту було розроблено діаграми, які показують взаємодію користувачів та їх функцій між собою та у середині системи.

Проаналізовано потоки даних у системі. Усі ці потоки були представленні на діаграмі потоків даних.

Було проведено функціональне тестування розробленого програмного продукту. Проведено автоматизоване тестування користувацького інтерфейсу з використанням плагіну Selenium IDE для браузера Mozilla Firefox. Що дозволило перевірити розроблений програмний продукт на недоліки. Описано розгортання програмного продукту та створено інструкцію для користувача.

ПЕРЕЛІК ПОСИЛАНЬ

- 1.Буч Г., Рамбо Д., Якобсон И.
Б90 Язык UML. Руководство пользователя. 2-е изд.: Пер. с англ. Мухин Н. – М.: ДМК Пресс, 2006. – 496 с.: ил.
- 2.Вендров А.М. CASE-технологии. Современные методы и средства проектирования информационных систем. – М.: Финансы и статистика. – 2000.
- 3.Голицина О. Л. Основы алгоритмизации и программирования /О.Л.Голицина, И.И.Попов// Учебное пособие. – М.: ФОРУМ: ИНФРА-М. – 2005.
- 4.Дивак М.П. Системний аналіз та проектування КІС /М.П.Дивак// Навчальний посібник – Тернопіль: Економічна думка. – 2004.
- 5.Мэтт Зандстра. РНР: объекты, шаблоны и методики программирования, 3-е издание = RHP Objects, Patterns and Practice, Third Edition. – М.: «Вильямс», 2010. – С. 560
- 6.Синицын С. В., Налютин Н. Ю. Верификация программного обеспечения. – М.: БИНОМ, 2008. – 368 с