

МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ
Західноукраїнський національний університет
Факультет комп'ютерних інформаційних технологій
Кафедра економічної кібернетики та інформатики

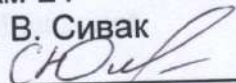
СИВАК Юлія Валеріївна

**Розробка веб-додатку для автоматизації
управління процесом надання послуг клієнтам
спортивного клубу.**
**Development of a web-application to automate the
providing services to customers of the sportsclub.**

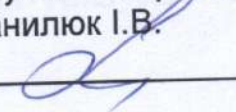
спеціальність: 124 - Системний аналіз
освітньо-професійна програма - Системний аналіз

Кваліфікаційна робота

Виконала студентка групи
САМ-21
Ю. В. Сивак



Науковий керівник:
Данилюк І.В.



Кваліфікаційну роботу
допущено до захисту:

" " _____ 20__ р.

Завідувач кафедри

Л. М. Буяк

ТЕРНОПІЛЬ - 2022

ЗМІСТ

ВСТУП.....	3
РОЗДІЛ 1. Технічне обґрунтування необхідності розробки веб – додатку для автоматизації управління персоналом наданих послуг клієнтам спортивного клубу.....	6
1.1. Поняття веб – додатку, переваги, хмарні технології.....	6
1.2. Характеристика об’єкта управління.....	9
1.3. Обґрунтування вибору проектних рішень щодо діяльності спортивного клубу.....	12
1.4. Рішення з інформаційного забезпечення діяльності спортивного клубу..	15
Висновки за розділом 1.....	32
РОЗДІЛ 2. Проектування системи управління процесом надання послуг клієнтам спортивного клубу.....	33
2.1. Рішення технічного забезпечення.....	33
2.2. Розроблення архітектури програмного забезпечення.....	36
2.3. Проектування структури бази даних.....	42
Висновки за розділом 2.....	50
РОЗДІЛ 3. Програмна реалізація веб – додатку для автоматизації управління процесом надання послуг клієнтам спортивного клубу.....	53
3.1. Програмна реалізація системи.....	53
3.2. Призначення та умови застосування програмної системи.....	63
3.3. Технологічний процес збору та обробки даних на периферійних пристроях при централізованій обробці даних.....	66
Висновки за розділом 3.....	67
СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ.....	72
ДОДАТКИ.....	78

ВСТУП

Актуальність дослідження. Автоматизація процесів управління як засобу підвищення продуктивності праці завжди була та залишається актуальною.

Сьогодні важко уявити суспільство без використання інформаційних технологій. Необхідність застосування останніх не викликає сумнівів, оскільки технологія – одна з розвинутих галузей сучасного життя.

Глобальна комп'ютерна мережа Інтернет, її вплив на сучасний світ просто не має історичних аналогів. Він проникає у всі сфери суспільного життя. Немає жодного підприємства, жодної людини, які б не скористалися Інтернетом.

Веб – технологія також змінила уявлення про роботу як з інформацією так і з комп'ютером. Коли інтерфейс між людиною та комп'ютером був спрощений до сприйняття звичайною людиною, вагомо зросла зацікавленість до використання інформаційних технологій як у всіх видах діяльності. Так і у повсякденному житті.

У дослідженнях зарубіжних та вітчизняних науковців можна спостерігати зацікавленість до віртуальних предметних спільнот, що підтверджено зростаючою кількістю та впровадженням новітніх технологій для підтримки їх діяльності. Що стосується досліджень у даному напрямку, вони відображені у працях М.І. Жалдака, Н.Т. Задорожної, В.М. Кухаренко, І.Д. Малицької, Н.В. Морзе, Є.Д. Патаркіна, Є.С. Полат, А.В. Хуторського, С. Віркуса, Д. Боудена та ін.

Процес розробки веб – додатків набуває подальшої актуальності, оскільки останній спрямований на покращення роботи користувача (надання інформації в будь – який час, з будь – якої відстані, надання потрібних даних, економія часу, економія ресурсів). Тому, використання веб – додатку для автоматизації управління процесом надання послуг клієнтам спортивного клубу на даний момент є актуальним.

Мета та завдання дослідження полягають в розробці веб – додатку для автоматизації управління процесом надання послуг клієнтам спортивного клубу.

Для досягнення поставленої мети передбачено реалізацію таких **завдань**:

- розглянути поняття веб – додатку;
- дослідити питання проектування інформаційних систем;
- здійснити огляд та аналіз інформаційних систем обліку послуг, що представлені на ринку України;
- розробити веб – додаток для організації діяльності працівників у спортивному клубі «SPA- Фітнес».

Об’єкт дослідження. Організаційна структура та діяльність спортивного клубу.

Предмет дослідження. Засоби розробки веб – орієнтованої системи для організації діяльності спортивного клубу.

Методи дослідження. В роботі використані загальнонаукові методи пізнання, наукові праці вітчизняних і закордонних вчених, а саме: методи теоретичного узагальнення, системного аналізу (для критичного аналізу розвитку економіки); системний підхід, методи аналізу та синтезу (для обґрунтування концепції розробки веб – сайтів); системно-структурний аналіз (з метою розробки механізму формування кумулятивної стратегії розвитку бізнесу); багатовимірний статистичний аналіз (для виявлення ретроспективних і перспективних тенденцій розвитку бізнесу); методи багатокритеріального оцінювання та сценаріїв майбутнього, тенденцій їх розвитку.

Вихідними даними для дослідження є інформація наукових установ, статистичні дані, опубліковані в періодичних виданнях, дані Державного комітету статистики України, бізнесу тощо.

Наукова новизна одержаних результатів. З метою реалізації даної роботи був обраний в якості інструменту програмний продукт WPwin фірми Platinum technology/Logic Works.

Під час моделювання підсистеми використані нотації SADT (Structured Analysis and Design Technique) і DFD (Data Flow Diagrams), завдяки яким найбільш адекватно відображено логіку предметної області та сприйняття для проведення аналізу бізнес – процесів.

Практичне значення одержаних результатів. Розроблений веб – додаток дозволить використовувати його в майбутньому для здійснення управління процесом надання послуг клієнтам спортивного клубу за допомогою мережі Інтернет.

Апробація результатів роботи. Результати роботи опубліковані у Збірнику тез міжнародної науково – практичної Інтернет конференції «Світ наукових досліджень. Випуск 13» , м. Тернопіль, від 25 жовтня 2022 року, С. 243-245.

Структура та обсяг роботи. Робота складається зі вступу, трьох розділів, висновків, списку використаних джерел та додатків. Загальний обсяг роботи складає 96 сторінок. Робота містить 5 таблиць, 18 рисунків, 5 додатків. Список використаних джерел нараховує 49 найменувань.

РОЗДІЛ 1

ТЕХНІЧНЕ ОБГРУНТУВАННЯ РОЗРОБКИ ВЕБ - ДОДАТКУ ДЛЯ АВТОМАТИЗАЦІЇ УПРАВЛІННЯ ПЕРСОНАЛОМ НАДАНИХ ПОСЛУГ КЛІЄНТАМ СПОРТИВНОГО КЛУБУ

1.1. Поняття веб – додатку, їх переваги, хмарні технології

Застосування та розвиток інформаційних продуктів є новим основним засобом організації послуг клієнтам спортивного клубу, а тому потребують узагальнень та розробки практичних рекомендацій на всіх рівнях.

Сьогодні спостерігається тенденція до збільшення Інтернет користувачів, у тому числі, користувачів мобільними пристроями (телефони, смартфони, планшети та ін.). Користувачі персональних комп'ютерів ознайомлені з однією з програм, яка працює в операційному середовищі ОС Windows.

Розглянемо, що з себе уявляє веб – додаток. За даними Вікіпідії – «...розподілений застосунок, в якому клієнтом виступає браузер, а сервером – веб – сервер» [11].

«WEB – додатки – прикладна програма, розроблена з архітектури «клієнт – сервер» і використовує в якості клієнта web – браузер та працює на стороні web – сервера» [49]. Дані програми написані або скриптовою мовою (PHP, Perl та ін.), або мовою високого рівня та відкомпільовані під операційну систему і призначені для створення інтерфейсу між користувачем та веб – сайтом.

Можна зробити узагальнення, що веб – додаток – комп'ютерна програма, що працює в браузері (так саме Microsoft Word в операційній системі Windows). Щоб отримати доступ до програми повинні бути у наявності браузер та Інтернет. При такій організації обробка та збереження інформації відбувається у віддаленому сервері, веб - переглядач – програмою – клієнтом, інтерфейс призначено для користувача.

Отже, для текстового пакету Microsoft Office буде працювати під Windows. Операційною системою та користувацьким інтерфейсом веб – додатку є браузер, тому для нього остання не має значення.

Веб – додаток відкриває унікальні можливості для користувача працювати зі своєю програмою на будь – якому зі своїх девайсів (від стаціонарного комп'ютера до персонального смартфона).

Великою перевагою веб – додатку є те, що його може використовувати людина без спеціальної освіти, оскільки вже розроблена велика кількість сервісів та програм, які можна просто застосовувати, вдосконалювати, крім того ще й безкоштовно.

В якості прикладу можемо навести один з веб - додатків для читання новин – Bloglines, який конкурує з іншими додатками. Або, всім відомий додаток Gmail, який має багато чисельні додаткові функції та є повноцінним поштовим клієнтом. Зазначені додатки працюють на сервері, а їх (UI) користувацький інтерфейс відображено в якості веб – сторінок.

«...Хмарні технології надають користувачам мережі Інтернет доступ до комп'ютерних ресурсів сервера, а також використання програмного забезпечення в якості онлайн – сервісу» [46]. Основне їх призначення – задоволення потреб користувачів, які повинні опрацювати інформацію віддалено.

До хмарних технологій відносять, знов такі, сервіси електронної пошти: Gmail, Google, Meta та ін. Працювати з ними можна при наявності Інтернету.

Постачання послуги хостингу через інтернет – це хмарні технології (загальний термін для всього). Послуги можна класифікувати на такі категорії (Рис. 1.1).

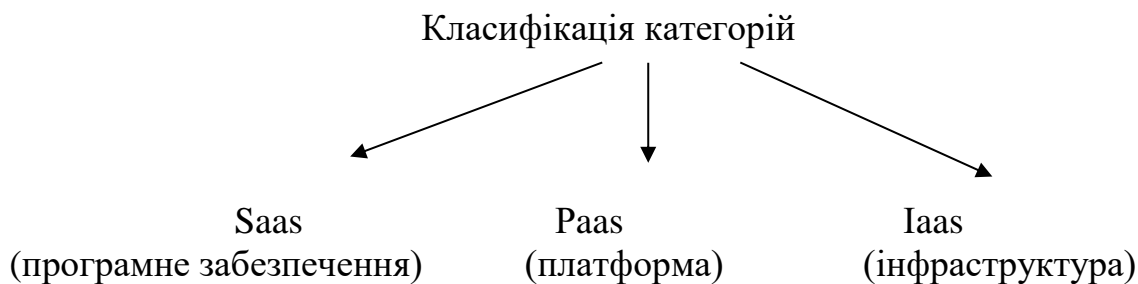


Рисунок 1.1. Види категорій

Характеристика категорій відображена в таблиці 1.1.

Таблиця 1.1.

Характеристика категорій [40]

Назва послуги	Характеристика
Програмне забезпечення як послуга (Saas)	«...За даною моделлю постачається апаратна інфраструктура та програмне забезпечення. Розробник забезпечує взаємодію через інтерфейс ний портал з користувачем. Перевага: модель досить розповсюджена; може надавати різноманітні послуги (починаючи від управління запасами до обробки бази даних); послуга доступна не зважаючи на відстань».
Платформа – як – сервіс (Paas)	«...Набір програмних продуктів, засобів розробки (розміщені на інфраструктурі провайдера). Розробники можуть створювати програми, а також використовувати сайт – портали, АРІ, шлюзи, ПЗ, що встановлене на комп'ютері у клієнта через Інтернет».
Інфраструктура як послуга (Iaas)	«...Віртуальний сервер instanceAPI, що слугує для доступу, запуску, зупинки та налаштування і своїх віртуальних серверів, і систем збереження. Сервер дозволяє компанії платити рівно стільки, скільки використано потужностей. Ще вона має назву «комінікальні обчислення».

Отже, приходимо до висновку, що розробка веб – додатків, в умовах сьогодення, є досить актуальним, оскільки зазначена технологія розвивається швидкими темпами, має суттєві переваги щодо економії часу, ресурсів, не обмежує користувача у виборі місцезнаходження для застосування своїх можливостей, можна працювати з будь – яких девайсів не залежно від операційної системи чи її технічних характеристик.

1.2. Характеристика об'єкта управління

Об'єктом дослідження магістерської роботи є процес управління діяльністю спортивного клубу, що передбачає розробку автоматизованої системи для введення обліку даними, організації діяльності співпраці із клієнтами, організація автоматизованої звітності діяльності фітнес клубу. Перш ніж розробляти програмне забезпечення слід визначити мету діяльності об'єкта, що вибраний для конструювання. Проаналізувавши різні джерела, ми прийшли до висновку, що мета діяльності клубу є створення умов та сприяння у розвитку різних можливостей для задоволення оздоровчих потреб його відвідувачів, зокрема фізичної культури, спорту, індивідуальних спортивних процедур, релаксації.

Реалізація даної мети може бути представлена у вигляді наступних організаційних структур: спортивні секції, фітнес-клуби, фізкультурно-спортивні організації. Серед всі вище наведених можливостей реалізації спортивних клубів слід виділити «фітнес-клуб», оскільки даний вид організації діяльності членів клубу забезпечує розвиток фізичної культури і спорту клієнта та дозволяє надавати індивідуальні послуги, щодо його оздоровлення у відповідності з потребами та можливостями їх оплати.

Основними видами послуг, які надаються клієнтам фітнес-клубу, що сформовані у групи за різновидами діяльності відображені в таблиці 1.2.

Для організація діяльності роботи досліджуваного об'єкту необхідно знати не лише його мету та аспекти діяльності, а й організаційну структуру.

Для успішної роботи спортивного клубу в його структурі крім основного персоналу (тренери фітнес-програм і менеджери – особи, які надають первинні консультації клієнтам та виконують управлінські функції, щодо організації діяльності), необхідні працівники, що займаються фінансовими справами клубу (фінансові працівники), просуванням на ринок фітнес-послуг (маркетологи), дієтологи, косметологи, а також обслуговуючий персонал (прибиральниці, охорона порядку, працівники ІТ сфери). Крім зазначених працівників,

спортивний клуб очолює генеральний директор, обов'язками якого є фінансова і правова відповідальність перед державними органами та загальний нагляд за діяльністю клубу, його працівниками.

Таблиця 1.2

Послуги, що надаються клієнтам спортивного клубу

Послуга	Характеристика
1. Фітнес – тест (послуга надається за бажанням клієнта):	<ul style="list-style-type: none"> - встановлення рівня фізичної працездатності клієнта; - допомога у визначенні оптимальних навантажень та фітнес – програм; - поради щодо питного режиму та режиму харчування; - формування безпечної та ефективної індивідуальної програми занять
1. Фітнес – програми:	<ul style="list-style-type: none"> - аеробні (трекінг, спортивні танці, аеробіка, його, пілатес); - силові програми (тренування з використанням навантажень); - водні програми; - бойові мистецтва (айкідо, капоейра, бокс); - ігрові програми (настільний теніс, міні – футбол, баскетбол, волейбол, бадмінтон, теніс)
2. Додаткові послуги:	<ul style="list-style-type: none"> - солярій; - косметичні послуги; - лазня, сауна; - фітнес – бар; - масаж; - гідромасажні ванни - трав'яні, заспокоюючі ванни

Організаційна схема діяльності спортивного клубу зображена на Рис. 1.2.



Рисунок 1.2. Структура організації спортивного клубу

Отже, із вище переліченого сегменту послуг можна сформуванати наступні аспекти, щодо організації діяльності СК (рисунок 1.3).

Процес організації діяльності спортивного клубу:
- «...організація високотехнологічного цифрового тренажерного комплексу, що дозволить імітувати фізіологію рухової системи людини»;
- «...організація персональних тренувань, розробка індивідуальних програм, що враховують потреби клієнтів, розробка раціону харчування дієтологом»;
- «...організація залу групових занять з широким спектром секцій та комплекс оздоровчих послуг, які подарують, хороший настрій, оптимізм та відчуття молодості»

Рисунок 1.3. Організація діяльності спортивного клубу

Незважаючи на прагнення адміністрації спортивного клубу організувати дозвілля найкращим чином, все ж існують проблеми, на які необхідно звернути увагу, зокрема: (рис. 1.4).

Проблеми:
- «...відсутні засоби, що забезпечують організацію функціонування діяльності нового SPA – центру»;
- «...відсутність засобів, які надаватимуть клієнту фітнес – клубу організувати он-лайн свій графік діяльності в клубі»;
- «...відсутність засобів для обміну інформацією між тренером, консультантом, які є закріплені за даним користувачем послуг у реальному часі»;
- «...відсутність засобів для формування автоматизованого звіту»

Рисунок 1.4. На основі проведеного дослідження виділено проблеми

Здійснивши аналіз підходів щодо організації спортивного клубу (СК), з метою покращення надання якісних послуг останнього, можна виділити наступні напрямки роботи реалізація, яких забезпечить процвітання спортивного клубу:

- програмне забезпечення СК (реалізується у вигляді веб-сайту);
- послуги, які надаються установою обмежуються організацією колективних та індивідуальних занять для формування фізичної культури клієнта;
- організація консультацій у реальному часі;
- організація можливості скоригувати розклад самим клієнтом СК.

1.3. Обґрунтування вибору проектних рішень щодо діяльності спортивного клубу

На сьогоднішній день традиційні додатки стрімко витісняються веб-орієнтованим програмним забезпеченням. Тепер сайт з авторизацією для зареєстрованих користувачів – корпоративний інтернет – засіб для ефективної спільної роботи. «...Відмінність браузерного додатка від звичайного веб-сайту в тому, що робоча сесія завжди починається з авторизації користувача.

Розвиток подібних систем дав зрослий рівень проникнення Інтернету і фінансова криза, що змусила компанії шукати можливості по скороченню витрат» [49].

Інтенсифікація впровадження онлайн-систем пояснюється низкою об'єктивних причин. Перший аргумент – несумісність традиційних програм, написаних для різних платформ: Windows, Linux або Mac OS. Кожна операційна система вимагає свого підходу до розробки і наявності в штаті компанії профільного спеціаліста. Web -додаткам достатньо наявності браузера в операційній системі. Цей підхід гнучкий в сенсі використання мобільних пристроїв, яким серверна частина віддає полегшену версію.

Браузерні додатки не вимагають інсталяції, а, отже, і ліцензії. Робоче місце може бути розгорнуто на будь-якому комп'ютері або мобільному пристрої, де є Інтернет (LAN, Wi-Fi, 3G, EDGE, GPRS), в гранично малий термін. Це ідеальне рішення для компанії з декількома офісами, яке не потребує спеціалізованого інфраструктурного рішення у вигляді віртуальної приватної мережі (VPN), а також для підключення мобільних терміналів, встановленої на техніці (навантажувачі). Не завжди правомірне і відповідне процесуальним процедурам вилучення комп'ютерної техніки не вплине на роботу компанії - всі дані залишаться на сервері, а доступ до системи для віддаленого комп'ютера буде заблоковано.

Реалізація веб-орієнтованого програмного забезпечення реалізується на основі web-сервісів проектування, із застосуванням таких підходів:

- ASP.NET;
- PHP.
- JavaScript фреймворків.

На нашу думку, з вище перелічених підходів, найкращим вибором буде технологія ASP.NET. Особливістю даного підходу є використання строго типізованих мов програмування та шаблону проектування MVC. Патерну MVC передбачає поділ проекту на три взаємодіючі компоненти, що реалізують різні напрямки конструювання програмного забезпечення, а саме: контролер

(реалізує логіку роботи програми), модель (реалізує логіку використання даних) і уявлення (реалізує користувацький інтерфейс програми).

Крім вище визначеної структури MVC у додатку використовуються проекти «бібліотеки класів», що використовуються для формування бізнес логіки проекту (BusinessLogic) та об'єктів конструйованого середовища (Domain).

Платформа ASP.NET MVC підтримує концепцію «движка представлень» - це замінні модулі, які реалізують вибір різного синтаксису шаблону. Для розробки було використано новий тип движка «Razor», який оптимізований під генерацію HTML-коду, фокусуючись на коді шаблону та дозволяє використовувати менше коду при поєднанні елементів мови С# та HTML розмітки.

Особливістю даного движка є:

- *Компактний, Виразний і Гнучкий:* навантаження на клавіатуру зменшує Razor. Цей процес необхідний для роботи у файлі. Крім того, останій надає доволі гнучкий та швидкий процес програмування. Ще однією відмінністю (від більшості синтаксису шаблонів) є те, що немає потреби переривати написання коду, щоб суттєво визначити серверні блоки коду в HTML. Парсер досить розумний, для того щоб зробити це за вас, ґрунтуючись на коді.
- *Легкість освоєння:* Razor – доволі простий в опануванні, надає можливість за відносно короткий термін стати найбільш продуктивним при мінімальному застосуванні зусиль, зокрема, є можливість використати всі існуючі навички мови програмування та HTML.
- *Не нова мова:* розробники можуть використовувати існуючі навички роботи з С#/VB (може бути інша мова) та надають синтаксис розмітки шаблону, що дає приголомшливий процес побудови HTML.
- *Працює в будь-якому текстовому редакторі:* Razor дає можливість бути продуктивним у будь-якому текстовому редакторі, не вимагає певного інструменту (блокнот не є винятком).

- *Підтримка Intellisense*: Razor був розроблений з врахуванням не бути прив'язаним до певного інструменту (редактору коду), має відмінну підтримку в Visual Studio. Visual Studi 2010 і Visual Web Developer 2010 (отримали оновлення для intellisense).

- *Юніт-тестування*: Нова реалізація движка підтримує можливість створювати юніт-тести над уявленнями (не вимагає web- сервера або контролера. Крім того, не має потреби в спеціальних доменах додатків, можна розміщувати в будь – якому проєкті юніт – тестування).

Проектування системи відбувається із поєднанням об'єктно-орієнтованого підходу до програмування з процедурно-орієнтованим. Всі класи документувалися інформаційно та систематично.

1.4. Рішення з інформаційного забезпечення діяльності спортивного клубу

Розроблене програмне забезпечення не використовує первісні сигнали (аналогові, дискретні сигнали і т. д.), як джерела формування даних, оскільки об'єкт автоматизації є веб-орієнтованим додатком для збереження, опрацювання даних у базі даних спортивного клубу.

Первісні дані, які отримує система для опрацювання надається користувачем та отримується у вигляді набору текстової і числової інформації, введеної працівником чи клієнтом. Вхідна інформація, яка формується користувачем в рамках функціоналу предметної області реалізується через об'єктні моделі, де властивості, що описує спроектовану сутність представляється у системі наступними типами даних:

- Стрічковий тип даних;
- Числовий тип даних;
- Тип даних дата і час.

Всі первісні дані, які користувачі формують у системі для збереження у базі даних реалізуються у записи відповідних таблиць із однойменними назвами класів моделей. Первісні дані, що вводяться у систему збираються на основі поданих клієнтом документів. Даними документами є:

- «Паспорт громадянина»;
- «Реєстраційна облікова картка платника податків»;
- «Медична довідка для осіб, що відвідують групові заняття».

Із паспорту громадянина у систему вносяться наступні дані:

- «Прізвище, ім'я, по батькові»;
- «Дата народження»;
- «Місце проживання»;
- «Серія та номер паспорту»;
- «Дата видачі паспорту та ким було видано»;

Із реєстраційної облікової картки платника податків заносять у систему ідентифікаційний номер платника податків.

На основі медичної довідки вносять дані про:

- «Дату проходження флюорографії»;
- «Медичні показання клієнта».

Дана інформація вноситься у систему менеджером для роботи із клієнтами.

Вихідна інформація формується на основі моделей даних та виводиться користувачу у вигляді сформованої html-сторінки (переважно у вигляді таблиці). Формування даних відбувається за запитом користувача та передбачає засоби фільтрування інформації. Процес фільтрування вихідної інформації залежить від ролі користувача системи. Вся вихідна інформація підлягає друку. Основними документами для друку є:

- «Ордер оплати послуг»;
- «Список клієнтів у розрізі групових занять»;
- «Загальний список клієнтів спортивного клубу»;

- «Список працівників СК (загальний список та по відділах)»;
- «Звіт щодо оплати послуг клієнтом за певний період».

Всі звіти відображаються у вигляді веб-сторінок та можуть бути сформовані у вигляді pdf файлів для друку.

Для одержання семантичних (концептуальних) моделей, що відображають проєктовану предметну область, інформаційні потреби користувачів, поняття про факти, події, предмети, якими буде оперувати зазначена інформаційна система було здійснено інфологічне проєктування, що використовується, в якості інструменту для побудови семантичних моделей даних, уніфікованого представлення даних, незалежного від реалізуючого його програмного забезпечення.

Модель «сутність – зв'язок» (entity-relationship model, ER-model) застосовується на етапі інфологічного проєктування, яка базується на опорній семантичній інформації щодо реального світу. Зазначена модель призначена для логічного подання даних у контексті взаємозв'язку з іншими даними. Модель вважають загальною, отже, з неї можуть бути породжені різні моделі, зокрема: мережна, ієрархічна, об'єктна, реляційна.

Модель «сутність-зв'язок» не визначає операцій над даними й обмежується описом тільки їхньої логічної структури. Інфологічна модель відображена у додатку (ДОДАТОК Б).

На етапі даталогічного проєктування здійснюється перехід від інфологічної моделі ПО до логічної (даталогічної) моделі, яка підтримується засобами конкретної СУБД. Процес переходу від інфологічної до даталогічної моделі має назву відображення.

БД, структурована на логічному рівні та орієнтована на конкретну СУБД має назву даталогічна модель. Проєктуючи дану модель необхідно врахувати, не існує формалізованих методів, які б давали можливість однозначно виконати даталогічне проєктування. Відповідно, результат проєктних робіт цілком залежить від кваліфікації, досвіду, уміння фахівців.

Можна отримати кілька варіантів побудови логічної моделі даних здійснюючи даталогічне проектування. Особливим етапом є оцінка отриманих моделей та вибір оптимального варіанта. Результату, який отримали, необхідно дати оцінку з точки зору відповідності наявним машинним ресурсам. При встановленні невідповідності обмеженням варто здійснити перепроєктування бази даних.. А на моделі, яку отримали, перевірити умови виконання запитів користувачів, вимоги прикладних програм (встановити умову адекватності логічної моделі інформаційній моделі предметної області). Після проведених дій нами отримано даталогічну модель даних, що представлена у вигляді ERD (в нотаціях IDEF1X) (ДОДАТОК В).

На основі даталогічної моделі даних розроблено SQL код для створення таблиць БД та зв'язків між ними.

Код таблиці, що містить дані про користувачів системи.

```
CREATE TABLE [dbo].[Users] (
    [Id] INT IDENTITY (1, 1) NOT NULL,
    [Login] NVARCHAR (10) NOT NULL,
    [Password] NVARCHAR (10) NOT NULL,
    [image] NVARCHAR (MAX) NOT NULL,
    [Surname] NVARCHAR (MAX) NOT NULL,
    [Name] NVARCHAR (MAX) NOT NULL,
    [MiddleName] NVARCHAR (MAX) NOT NULL,
    [DateBirth] DATETIME NOT NULL,
    [HomeAddress] NVARCHAR (MAX) NOT NULL,
    [PassportNumber] NVARCHAR (MAX) NOT NULL,
    [PassportAddress] NVARCHAR (MAX) NOT NULL,
    [PassportDate] DATETIME NOT NULL,
    [PassportSeries] NVARCHAR (MAX) NOT NULL,

    [IdRole] INT NULL,
    [Email] NVARCHAR (MAX) NOT NULL,
    [PhoneNumber] NVARCHAR (13) NOT NULL,
    [CreateDate] DATETIME NOT NULL,
    [IdDepartment] INT NULL,
    [Discriminator] NVARCHAR (128) NOT NULL,
    [Department_Id] INT NULL,
    [Role_Id] INT NULL,
    CONSTRAINT [PK_dbo.Users] PRIMARY KEY CLUSTERED ([Id] ASC),
    CONSTRAINT [FK_dbo.Users_dbo.Departments_Departament_Id] FOREIGN KEY ([Department_Id])
REFERENCES [dbo].[Departments] ([Id]),
    CONSTRAINT [FK_dbo.Users_dbo.Roles_Role_Id] FOREIGN KEY ([Role_Id]) REFERENCES
[dbo].[Roles] ([Id])
);

GO
CREATE NONCLUSTERED INDEX [IX_Departament_Id]
ON [dbo].[Users]([Department_Id] ASC);
```

Код таблиці, що містить дані про послуги, які надає клієнтам спортивний клуб.

```
CREATE TABLE [dbo].[Services] (
    [Id] INT IDENTITY (1, 1) NOT NULL,
    [Name] NVARCHAR (15) NOT NULL,
    [Description] NVARCHAR (150) NOT NULL,
    [Price] REAL NOT NULL,
    [IDBonusProgram] INT NOT NULL,
    [BonusProgram_Id] INT NULL,
    [Client_Id] INT NULL,
    CONSTRAINT [PK_dbo.Services] PRIMARY KEY CLUSTERED ([Id] ASC),
    CONSTRAINT [FK_dbo.Services_dbo. BonusPrograms_BonusProgram_Id] FOREIGN KEY
([BonusProgram_Id]) REFERENCES [dbo].[BonusPrograms] ([Id]),
    CONSTRAINT [FK_dbo.Services_dbo.Users_Client_Id] FOREIGN KEY ([Client_Id]) REFERENCES
[dbo].[Users] ([Id])
);

GO
CREATE NONCLUSTERED INDEX [IX_BonusProgram_Id]
ON [dbo].[Services]([BonusProgram_Id] ASC);
```

Код таблиці, що містить дані про ролі користувачів системи на основі яких визначаються дія користувача у системі.

```
CREATE TABLE [dbo].[Roles] (
    [Id] INT IDENTITY (1, 1) NOT NULL,
    [Name] NVARCHAR (MAX) NOT NULL,
    CONSTRAINT [PK_dbo.Roles] PRIMARY KEY CLUSTERED ([Id] ASC)
);
```

Код таблиці, що містить дані про оплату внесену користувачем системи.

```
CREATE TABLE [dbo].[Payments] (
    [Id] INT IDENTITY (1, 1) NOT NULL,
    [IDUser] INT NOT NULL,
    [IDService] INT NOT NULL,
    [AccruedFees] REAL NOT NULL,
    [CreateDate] DATETIME NOT NULL,
    [Client_Id] INT NULL,
    CONSTRAINT [PK_dbo.Payments] PRIMARY KEY CLUSTERED ([Id] ASC),
    CONSTRAINT [FK_dbo.Payments_dbo.Users_Client_Id] FOREIGN KEY ([Client_Id]) REFERENCES
[dbo].[Users] ([Id])
);

GO
CREATE NONCLUSTERED INDEX [IX_Client_Id]
ON [dbo].[Payments]([Client_Id] ASC);
```

Код таблиці, що містить дані створені новини маркетологом спортивного клубу.

```
CREATE TABLE [dbo].[News] (
    [Id] INT IDENTITY (1, 1) NOT NULL,
    [IdUser] INT NOT NULL,
    [Title] NVARCHAR (15) NOT NULL,
    [ShortDescription] NVARCHAR (250) NOT NULL,
    [Text] NVARCHAR (MAX) NOT NULL,
```

```

[CreateDate]          DATETIME          NOT NULL,
CONSTRAINT [PK_dbo.News] PRIMARY KEY CLUSTERED ([Id] ASC)
);

```

Код таблиці, що містить дані про створені повідомлення, які створюють користувачі системи.

```

CREATE TABLE [dbo].[Messages] (
    [Id]                INT                IDENTITY (1, 1) NOT NULL,
    [IDUserIncomingMessage] INT          NOT NULL,
    [IDUserOutgoinMessage] INT          NOT NULL,
    [Title]             NVARCHAR (15)    NOT NULL,
    [ShortDescription]  NVARCHAR (250)   NOT NULL,
    [Text]              NVARCHAR (MAX)   NOT NULL,
    [GreatedDate]      DATETIME         NOT NULL,
    CONSTRAINT [PK_dbo. Messages] PRIMARY KEY CLUSTERED ([Id] ASC)
);

```

Код таблиці, що містить дані про історію попередніх робіт своїх працівників.

```

CREATE TABLE [dbo].[HistoryWorks] (
    [Id]                INT                IDENTITY (1, 1) NOT NULL,
    [IdUser]            INT                NULL,
    [Post]              NVARCHAR (MAX)    NOT NULL,
    [Institutions]     NVARCHAR (MAX)    NOT NULL,
    [DateReception]    DATETIME         NOT NULL,
    [DateDismissal]    DATETIME         NOT NULL,
    [AdditionalInfo]   NVARCHAR (MAX)    NOT NULL,
    [User_Id]          INT                NULL,
    [Employee_Id]     INT                NULL,
    CONSTRAINT [PK_dbo.HistoryWorks] PRIMARY KEY CLUSTERED ([Id] ASC),
    CONSTRAINT [FK_dbo.HistoryWorks_dbo.Users_User_Id] FOREIGN KEY ([User_Id]) REFERENCES
[dbo].[Users] ([Id]),
    CONSTRAINT [FK_dbo.HistoryWorks_dbo.Users_Employee_Id] FOREIGN KEY ([Employee_Id])
REFERENCES [dbo].[Users] ([Id])
);

```

```

GO
CREATE NONCLUSTERED INDEX [IX_Employee_Id]
    ON [dbo].[HistoryWorks]([Employee_Id] ASC);

```

Код таблиці, що містить дані про історію навчання працівників, їх освітні здобутки.

```

CREATE TABLE [dbo].[HistoryEducations] (
    [Id]                INT                IDENTITY (1, 1) NOT NULL,
    [IdUser]            INT                NULL,
    [Specialty]         NVARCHAR (MAX)    NOT NULL,
    [Institutions]     NVARCHAR (MAX)    NOT NULL,
    [DateStart]        DATETIME         NOT NULL,
    [DateCompletion]   DATETIME         NOT NULL,
    [AdditionalInfo]   NVARCHAR (MAX)    NOT NULL,
    [User_Id]          INT                NULL,
    [Employee_Id]     INT                NULL,
    CONSTRAINT [PK_dbo.HistoryEducations] PRIMARY KEY CLUSTERED ([Id] ASC),
    CONSTRAINT [FK_dbo.HistoryEducations_dbo.Users_User_Id] FOREIGN KEY ([User_Id])
REFERENCES [dbo].[Users] ([Id]),

```

```

        CONSTRAINT [FK_dbo.HistoryEducations_dbo.Users_Employee_Id] FOREIGN KEY ([Employee_Id])
REFERENCES [dbo].[Users] ([Id])
);

```

```

GO
CREATE NONCLUSTERED INDEX [IX_User_Id]
ON [dbo].[HistoryEducations] ([User_Id] ASC);
);

```

```

GO
CREATE NONCLUSTERED INDEX [IX_Employee_Id]
ON [dbo].[HistoryEducations]([Employee_Id] ASC);

```

Код таблиці, що містить дані створені новини маркетологом спортивного клубу.

```

CREATE TABLE [dbo].[Equipments] (
    [Id] INT IDENTITY (1, 1) NOT NULL,
    [IdActive] INT NULL,
    [Name] NVARCHAR (MAX) NOT NULL,
    [Count] INT NOT NULL,
    [CreateDate] DATETIME NOT NULL,
    [Activ_Id] INT NULL,
    CONSTRAINT [PK_dbo.Equipments] PRIMARY KEY CLUSTERED ([Id] ASC),
    CONSTRAINT [FK_dbo.Equipments_dbo.Activs_Activ_Id] FOREIGN KEY ([Activ_Id]) REFERENCES
[dbo].[Activs] ([Id])
);

```

```

GO
CREATE NONCLUSTERED INDEX [IX_Activ_Id]
ON [dbo].[Equipments]([Activ_Id] ASC);

```

```

CREATE TABLE [dbo].[Departments] (
    [Id] INT IDENTITY (1, 1) NOT NULL,
    [Name] NVARCHAR (20) NOT NULL,
    CONSTRAINT [PK_dbo.Departments] PRIMARY KEY CLUSTERED ([Id] ASC)
);

```

Код таблиці, що містить дані про відділи спортивного клубу, тобто його організаційну структуру.

```

CREATE TABLE [dbo].[Comments] (
    [Id] INT IDENTITY (1, 1) NOT NULL,
    [IdNew] INT NOT NULL,
    [IdUser] INT NOT NULL,
    [Text] NVARCHAR (MAX) NOT NULL,
    [CreateDate] DATETIME NOT NULL,
    CONSTRAINT [PK_dbo.Comments] PRIMARY KEY CLUSTERED ([Id] ASC)
);

```

Код таблиці, що містить дані про акційні програми спортивного клубу.

```

CREATE TABLE [dbo].[BonusPrograms] (
    [Id] INT IDENTITY (1, 1) NOT NULL,
    [Name] NVARCHAR (15) NOT NULL,
    [Description] NVARCHAR (MAX) NOT NULL,
    [PercentDiscount] INT NOT NULL,

```

```

[CreateDate]      DATETIME      NOT NULL,
[Client_Id]      INT              NULL,
CONSTRAINT [PK_dbo.BonusPrograms] PRIMARY KEY CLUSTERED ([Id] ASC),
CONSTRAINT [FK_dbo.BonusPrograms_dbo.Users_Client_Id] FOREIGN KEY ([Client_Id])
REFERENCES [dbo].[Users] ([Id])
);

```

```

GO
CREATE NONCLUSTERED INDEX [IX_Client_Id]
ON [dbo].[BonusPrograms]([Client_Id] ASC);

```

Код таблиці, що містить дані про кабінети спортивного клубу.

```

CREATE TABLE [dbo].[Activs] (
[Id]              INT              IDENTITY (1, 1) NOT NULL,
[Name]            NVARCHAR (MAX)  NOT NULL,
[Number]          NVARCHAR (MAX)  NOT NULL,
[IdDepartment]   INT              NOT NULL,
[Department_Id]  INT              NULL,
CONSTRAINT [PK_dbo.Activs] PRIMARY KEY CLUSTERED ([Id] ASC),
CONSTRAINT [FK_dbo.Activs_dbo.Departments_Department_Id] FOREIGN KEY ([Department_Id])
REFERENCES [dbo].[Departments] ([Id])
);

```

```

GO
CREATE NONCLUSTERED INDEX [IX_Department_Id]
ON [dbo].[Activs] ([Department_Id] ASC);

```

При побудові таблиць бази даних (БД) було використано індексовані первинні ключі, які забезпечують унікальність кожного запису БД. Взаємозв'язок між таблицями здійснюється за рахунок зовнішнього ключа (поле, яке містить первинний ключ рядка даних іншої таблиці). Даним способом реалізується зв'язок записів виду один-до-одного та один-до-багатьох.

Зв'язки виду багато-до-багатьох реалізуються створенням сутності-зв'язку, що зберігає записи ідентифікатори полів обох таблиць, що беруть участь у відношенні.

Розроблений програмний додаток створюється на основі технології ASP.NET, яка призначена для реалізації веб-додатків. Перш ніж перейти до процесу конструювання html-сторінок, слід розробити прототип дизайну сайту.

Розробка сайту починається із створення майстер-сторінки, що забезпечує відображення стандартних елементів однаково для всіх сторінок. Шаблон складатиметься із трьох елементів і міститиме:

- «Секція для формування заголовку сторінки, де є назва сайту та емблема»;
- «Секція із рядом спадного меню»;
- «Секція із авторством розробки сайту та довідковою інформацією».

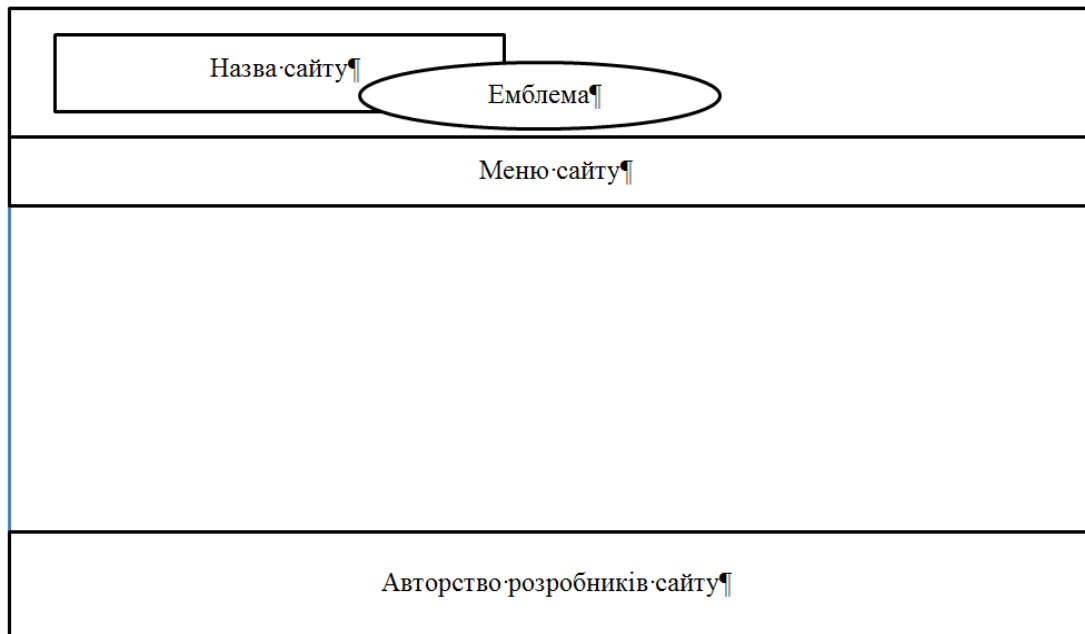


Рисунок 1.5. Майстер-сторінка сайту

Основний зміст сайту (наповненість) розміщується у центральній частині сторінки та складається із трьох частин, що відображаються залежно від сторінки, яку викликає користувач даної системи.

Основний блок може містити наступні елементи:

- «Секція для відображення загальної інформації за обраним розділом»;
- «Секція основного контенту сторінки»;
- «Секція відображення новин».

Центральний вигляд шаблону може змінюватися залежно від вибраного розділу користувачем системи.

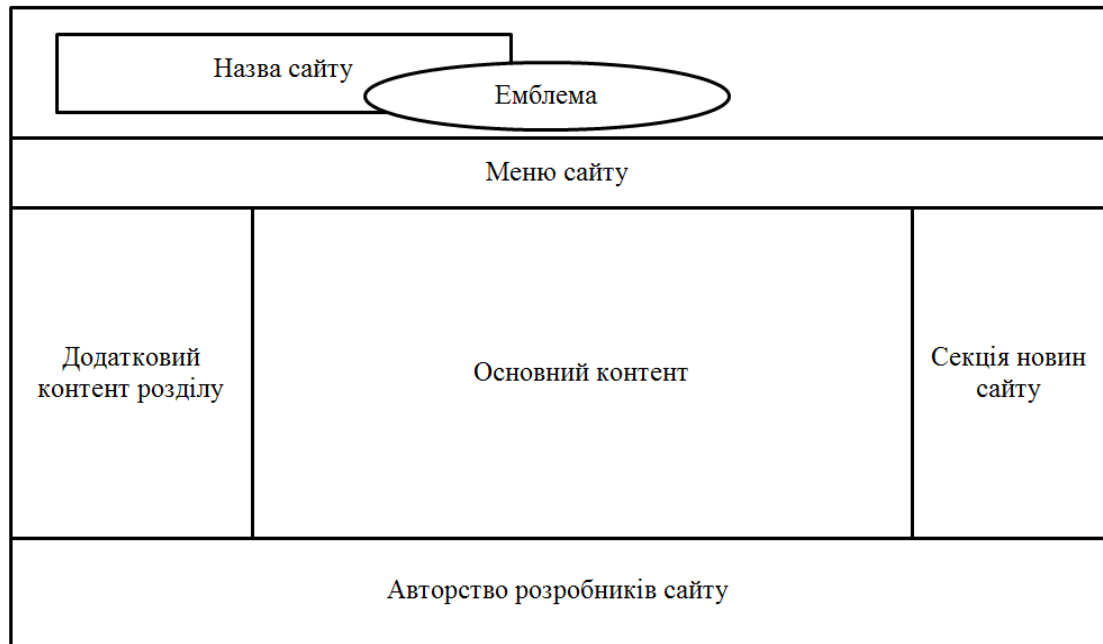
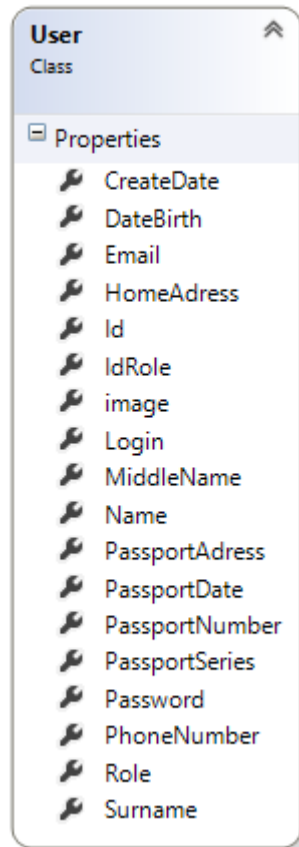


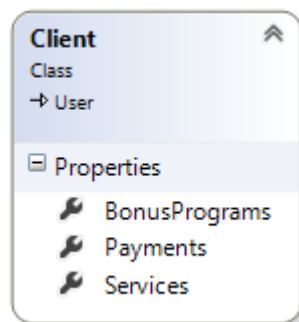
Рисунок 1.6. Прототип головного вікна програми

При реалізація web -додатку було вибрано підхід Code first, який передбачає розробку моделей даних перед створенням таблиць БД, що відбувається автоматично засобами на основі яких реалізована технологія Entity Framework. При розробці автоматизованої системи СК використано 15 об'єкт предметної галузі. Враховуючи дану сукупність сутностей було розроблено бізнес-логіку програмної системи. Реалізована бізнес-логіка через розподіл функціоналу на моделі (опис властивостей, якими володіє сутність) та інтерфейси (методи, як реалізують процес обробки властивостей). Сутності, що були використанні при розробці програмної системи є наступні:

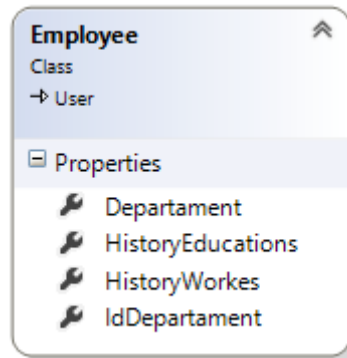
Користувач (*User*) – клас у якому описується загальна інформація про людину, як суб'єкта суспільства, а також інформація за рахунок, якої дана особа ідентифікується у системі. Даний клас являється батьківським класом для класів Клієнт (*Client*) і Працівник (*Employee*).



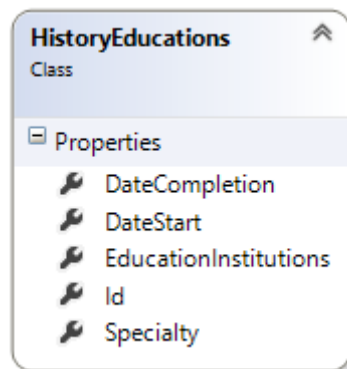
Клієнти (*Client*) – клас призначений для зберігання інформації про послуги, які надаються клієнту та акційні програми, які отримує клієнт за рахунок їх покупки або за роки співпраці із СК.



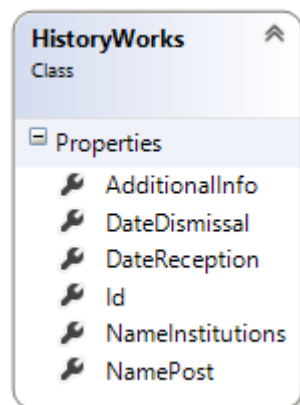
Працівник (*Employee*) – клас призначений для зберігання інформації про освіту працівника СК його історію попередніх місць роботи, відділ у якому він працює.



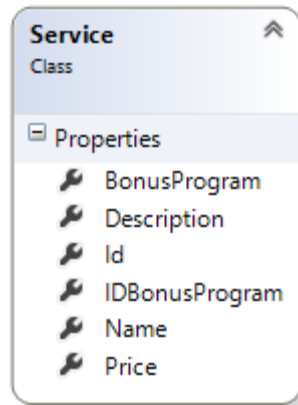
«Історія освіти» (*HistoryEducations*) – клас призначений для зберігання інформації про спеціальності та навчальні заклади у яких навчався працівник СК.



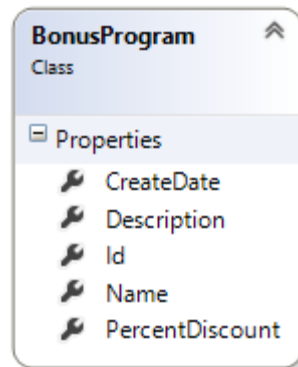
«Історія місця робіт» (*HistoryWorks*) – клас призначений для зберігання інформації про всі попередні місця роботи працівника СК.



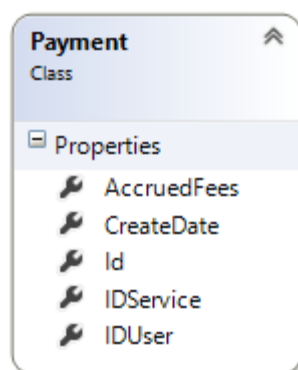
Послуги (*Services*) – клас призначений для зберігання інформації про послуги, які надає СК для клієнтів закладу.



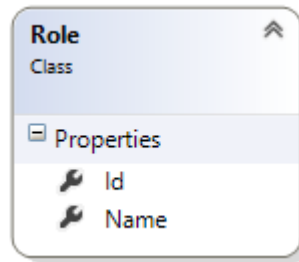
Акційні програми (*BonusProgram*) – клас призначений для зберігання інформації про акційні програми СК, які дозволяють надавати знижки постійним клієнтам та заохочувати їх для подальшої співпраці.



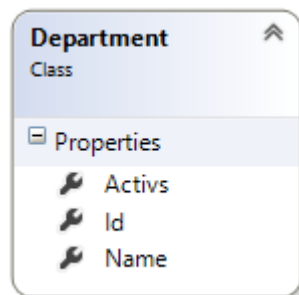
Оплата (*Payment*) – клас призначений для зберігання інформації про оплату проведену користувачем за отриманні послуги від СК.



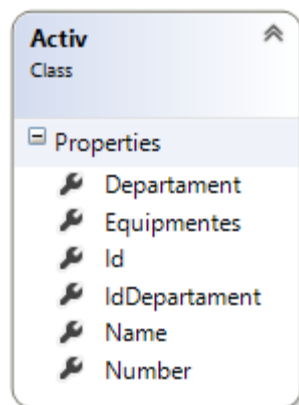
Роль (*Role*) – клас призначений для зберігання інформації про ролі користувачів системи, які визначають доступ до методів з допомогою яких відбувається опрацювання введеної інформації.



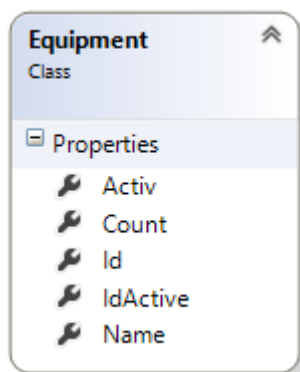
Відділ (*Department*) – клас призначений для зберігання інформації про структурну організацію СК.



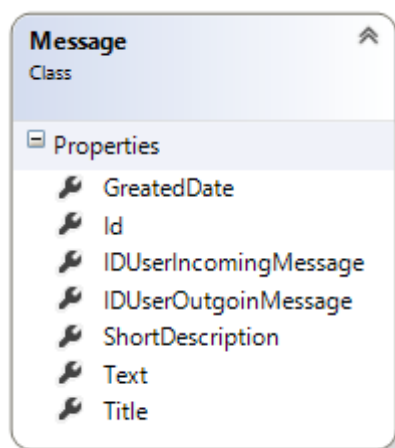
Актив (кабінет) (*Activ*) – клас призначений для зберігання інформації про кабінети, які відносяться до визначеного адміністратором відділу відносно розробленої адміністрацією СК структурну організацію.



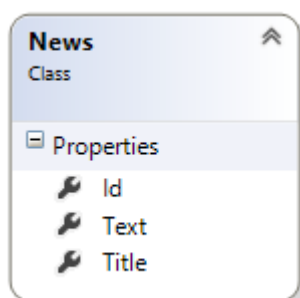
Інвентар (*Equipment*) – клас призначений для зберігання інформації про інвентар, який знаходиться на балансі СК та закріплений за певним кабінетом.



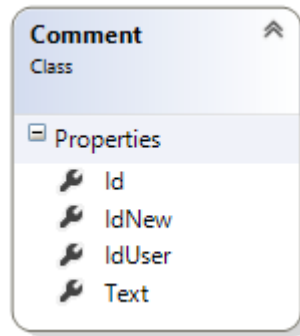
Повідомлення (*Message*) – клас призначений для зберігання інформації про повідомлення, які були надіслані користувачеві та отримані ним від інших користувачів системи.



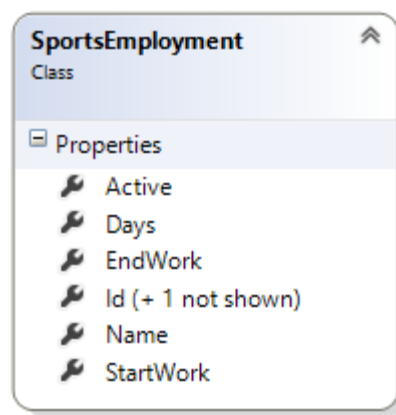
Новини (*News*) – клас призначений для зберігання інформації про новини і статті, які формують на сайті менеджери СК.



Коментарі (*Comment*) – клас призначений для зберігання інформації про коментарі, які залишають клієнти відносно нової інформації та статей, які створенні менеджерами СК.



«Спортивні заняття» (*SportsEmployment*) – клас призначений для зберігання інформації про спортивні заняття, які проводяться в СК.



Розроблена система управління діяльністю СК надає можливість виводити інформації різного характеру (інформації про клієнта, тренера, послуги тощо), а також діалогові повідомлення для реалізації зворотного зв'язку із клієнтом (повідомлення про успішність дії чи повідомлення із помилками допущеними клієнтом). Розглянемо більш інформаційні звіти, які формує система:

Інформація про клієнтів містить наступні дані:

- «Прізвище, ім'я, по батькові клієнта»;
- «Адреса проживання»;
- «Список послуг, які відвідує клієнт в даний день».

Інформація про працівника містить наступні дані

- «Прізвище, ім'я, по батькові клієнта»;
- «Відділ роботи»;

- «Список занять, які проводить працівника протягом дня».

Звіт про сплату за надані послуги клієнту містить наступну інформацію:

- «Прізвище, ім'я, по батькові клієнта»;
- «Список послуг із наступною структурою»:
 - Назва послуги;
 - Дата надання послуги клієнту;
 - Вартість послуги;
 - Сплачена сума клієнтом;
 - Дата сплати.

Всі інші звіти містять інформацію про відділи, кабінети, групові заняття, акційні програми тощо.

Інформація, яка підлягає для збереження у БД формується із множини даних введених користувачем у форми розроблені у вигляді html-сторінок. Форми web -сторінки реалізується на основі моделі даних та передбачають відображення всіх властивостей об'єкта у вигляді окремих елементів.

Форма для реєстрації даних користувача системи

Дані про користувача

Логін

Пароль

Прізвище

Ім'я

По батькові

Файл не вибран

Дата народження

Домашня адреса

Серія паспорта

Номер паспорта

Рисунок 1.7. Форма реєстрації даних про клієнта спортивного клубу

Процес формування даних (перевірка коректності введених даних) здійснюється на стороні клієнта та передбачає перевірку:

- введення інформації у поле форми;
- відповідність введеної інформації наперед заданому шаблону (типу даних та параметрам заданим розробниками);

Крім реалізації перевірки даних сформованих користувачем відбувається перевірка на стороні сервера, тобто відповідність моделі існуючій таблиці даних у БД.

Висновки за розділом 1

Застосування та розвиток інформаційних продуктів є новим основним засобом організації послуг клієнтам спортивного клубу, а тому потребують узагальнень та розробки практичних рекомендацій на всіх рівнях.

Розробка web – додатків, на сьогодні, є досить актуальним, оскільки зазначена технологія розвивається швидкими темпами, має суттєві переваги щодо економії часу, ресурсів, не обмежує користувача у виборі місцезнаходження для застосування своїх можливостей, можна працювати з будь – яких девайсів не залежно від операційної системи чи її технічних характеристик.

Реалізація веб-орієнтованого програмного забезпечення реалізується на основі web-сервісів проектування, яких здійснюється на основі таких підходів:

- ASP.NET;
- PHP.
- JavaScript фреймворків.

Проектування системи відбувається із поєднанням об'єктно-орієнтованого підходу до програмування з процедурно-орієнтованим. Всі класи документувалися інформаційно та систематично.

РОЗДІЛ 2

ПРОЕКТУВАННЯ СИСТЕМИ УПРАВЛІННЯ ПРОЦЕСОМ НАДАННЯ ПОСЛУГ КЛІЄНТАМ СПОРТИВНОГО КЛУБУ

2.1. Рішення з технічного забезпечення

Мережа спортивного клубу «SPA-Фітнес» налічує до сорока робочих станцій, три комутатори, крім того використовуються різноманітні загальнопоширені і спеціалізовані апаратні, програмні засоби, а також три спеціалізовані сервери:

- 1) С-1 - відповідає за організацію доменної мережі, за допомогою якого організовується доступ та реєстрації користувачів при вході в домен.
- 2) С-2 - сервер баз даних білінгових та інших програм;
- 3) С-3 - сервер, що керує доступом всіх робочих станцій до загальних серверів фітнес-клубу та до мережі Інтернет. Загальна схема мережі об'єкта практики зображена на рис. 2.1. (ДОДАТОК Г).

Мережа СК «SPA-Фітнес» включає в себе три комутатори Allied Telesyn AT-FS716. Один із них знаходиться в серверній, і є основним вузлом в мережі, а інші знаходяться на поверхах.

Allied Telesyn AT-FS716 має такі характеристики:

- «16 портів Fast Ethernet з функцією Auto-MDI/MDI-X»;
- «можливість передачі в режимі дуплекса (напівдуплекса) на кожному порті»;
- «комутація методом зберігання та переадресації»;
- «інтегрована підсистема перегляду адрес з таблиці абсолютних MAC-адрес на 8192 записи»;
- «можливість буферизації даних об'ємом до 1536 Кб»;
- «контроль потоку IEEE 802.3x для дуплексного режиму»;
- «підтримка стандартів 802.3u, 802.3x

Отже, можна стверджувати, що мережа СК є добре розробленою та забезпечує функціонування всіх відділів фітнес-клубу.

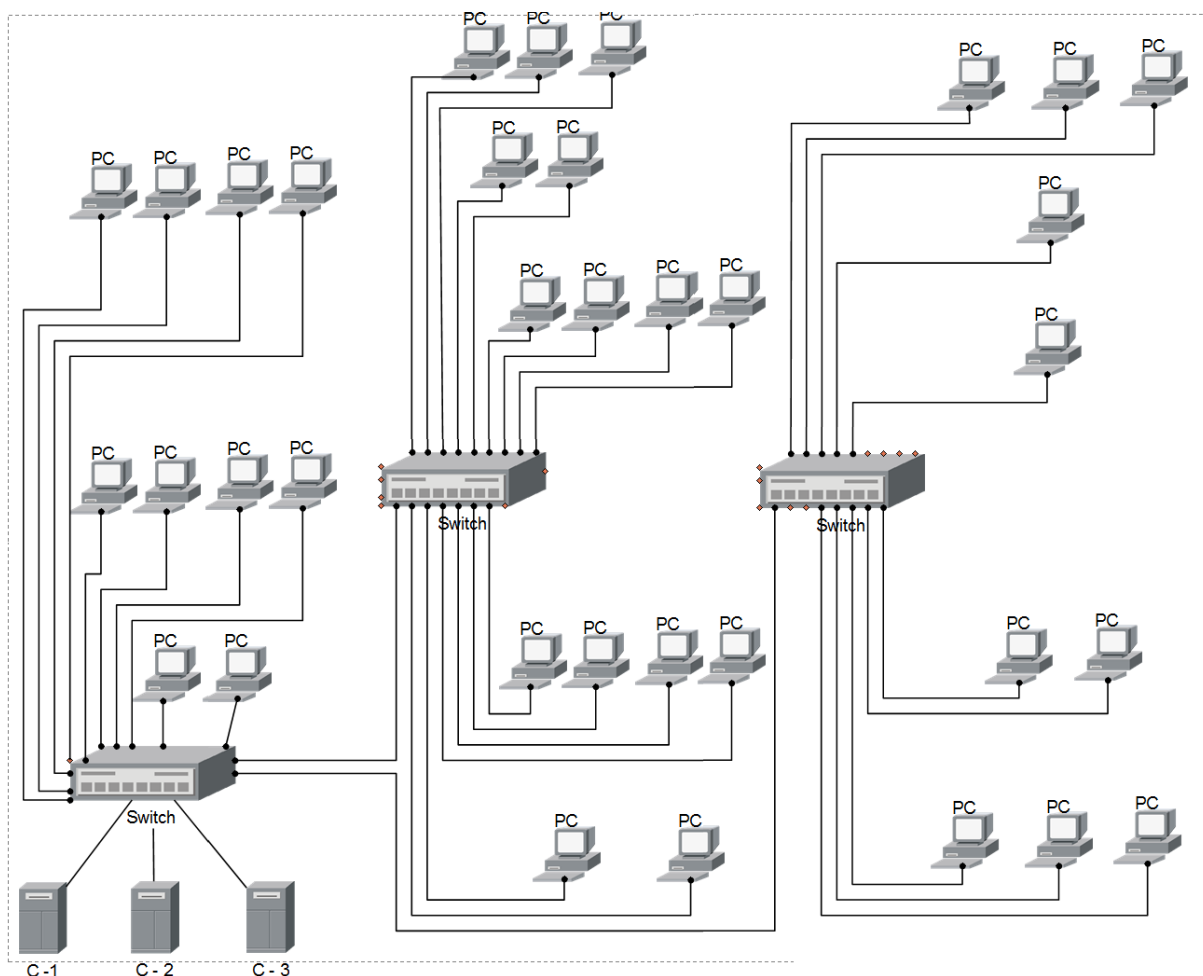


Рис. 2.1. Мережа спортивного клубу «SPA-Фітнес»

Для організації процесу передач даних між робочими станція фітнес-клубу використана комп'ютерна мережа із топологією зірки. Архітектура такого типу комп'ютерної мережі полягає в тому, що всі її вузли сполучені з одним центральним вузлом (рис. 2.2). Мережі з такою топологією є найпоширенішими. Вузол, у мережі такого типу, несе відповідальність за маршрутизацію трафіка до інших компонентів через себе та локалізацію несправностей.

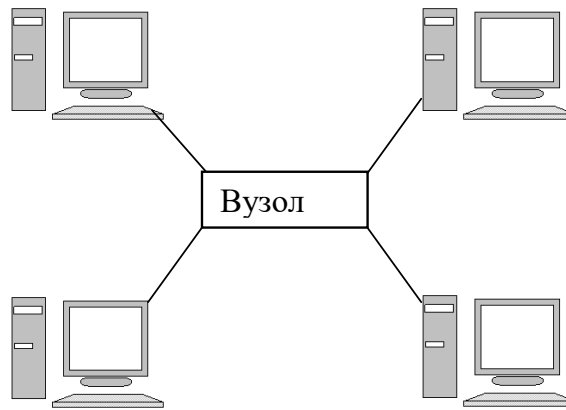


Рисунок 2.2. Топологія зірки

Зазначимо позитивні властивості мереж з топологією зірки. До них належать:

- легкість керування;
- не складне програмне забезпечення;
- можливість швидкого виявлення та ліквідації помилок;
- невеликі витрати та поміркована простота при потребі нарощування мереж.

Фізичний зв'язок між комп'ютерами – це восьмижильна екранована вита пара, яка дає змогу досягати швидкості передачі даних до 100 Мбіт/с. На відміну від коаксіального кабелю (швидкість – 10 Мбіт) і від оптоволоконного (до 1000 Мбіт) вита пара є найкращим співвідношенням ціни, швидкості та якості передачі даних. Кабель на основі витих пар (UTP/STP) містить дві або більше пари проводів, закручених один навколо одного і одночасно навколо інших пар в межах однієї оболонки по всій довжині кабелю. Скручування дозволяє підвищити завадостійкість та знизити вплив кожної пари на всі інші.

Ефективність взаємодії робочих станцій у локальних мережах багато в чому визначається використанням правил доступу до загального передавального середовища в мережах із шинною, кільцевою і топологією концентратора в деревоподібних та зіркоподібних мережах. Правило, за допомогою якого організовується доступ робочих станцій до передавального середовища на організації має назву «метод випадкового доступу».

Наступним компонентом комп'ютерних мереж є мережеві протоколи. Вони представляють собою сукупність правил і принципів, згідно з якими комп'ютери взаємодіють між собою. У мережі використовують протоколи TCP/IP. Останній почав розроблятися міністерством оборони США для експериментальної мережі ARPANET. Цей стек є найбільш поширеним, він забезпечує широкий спектр можливостей, включаючи маршрутизацію. Кожному комп'ютеру присвоюється унікальна IP-адреса і маска, яка задає діапазон IP-адрес в локальній мережі.

Підтримка TCP/IP є вбудованою в найбільш поширені операційні системи (UNIX, Windows 9x/NT). Також він є основним протоколом глобальної мережі Інтернет.

Оскільки канал для передачі даних в мережі є загальним, то кожний вузол в мережі має мати можливість відрізнити ті дані, які призначені саме для нього. Для того щоб здійснити це, дані в мережі розповсюджуються у вигляді пакетів. Кожний такий пакет повинен містити в собі представлену в спеціальному форматі інформацію про те, який мережений пристрій передав в мережу цей пакет, і для якого мереженого пристрою він призначений. При поширенні сигналу по мережі кожен комп'ютер перевіряє надходження необхідного пакету, якщо так, то він його розшифровує.

Підсумовуємо вище наведену інформацію: функцією мережевого адаптера є передача й прийом мережевих сигналів із кабелю. Адаптер сприймає команди і дані від операційної системи, перетворює цю інформацію в один із стандартних форматів та передає останню в мережу через підключений до нього кабель.

2.2. Розроблення архітектури програмної системи

Програмне забезпечення реалізується на основі тривірневої архітектури та передбачає наявність наступних компонент програми: до сервера за стосунків підключено клієнтський застосунок (реалізація – «тонкий клієнт»),

котрий, крім того, підключено до серверу бази даних. Розглянемо більш детально кожен компонент програми (рис. 2.3).

Трирівнева архітектура:
- «клієнт – перший рівень, інтерфейс ний компонент»;
- «сервер за стосунків – другій рівень»;
- «сервер бази даних – третій рівень»

Рис. 2.3. Реалізація програмного забезпечення на основі трирівневої архітектури

Дамо характеристику рівнів.

1. Клієнт — за стосунок для кінцевого користувача, а також інтерфейс ний компонент, представлений першим рівнем. Без прямих зв'язків з БД відбувається реалізація першого рівня (з метою безпеки), виконуючи вимогу масштабованості, не буде навантаження бізнес – логікою та відповідаючи вимозі надійності буде зберігати стан програми. Отже, на першому рівні відображена найпростіша бізнес – логіка, зокрема: інтерфейс авторизації, перевірка вхідних значень, допустимість та відповідність формату.

2. Що стосується другого рівня, на ньому розташовується сервер застосунків, де значиться бізнес – логіка системи.

3. На третій рівень виноситься сервер БД де забезпечує зберігання останніх (процес реалізації відбувається в якості об'єктно – орієнтованої СУБД або стандартної реляційної).

Процес реалізації програмного продукту виглядає таким чином: сервер, фізично під'єднаний з сервером БД комп'ютера, до якого підключено декілька/один термінали.

Основними характеристиками даної трирівневої структури є Таблиця 2.1:

Таблиця 2.1

Характеристики трирівневої структури

Назва	Визначення
- масштабованість	«...здатність підтримувати великі кількості архітектурних компонентів та з'єднань між ними» [вікіпедія];
- конфігурованість	«...ізолюваність рівнів один від одного дозволяє (за умови правильного розгортання архітектури) швидко та простими засобами перебудувати систему у разі виникнення збоїв або під час планового обслуговування на одному з рівнів»;
- високий рівень безпеки	
- висока надійність	
- низькі вимоги до швидкості каналу/мережі між терміналами і сервером застосунків	
- низькі вимоги до продуктивності і технічних характеристик терміналів, відповідно, зниження їхньої вартості (терміналом може бути не лише комп'ютер, як варіант – телефон)	

Структурна схема програмного забезпечення реалізується, як трирівнева аплікація, що передбачає наступні рівні:

- рівень клієнта;
- сервер бізнес-логіки;
- сервер баз даних.

Взаємодія між цими рівнями відображена на рис. 2.4. (ДОДАТОК Д).

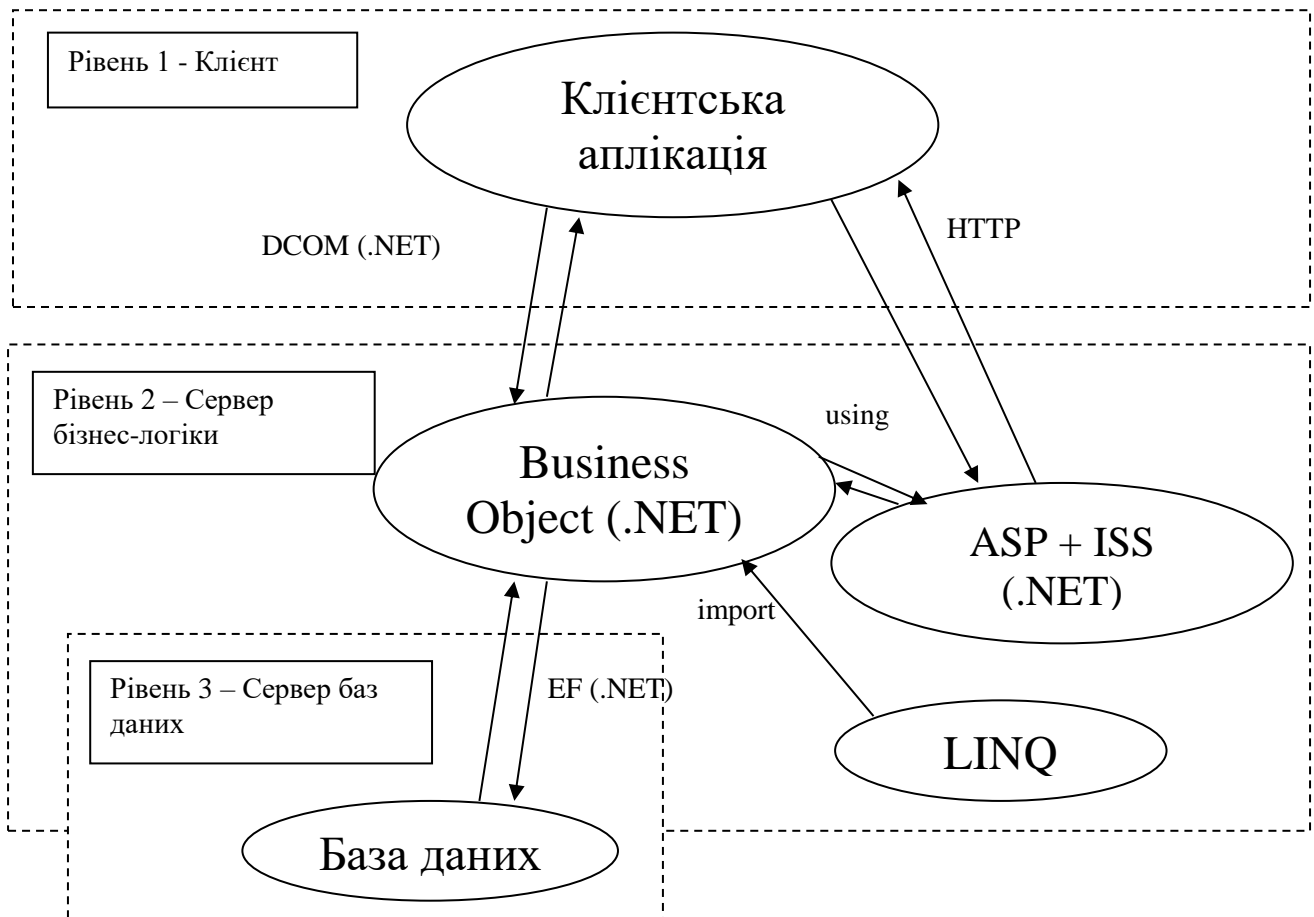


Рисунок 2.4. Архітектура програмного додатку

Кожен рівень реалізується на основі певних платформ та технологій. Розглянемо детально кожен із рівнів:

Клієнтський рівень реалізується із використанням мови гіпертекстової розмітки HTML, каскадних стилів CSS, а також скриптової мови JavaScript, що забезпечують динамічність сторінки.

Все частіше використання «чистого» JavaScript коду, замінюється на javascript-фреймворками. Найпопулярніший із них jQuery, що об'єднує у собі цілу екосистему бібліотек, побудованих навколо базової бібліотеки: це і бібліотека jquery.ui, призначена для створення візуальних інтерфейсів, це і jquery.mobile, використовувана при розробці мобільних сайтів та ін. Переваги використання jQuery відображені на рисунку (рис. 2.5).

jQuery – позитивні особливості:
- «...виділення коду розмітки від поведінки. Розмітка html стає чистою та зрозумілою коли винесемо код javascript з останньої»;
- «...спрощення роботи з кодом. Задля маніпулювання елементами на web – сторінці jQuery пропонує простий елегантний синтаксис»;
- «...розширюваність. Весь код jQuery відкритий для перегляду та внесення змін. Якщо щось не влаштовує в бібліотеці, можна здійснити модифікацію»;
- «кроссбраузерність. jQuery має підтримку більшості відомих браузерів, у тому числі такі, як IE 7, 8»

Рисунок 2.5. Переваги застосування jQuery

Сервер бізнес-логіки реалізується на платформі .NET, яка являється потужною основою для створення додатків. Основними позитивними рисами даної платформи можна вважати: (рис. 2.6).

Риси платформи jQuery:
- «підтримка декількох мов»;
- «кроссплатформеність»;
- «потужна бібліотека класів»;
- «різноманітність технологій»

Рисунок 2.6. Основні риси jQuery

Тепер перейдемо до характеристики основних рис вищезгаданої платформи.

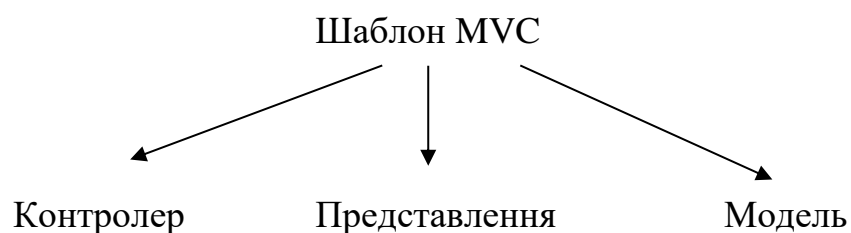
«Підтримка декількох мов». Основою платформи є загальномовне середовище виконання Common Language Runtime (CLR), завдяки чому .NET підтримує декілька мов: поряд з C# це також VB.NET, C++, F#, а також різні діалекти інших мов, прив'язані до .NET, припустимо, Delphi.NET. При компіляції код на будь-якому з цих мов компілюється в збірку спільною мовою CIL (Common Intermediate Language) – свого роду асемблер платформи .NET. Тому, на окремих мовах можна розроблювати окремі модулі однієї програми.

«Кросплатформеність». .NET є переносимою платформою. Так, остання версія платформи на даний момент .NET Framework 4.5.1. підтримується на більшості сучасних ОС Windows (Windows 8.1/8/7/Vista). А завдяки проекту Mon є можливість створювати додатки, які будуть працювати на інших ОС сімейства Linux, в тому числі на мобільних платформах Android і iOS.

«Потужна бібліотека класів». .NET представляє єдину для всіх підтримуваних мов бібліотеку класів.

«Різноманітність технологій». Загальномовне середовище виконання CLR і базова бібліотека класів є основою для цілого стека технологій, які розробники можуть використати при побудові різних додатків. «...Припустимо, для роботи з базами даних в цьому стеці технологій призначена технологія ADO.NET. Для побудови графічних додатків з багатим насиченим інтерфейсом – WPF технологія. Для створення web -сайтів – технологія ASP.NET і т. п.» [46, 47].

На основі вище викладених особливостей платформи .NET проєктована система буде реалізована на технології ASP.NET MVC, що використовується для створення веб-додатків з використанням патерну (або шаблону) MVC (model - view - controller). Шаблон MVC, що лежить в основі платформи реалізує взаємодію трьох компонентів:



«Контролер (controller)» - це клас, з якого бере початок роботи програма. Він здійснює зв'язок між моделлю і представленням. Після введення даних користувачем, включаючи внутрішню логіку, контролер, за необхідності, звертається до моделі та генерує певне подання.

«Представлення (view)» – користувацький інтерфейс програми/візуальна частина. Припустимо, користувач, через html-сторінку, заходить на сайт, і в подальшому взаємодіє з web -додатком.

«Модель (model)» - це група класів, що займаються описом логіки використовуваних даних.

ASP.NET MVC – можна вважати конкурентом web -форм (ASP.NET Web Forms, PHP). Якщо їх порівнювати, тоді можна зазначити такі переваги:

- «...Розподіл відповідальності. MVC додаток має складові, які виконують свої специфічні функції й до них належать: контролер, уявлення, моделі. Відповідно, додаток легше підтримувати та модифікувати в майбутньому».

- «...Поділ відповідальності. У додатках створених на основі mvc можна краще реалізовувати тестування (тестування окремих компонентів незалежно один від одного)».

- «...Відповідність протоколу HTTP. На відміну від web –форм, додатки MVC не підтримують об'єкти стану (ViewState). Тому, більшого контролю над роботою додатка дозволяють домогтися простота та зрозумілість самої платформи».

- «...Гнучкість – дає можливість налаштувати компоненти платформи залежно від свого бачення, досвіду та, в залежності від своїх потреб, змінювати різні частини конвеєра роботи MVC» [44, 45].

У порівнянні із технологією ASP.NET Web Forms використання ASP.NET MVC не надає можливості контролю над розміткою сторінки (перегляд вигляду сторінки в реальному часі у візуальному редакторі).

2.3. Проектування структури бази даних

Рівень сервера баз даних розробленого програмного забезпечення здійснюється із використанням системи Microsoft SQL Server. Дана система

управління використовується для різних підприємств за масштабом, починаючи від невеликих і середніх за розміром БД, так і для великих БД.

Мова, що використовується для створення структури бази даних є Transact-SQL. Transact-SQL є реалізацією стандарту ANSI/IS щодо структурованої мови запитів (SQL) із розширеннями. T-SQL дозволяє використовувати додатковий синтаксис процедур, що зберігаються і забезпечує підтримку транзакцій (взаємодія бази даних з керуючим застосунком).

Технологія, яка забезпечує обмін даними ґрунтується на використанні Entity Framework. Даний засіб взаємодії представляє спеціальну об'єктно-орієнтовану технологію на базі фреймворка .NET для роботи з даними. Entity Framework дозволяє абстрагуватися від бази даних і працювати з даними незалежно від типу сховища, тобто здійснюється перехід від фізичного рівня, зокрема: оперування таблицями, індексами, первинними і зовнішніми ключами й до концептуального рівня (оперування з сутностями системи) управління даними. При цьому будь-яка сутність, як і будь-який об'єкт з реального світу, має ряд властивостей за допомогою, яких реалізується опис даної сутності (масив інформації, що описує об'єкт). «...За аналогією із реляційною моделлю у кожній сутності може бути одне (декілька властивостей), які відрізнятимуть цю сутність від інших і будуть унікально визначати цю сутність (первинні ключі таблиць баз даних). Також сутності можуть бути пов'язані асоціативними зв'язками один-до-багатьох, один-до-одного і багато-до-багатьох, подібно до того, як в реальній базі даних відбувається зв'язок через зовнішні ключі» [1, 2].

«...Особливістю Entity Framework можна вважати використання запитів LINQ для вибірки даних з БД. Завдяки LINQ можна не тільки отримувати відповідні рядки, що зберігають об'єкти з БД, а ще й об'єкти, пов'язані різними асоціативними зв'язками» [2].

Модель Entity Data Model з'являє класи сутностей з реальними таблицями в базі даних та складається з трьох рівнів, що відображені на рисунку 2.7.

Складові Entity Data Model:
- «концептуальний рівень»;
- «рівень сховища»;
- «рівень зіставлення (маппінга)»

Рисунок 2.7. Складові моделі Entity Data Model

- На *концептуальному рівні* відбувається визначення класів сутностей, що використовуються в додатку.
- *Рівень сховища* визначає таблиці, стовпці, відносини між таблицями і типи даних, з якими зіставляється використовувана база даних.
- *Рівень зіставлення (маппінга)* служить посередником між попередніми двома, визначаючи зіставлення між властивостями класу сутності і стовпцями таблиць.

Таким чином реалізується процес взаємодіяти з таблицями з БД через класи, визначені в додатку. Також Entity Framework передбачає три можливі способи взаємодії з базою даних:

- **«Database first:** Entity Framework створює набір класів, які відображають модель конкретної БД»;
- **«Model first:** спочатку розробник створює модель БД, по якій потім Entity Framework створює реальну БД на сервері»;
- **«Code first:** розробник створює клас моделі даних, які будуть зберігатися в БД, а потім Entity Framework з цього моделі генерує БД та її таблиці» [44].

Як бачимо, дані підходи призначені для пришвидшення розробки програмного продукту і надають дивелопера різноманітні варіанти для проектування БД і концептуальних моделей. У даному випадку реалізація БД програмного продукту здійснюється на основі підходу Code first. Даний підхід дозволяє проектувати БД навіть не знаючи мову SQL. Хоча він має недоліки, оскільки за реалізації більш складніших зв'язків між даними виникає потреба у створенні додаткових моделей, що їх реалізовуватимуть.

При проведенні дослідження, в якості інструменту для моделювання системи, був обраний програмний продукт VPwin від фірми Platinum technology/Logic Works. Він позиціонується для використання проектувальниками, аналітиками, розробниками. VPwin – це CASE-засіб візуального проектування інформаційних систем, що дозволяє моделювати бізнес- процеси, що реалізує метод IDEF1X. Поточна версія VPwin підтримує також діаграми потоків даних і потоків робіт.

На етапі моделювання підсистеми використовуються нотації SADT (Structured Analysis and Design Technique) і DFD (Data Flow Diagrams), оскільки вони дозволяють найбільш адекватно відобразити логіку предметної області і легко сприймаються для аналізу. Діаграми потоків даних і діаграми "сутність-зв'язок" - найбільш часто використовувані в CASE-засобах види моделей. На стадії формування вимог до розроблюваної підсистемі SADT-моделі і DFD використовуються для побудови моделі "AS-IS" і моделі "TO-BE", відображаючи, таким чином, існуючу і пропоновану структуру бізнес-процесів спортивного клубу та взаємодію між ними.

Процес розробки програмної системи (створення програмного коду) здійснюється із використанням програмного продукту Microsoft Visual Studio – серія продуктів фірми Майкрософт. До останніх відноситься інтегроване середовище розробки програмного забезпечення, а також інші інструментальні засоби. Зазначені продукти дають можливість здійснити розробку консольних програм, з графічним інтерфейсом програм (з підтримкою технології Windows Forms), web –сайти (застосунки, служби) не лише в рідному, а й в керованому кодах для різних платформ, що підтримуються Microsoft Windows, Windows Mobile, Windows CE, .NET Framework, .NET Compact Framework та Microsoft Silverlight.

Для функціонування розробленого програмного продукту необхідно, щоб були використані наступні програмні засоби:

- операційна система сімейства Windows для організації діяльності працівників фітнес-клубу на локальній машині;

- операційні системи сімейства Windows, що дозволятимуть організувати роботу локальної станції, як сервери.

Розглянемо детальніше версії ОС, які мають бути встановлені на ПК клієнтів ФК:

Windows XP Home Edition — система для домашнього застосування. Володіє наступними можливостями: шифрування файлів (при допомозі Encrypting File System), центральне управління правами доступу і підтримка багатопроцесорних систем. Характеристики ОС є наступні:

- Процесор: 300 MHz або вище
- Оперативна пам'ять: 128 Мб RAM або вище
- Відеоадаптер і монітор: Super VGA (800 x 600) або більше розширення
- Вільне місце на HDD: 1.5 Гб або вище
- Оптичні накопичувачі: CD-ROM або DVD-ROM.

Крім даної операційної системи, що є найпоширенішою можуть використовуватися і новіші версії, однією із таких популярних версій є Windows 7 Home.

При організації роботи з клієнтом, основне - це робота з браузером, і неважливо, яка операційна система встановлена.

Оскільки, реалізації сучасних веб-переглядачів передбачає кросбраузерність, то можна використовувати й інші ОС. Зокрема безкоштовні версії ОС сімейства Unix та найбільш поширена серед Unix подібних систем – Linux.

Для роботи фітнес клубу, його працівників, будемо використовувати операційну систему Windows Vista / 7 / 8, яку в подальшому, опишемо та розглянемо.

Windows 7 Ultimate – переважно застосовується корпоративними клієнтами або клієнтами з різнобічними можливостями. Реалізація системи здійснюється для x32і x64 бітних систем. Ця система дає можливість працювати в будь – якому місці знаходження, оптимізує управління

комп'ютером, покращує контроль і безпеку, має гнучкий інтерфейс, підтримує різні мови, а також поєднує розважальні функції Windows 7 Home Premium та функції для бізнесу випуску Windows 7 Professional. Крім вище зазначених можливостей, цей випуск включає наступні (рис. 2.8):

Windows 7 Ultimate:	
-	«...шифрування диска та захист даних Microsoft BitLocker і BitLocker To Go на внутрішніх і зовнішніх дисках, пристроях зберігання»;
-	«...технологія DirectAccess – більш безпечна для підключення до корпоративної мережі Інтернет»;
-	«...технологія Microsoft BranchCache – забезпечує прискорений доступ до вмісту, віддалених файлових, web-серверів філій»;
-	«...технологія Microsoft AppLocker – запобігає запуску несанкціонованого програмного забезпечення на комп'ютерах працівників»;
-	«...області пошуку в корпоративному середовищі, що полегшують виявлення, пошук вмісту на порталах інтрамережі»

Рис. 2.8. Можливості Windows 7 Ultimate

Таблиця 2.2

Порівняльні характеристики ОС Windows для 32 і 64 розрядних версій

	32-бітна версія Windows 7	64-бітна версія Windows 7
Центральний процесор	x86 чи x86-64 з тактовою частотою 1 ГГц	x86-64 з тактовою частотою 1 ГГц
Оперативна пам'ять	1 Гб	2 Гб
Відеоадаптер	З підтримкою DirectX 9 та драйвера WDDM версії 1.0 та вище.	
Вільний дисковий простір	16 Гб	20 Гб
Оптичні накопичувачі	DVD-RW	

Для організації діяльності роботи локальної машини, як сервера можна використати, як звичні ОС Windows версії Vista та вищі, так і спеціалізовані ОС Windows Server. Розглянемо детальніше спеціалізовані версія ОС:

Windows Server 2008 / R2 / 2012 – може здійснювати декілька головних ролей, зокрема: може бути контролером домену/Доменні служби Active Directory, ADLDS (ADAM), DNS-сервер, DHCP сервер, файловий сервер,

сервер друку, Windows Media Server, IIS 7 web -сервер і Hyper-V віртуальний сервер. Переписаний мережевий стек є особливістю такої версії (протокол IPv6, протокол бездротової мережі), на основі установки, відновлення та розгортання здійснюється покращення зображень і діагностики, крім того моніторингу, реєстрації подій, також формування звітності. Щодо до нових функції безпеки поліпшення зображень на основі установки, розгортання і відновлення, поліпшення діагностики, моніторингу, реєстрації подій і звітності; нові функції безпеки, також притаманні цій системі BitLocker і ASLR (Address Space Layout Randomization); покращений Windows Firewall з безпечної конфігурації за замовчуванням; .NET Framework технології, зокрема, Windows Communication Foundation, Microsoft Message Queuing і Windows Workflow Foundation, і поліпшення файлової системи. Пам'ять пристрою та процесори моделюються як Plug and Play пристрою. Це дає великі можливості системним ресурсам бути розділеними динамічно за допомогою динамічного апаратного секціонування; крім того, кожен розділ має свою власну пам'ять, процесор і I/O міст пристрою незалежно від інших розділів.

Крім використання ОС потрібно ще використовувати програмні засоби, що організовуватимуть роботу сервера (веб-сервер IIS 7.5) та діяльність із вебдодатком (браузери).

Описані вище ОС необхідні для організації роботи на локальній станції, що виконуватиме роль сервера в обслуговуванні роботи системи. Серед всіх версій web -сервера IIS найбільш продуктивну роботу може виконувати серія IIS 7.5. у даній серії реалізовано модульну архітектуру, що дозволяє здійснювати вибірку системним адміністратором для роботи необхідних модулів. Web -сервер випускається із такими стандартними модулями, як HTTP, безпеки, контенту, стиснення, кешування, протоколювання та діагностики. Всі вище описані web -сервери підтримують аутентифікації, що реалізується наступними механізмами:

- «...Базова аутентифікація (Basic access authentication)»;
- «...Стисла аутентифікація (Digest access authentication)»;

- «...Інтегрована аутентифікація Windows (Integrated Windows Authentication)»;
- «...NET Passport аутентифікація» [44, 45].

Конфігураційна інформація даного сервера зберігається лише в конфігураційних XML файлах (не в метабазах). Сервер має глобальний конфігураційний файл, який містить налаштування за замовчуванням, і кожна коренева віртуальна web -директорія (аналогічно її піддиректорії) можуть містити web.config файл (у ньому глобальні налаштування розширюються/заміщуються). Файли починають діяти відразу після внесених змін. IIS 7.5 також має змінений повністю адміністративний інтерфейс. Останній використовує панель задач та асинхронні операції. Налаштування ASP.NET ще більше інтегроване в адміністративний інтерфейс.

Дана серії web -серверів використовуються разом із наступними операційними системами:

- Windows Vista та вищі версії (7/8);
- Windows Server 2008 та вищі версії (Server 2008 R2, 2012).

Для організації роботи клієнтів та працівників із розробленою аплікацією необхідно, щоб було встановлено браузер Google Chrome 35.0.1961. Зазначений браузер має певні особливості, які надають йому не аби якої популярності:

- За допомогою системи Chrome постійно відбувається завантаження оновлених чорних списків (фітінгові сайти та сайти, що містять шкідливе програмне забезпечення), попереджаючи користувача про небезпеку.
- Даною системою для кожної вкладки створюється окремий процес (з метою унеможливлення впливу на вміст одна на одну). Крім того, при зависанні процесу обробки вмісту вкладки є можливість відновлення без ризику втрати даних інших вкладок.
- Наступна властивість - процесу вкладки притаманне обмежена кількість привілеїв, зокрема, можна здійснювати обчислення, а функція -

записувати або зчитувати неможлива. Інакше кажучи, вкладка знаходиться в пісочниці.

- Google Chrome використовує високопродуктивний спосіб обробки JavaScript V8. Також, для прискорення завантаження сторінок Google Chrome може використовувати попередній виклик DNS.

- За допомогою цієї системи можна розширити функції та можливості браузера. Особливістю є те, що розширення дозволяють додавати в Google Chrome лише необхідні можливості, при цьому уникаючи накопичення зайвих функцій. Розширення дозволяють додавати в Google Chrome тільки потрібні можливості, уникаючи накопичення функцій, які не використовуються. Google створив спеціальну галерею розширень від третіх осіб.

- Позитивним є постійне автоматичне оновлення Google Chrome при виявленні нової версії браузера, що відбувається у фоновому режимі та не вимагає жодних дій від користувача.

Висновки за розділом 2

Мережа спортивного клубу «SPA-Фітнес» налічує до сорока робочих станцій, три комутатори, крім того використовуються різноманітні загальнопоширені і спеціалізовані апаратні та програмні засоби та три спеціалізовані сервери:

- 1) С-1 - відповідає за організацію доменної мережі, за допомогою якого організовується доступ та реєстрації користувачів при вході в домен.
- 2) С-2 - сервер баз даних білінгових та інших програм;
- 3) С-3 - сервер, що керує доступом всіх робочих станцій до загальних серверів фітнес-клубу та до мережі Інтернет.

Для організації процесу передач даних між робочими станціями фітнес-клубу використана комп'ютерна мережа із топологією зірки. Архітектура

такої комп'ютерної мережі полягає в тому, що усі вузли мережі сполучені з одним центральним вузлом.

До позитивних властивостей мереж з топологією зірки можна віднести такі:

- легкість керування;
- просте програмне забезпечення;
- можливість швидкого виявлення і ліквідації помилок;
- відносна простота та незначні витрати при потребі нарощування

мереж.

Програмне забезпечення реалізується на основі трирівневої архітектури та передбачає наявність наступних компонент програми: клієнтський застосунок (реалізується, як «тонкий клієнт»), підключений до сервера застосунків, який в свою чергу підключений до серверу бази даних.

Структурна схема програмного забезпечення реалізується, як трирівнева аплікація, що передбачає наступні рівні:

- рівень клієнта;
- сервер бізнес-логіки;
- сервер баз даних.

Рівень сервера баз даних розробленого програмного забезпечення здійснюється із використанням системи Microsoft SQL Server. Дана система управління використовується як для невеликих і середніх за розміром БД, так і для великих БД масштабу підприємства.

Мова, що використовується для створення структури бази даних є Transact-SQL. Transact-SQL є реалізацією стандарту ANSI/IS щодо структурованої мови запитів (SQL) із розширеннями. T-SQL дозволяє використовувати додатковий синтаксис процедур, що зберігаються і забезпечує підтримку транзакцій (взаємодія бази даних з керуючим застосунком).

Технологія, яка забезпечує обмін даними ґрунтується на використанні Entity Framework. Даний засіб взаємодії представляє спеціальну об'єктно-

орієнтовану технологію на базі фреймворка .NET для роботи з даними. Entity Framework дозволяє абстрагуватися від бази даних і працювати з даними незалежно від типу сховища, тобто здійснюється перехід від фізичного рівня (оперування таблицями, індексами, первинними і зовнішніми ключами) до концептуального рівня (оперування з сутностями системи) управління даними. При цьому будь-яка сутність, як і будь-який об'єкт з реального світу, має ряд властивостей з допомогою, яких реалізується опис даної сутності (масив інформації, що описує об'єкт).

РОЗДІЛ 3

ПРОГРАМНА РЕАЛІЗАЦІЯ WEB –ДОДАТКУ ДЛЯ АВТОМАТИЗАЦІЇ УПРАВЛІННЯ ПРОЦЕСОМ НАДАННЯ ПОСЛУГ КЛІЄНТАМ СПОРТИВНОГО КЛУБУ

3.1. Програмна реалізація системи

Зміни в організаційній структурі об'єкту пов'язані із планами дирекція ФК, щодо розширення асортименту послуг та впроваджені нових напрямків діяльності. Передбачається створення центру із надання послуг із SPA-процедур, створення нових тренувальних програм у групах, що відповідатимуть потребам клієнтів.

Реалізація даного задуму потребує не тільки додаткового персоналу, а й зміни структури роботи вже наявних відділів, а саме відділ надання послуг із групових занять. Під поняттям «групові заняття» розумітимемо заняття у секціях (наприклад, занятті із айкідо, аеробіки тощо). Під поняттям «індивідуальні заняття» розумітимемо заняття у тренажерних залах, зокрема фітнес-тести, надання консультативних послуг клієнтам. Крім роботи із групами тренера можуть проводити індивідуальні заняття із клієнтами у вільний час, щоб більше приділяти уваги до нових клієнтів.

Крім звичних послуг буде створено SPA-центр, який розширить список послуг (послуги дієтологів, гідромасажі, spa-процедури тощо), а значить збільшить кількість клієнтів, які орієнтовані лише на дану сферу діяльності.

Коригування діяльності усього ФК здійснюватиме управляючий, який співпрацюватиме із старшими менеджерами кожного відділу.

Особливістю СК за типом фітнес-клубу є невелика кількість основного персоналу для організації діяльності закладу. Загальна кількість працівників, які працюють в організації становить 35 осіб. Організаційна структура фітнес-клубу складається із таких структурних підрозділів:

Підрозділ допоміжного персоналу – дана структура включає персонал, що займається організацією та підтримкою чистоти фітнес-клубу, здійснює контроль із охорони закладу. Чисельність даного підрозділу становить 6 осіб.

Підрозділ «Програмного забезпечення та управління кадрами» – відповідає за організацію роботи із програмним забезпеченням, технічним оглядом техніки, поширення інформації про ФК та роботою із працівниками структури. Чисельність даного підрозділу становить 4 особи. Даний підрозділ включає такі посади (рис. 3.1).

Посади:
- менеджер з підбору персоналу;
- розробники програмного забезпечення (девелопери);
- маркетолог

Рис. 3.1. Посади підрозділу «Програмного забезпечення та управління кадрами ФК»

Зазначимо їх посадові обов'язки.

- Менеджер підбору персоналу – відповідає за пошук та найм працівників на роботу, стежить за їх трудовою діяльністю, слідкує за їхнім кваліфікаційним рівнем;
- Девелопери (розробники програмного забезпечення) – відповідають за розробку, удосконалення та супровід автоматизованої системи – розробка і технічна підтримка сайту. Технічний огляд комп'ютерного забезпечення, що є в наявності клубу;
- Маркетолог – відповідає за вивчення потреб клієнтів фітнес-клубу, наповненості сайту матеріалами та поширення сайту у пошукових системах.

Підрозділ «Бухгалтерія» – персонал даного відділу відповідає за кошторис клубу (стежить за станом покупок на матеріально-технічне забезпечення), стежить за процесом оплати та нарахування заробітної плати працівників. У даному відділі можуть працювати одна або дві особи.

Підрозділ для роботи із індивідуальних занять з клієнтами – виконує діяльність із організації індивідуальної роботи з клієнтами, що здійснюється у

тренажерних залах . Даний підрозділ налічує 7 осіб. Управляє діяльність даних осіб старший менеджер.

- Тренер із індивідуальних занять – виконують індивідуальні заняття у визначені дні та години виходячи із потреб клієнтів, що включає вихідні дні. Графік роботи тренерів становить половину робочого дня. Тривалість одного заняття становить половину години.

- Старший менеджер із індивідуальних занять – складає графік проведення індивідуальних занять, назначає фітнес-тренерів для роботи з індивідуальних занять з клієнтом, слідкує за процесом діяльності тренерів тощо.

Підрозділ для роботи із групових занять з клієнтами – виконує діяльність із організації групової роботи з клієнтами. Даний підрозділ налічує 5 осіб, графік роботи, яких становить погодинно. Управляє діяльність даних осіб старший менеджер.

- Тренер із групових занять – виконують групові заняття у визначені дні згідно графіка, що включає вихідні дні. Чисельність групи для кожного тренера становить 20 осіб. Протягом дня групові заняття можуть організовуватися 3-4 рази для одного тренера.

- Старший менеджер із групових занять – складає графік проведення групових занять, назначає фітнес-тренерів для роботи із групами клієнтів, слідкує за процесом діяльності тренерів тощо.

Підрозділ для роботи із клієнтами у SPA-центрі – відповідає за формування графіку надання SPA-процедур, закупівлю необхідного матеріалу тощо. Чисельність даного підрозділу становить 10 осіб та включає наступних контингент працівників:

Дієтологи – розробляють індивідуальні графіки харчування клієнтів, надають консультативні послуги, щодо вибору оздоровчих процедур на основі медичних обстежень клієнта. Даний напрям роботи очолюватиме старший менеджер SPA-центру.

- Спеціалісти із SPA-процедур – надають послуги із spa-процедур.

- Старший менеджер для роботи із клієнтами у SPA-центрі – складає графік діяльності SPA-центру, слідкує за процесом діяльності персоналу тощо.

Управління діяльністю клубу, його підрозділів керують відповідні менеджери, якими являються наступні особи:

- IT-менеджер – старший програмний розробник. Відповідає за організацію діяльності девелоперів та маркетолога;
- Старший менеджер із індивідуальних занять;
- Старший менеджер із групових занять;
- Старший лікар – управляє персоналом SPA-центру;
- Управлінський менеджер – виконує функцію контролю за персоналом фітнес-клубу. Працює із старшими менеджерами різних напрямків робіт.

Реорганізація існуючих підрозділів передбачає розподіл діяльності групових та індивідуальних занять, а також створення нового напрямку діяльності – spa-центру.

Розмежування відповідальності між структурами дозволить розподілити роботи між тренерами, визначити їх години роботи, сформувати чіткі групи клієнтів щодо працівників ФК. Кожен відділ очолюватиме старший менеджер певного напрямку діяльності.

Взаємодія між відділами групових та індивідуальних занять здійснюється наступним чином:

- тренера, що працюють із групами можуть мати індивідуальні заняття із клієнтом, але не більше 6 осіб протягом дня. Дана діяльність узгоджується із старшим менеджером індивідуальних занять та проводиться у вільний час тренера між груповими заняттями;
- тренера із індивідуальних занять виконують свою діяльність у тренажерних залах та виконують консультативні функції (фітнес-тести; робота із клієнтами, які не є групі).

Як бачимо дана взаємодія структур не прив'язує тренера до певного виду діяльності, а й дає можливість змінювати їх.

Проблема складності є головною проблемою, яку доводиться вирішувати при створенні систем будь-якої природи, в тому числі й ЕОС. Жоден розробник не в змозі вийти за межі людських можливостей і зрозуміти всю систему в цілому. Однак, можна знайти вихід з цієї ситуації, якщо створити (побудувати) складну систему, що містить кілька частин, а останні, у свою чергу, складаються з частинок ще меншого розміру і т.п., поки процес не набуде зворотного шляху, зокрема: побудова невеликих частин з наявного матеріалу. Цей процес (підхід) має назву ієрархічна декомпозиція. По відношенню до проектування програмної системи це означає, що складну систему маємо розділити на невеликі підсистеми, а тоді, кожен новостворену підсистему у подальшому розробляти незалежно від інших. Це дозволяє нам при розробці підсистеми обліку послуг спортивного клубу тримати в розумі інформацію тільки про неї, а не про всіх інших частинах системи. «Правильна система» по відношенню до декомпозиції має відповідати наступним вимогам: насамперед, мінімальними має бути кількість зв'язків між окремими підсистемами і навпаки, максимальною має бути зв'язність окремих частин в середині кожної підсистеми.

Успіх будь-якого проекту з розробки ПЗ визначається трьома складовими: методологія проектування, система позначень (мова, нотація) і інструмент.

Методологія (процес) описує послідовність кроків, моделі та підходи, проходження яким призводить до добре працюючим системам. Хоча методологія не гарантує якості систем, тим не менш, вона допомагає врахувати всі важливі етапи розробки, забезпечує організаційну підтримку [29, 30].

На сьогоднішній день в програмній інженерії існують два основних підходи до розробки програмного забезпечення ЕОС, принципова відмінність між якими обумовлено різними способами декомпозиції систем (рис. 3.2).

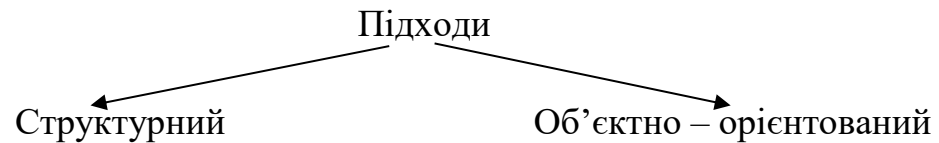


Рисунок 3.2. Класифікація підходів до розробки програмного забезпечення

Перший підхід має назву – функціонально – модульний (структурний). Суть його полягає у застосуванні принципу функціональної декомпозиції, коли структура системи подається в термінах ієрархії її функцій та передачі інформації між певними функціональними елементами. Об'єктно – орієнтований підхід містить об'єктну декомпозицію. При даному підході, в термінах об'єктів та зв'язків між ними описується структура системи. Що стосується поведінки системи зазначеного підходу – опис відбувається в термінах обміну повідомленнями між об'єктами.

При розробці підсистеми обліку послуг будемо спиратися на структурний підхід. Розглянемо його більш детально. Сутність такого підходу полягає в декомпозиції (розбитті) та автоматизує такі функції: перше – система дробиться на підсистеми функціональні, які пізніше розбиваються на підфункції, останні, у свою чергу, на завдання, і, кінцевий етап – до конкретних процедур.

Структурний підхід дотримується певних базових принципів до яких належать:

- «...Принцип "розділяй і володарюй"»;
- «...Принцип ієрархічного упорядкування» - коли система набуває ієрархічну деревоподібну структуру з поступовим додаванням на кожному рівні нових деталей;
- «...Принцип абстрагування» - виділення істотних аспектів системи і відволікання від несуттєвих;
- «...Принцип несуперечності» - обґрунтованість і узгодженість елементів системи;

- «...Принцип структурування даних» - дані повинні бути як структуровані так і ієрархічно організовані [44].

Практично у всіх методологіях структурного аналізу використовуються три групи засобів моделювання (нотацій):

- Діаграми, що ілюструють функції, які повинна виконувати система, і зв'язки між цими функціями – для цієї мети найчастіше використовуються нотації SADT і DFD;

- Діаграми, що моделюють дані і їх взаємозв'язок (діаграми сутність-зв'язок – ERD);

- Діаграми, що моделюють поведінку системи (діаграми переходів станів – STD).

Переваги структурного підходу в тому, що при цьому розроблена нами підсистема, зберігає цілісну композицію, де всі компоненти взаємопов'язані. При розробці системи "знизу вгору", від окремих завдань до всієї системи, цілісність втрачається, виникають проблеми при описі інформаційної взаємодії окремих компонентів.

Інструментальна база структурного підходу представлена – широкою номенклатурою CASE-засобів, що підтримують різні методології і види проектної діяльності. Будь – який програмний засіб, що автоматизує певну сукупність процесів життєвого циклу програмного забезпечення, переважно відносять до CASE – технологій. Такий програмний засіб має відповідати характеристикам:

- для опису, документування інформації, а також забезпечення інтерфейсу з наданням у подальшому, в процесі роботи, розвитку та творчої можливості, повинен мати потужні графічні засоби;

- для забезпечення керованості процесом розробки інформаційної системи – інтеграція окремих компонент Інтеграція окремих компонент CASE – технологій;

- застосування специфічно організованого репозиторію (сховища проектних даних)

Розроблене ПЗ дозволяє виконувати наступні напрямки діяльності (рис.

3.3.

Напрями діяльності ПЗ:
- додавання даних у систему про працівників та клієнтів фітнес – клубу;
- додавання даних у систему для формування організаційної структури фітнес - клубу;
- організація процесу відслідковування за отриманими послугами клієнтів фітнес – клубу;
- організація процесу оплати за отримані послуги;
- модифікація та видалення даних введених в систему

Рисунок 3.3. Напрями діяльності програмного забезпечення

Розглянемо більш детально кожний напрям роботи системи у розмежуванні діяльності, щодо можливих операцій.

Операції, що організовують діяльність із даними про працівників фітнес-клубу:

- 1) Операція «Додавання даних про працівника» включає такі кроки:
 - додавання інформації про працівника у БД (загальна інформація);
 - додавання інформації про навчання працівника (сюди включається інформація про курси, тренінги і т. д., які відвідувала дана особа);
 - додавання інформації про трудову діяльність, а також видів робіт, які виконує працівник у фітнес-клубі.
- 2) Операція «Модифікація даних про працівника».
- 3) Операція «Вилучення даних про працівника».

Вище перелічені операції виконує менеджер для роботи з персоналом. Перелічені операції виконуються наступним чином: заповнюється форма реєстрації клієнта та відсилаються дані на сервер, де відбувається їх обробка. Після обробки даних відкривається особиста сторінка працівника, де менеджер реєструє історію трудової діяльності працівника та його навчання. У випадку необхідності здійснюється їх модифікація, або вилучення із БД системи. Інформація вноситься в систему при надавленні оригіналів документів, які залишаються в фітнес-клубі.

Операції, що організовують діяльність із даними про клієнтів фітнес-клубу такі кроки:

- 1) Операція «Додавання даних про клієнта» включає:
 - додавання інформації про клієнта у БД (загальна інформація);
 - додавання інформації про послуги, які отримує клієнт (клієнту вказується місце та час проведення заняття чи надання послуги);
 - додавання інформації про здійснену оплату клієнтом за отримані послуги.
- 2) Операція «Модифікація даних про клієнта».
- 3) Операція «Вилучення даних про клієнта».

Вище перелічені операції виконує менеджер для роботи з клієнтами. Перелічені операції виконуються наступним чином: заповнюється форма реєстрації клієнта та відсилаються дані на сервер, де відбувається їх обробка. Після обробки даних відкривається особиста сторінка клієнта, де менеджер реєструє послуги, які обирає клієнт. У випадку необхідності здійснюється їх модифікація, або вилучення з БД системи. Інформація вноситься в систему при додаванні оригіналів документів.

Операції, що організовують діяльність із даними організаційної структури фітнес-клубу:

- 1) Операція «Додавання нових даних» включає:
 - додавання інформації про структурні відділи організації;
 - додавання інформації про кабінети відділів організації;
 - додавання інформації про наявний інвентар у користуванні клубу;
 - додавання інформації про систем знижок фітнес-клубу;
 - додавання інформації про наявні послуги, що надає клуб;
- 2) Операція «Модифікація даних»;
- 3) Операція «Вилучення даних».

Вище перелічені операції виконує управляючий. Перелічені операції виконуються наступним чином: заповнюється форма реєстрації клієнта та відсилаються дані на сервер, де відбувається їх обробка. Після обробки дана

інформація стає доступною для користування іншим користувачам системи. У випадку необхідності здійснюється їх модифікація, або вилучення із БД системи. Інформація вноситься в систему на основі затверджених наказів даної організації.

Операції, що організовують діяльність із даними щодо оплати клієнтами отриманих послуг:

- 1) Операція «Нарахування оплати».
- 2) Операція «Оплата рахунку».

Робота із вище перелічених операцій виконує бухгалтер. Перелічені операції виконуються наступним чином: клієнт обирає послугу, де вказується рахунок для сплати. Автоматично створюється запис із зазначеною сумою та друкується ордер на сплату нарахованої суми. Інформацію про оплату вноситься бухгалтер на основі звітів, які надає клієнт у вигляді чеків про оплату.

Операції, що організовують обміну повідомленнями між клієнтами та дієтологами, тренерами:

- 1) Операція «Надіслати повідомлення».
- 2) Операція «Зберегти чорновий варіант повідомлення».
- 3) Операція «Видалити повідомлення».

Даний вид операцій можуть виконувати всі користувачі, які є зареєстровані у системі.

Операції, що дають можливість створювати відкриті повідомлення (коментарі) видимі для всі користувачів системи створених до статей маркетологом фітнес-клубу:

- 1) Операція «Створити коментар».
- 2) Операція «Опублікувати коментар».
- 3) Операція «Модифікувати коментар».
- 4) Операція «Видалити коментар».

Даний вид операцій можуть виконувати всі користувачі, які є зареєстровані у системі.

3.2. Призначення та умови застосування програмної системи

Програмне забезпечення призначене для використання у сферах щодо надання послуг, зокрема фітнес-клуби, футбольні клуби, секції тощо. Програмний продукт використовується для автоматизації дій працівників організації, забезпечує оперативність роботи та дозволяє створювати звітність їх діяльності у відповідності до розроблених вимог щодо даних, які розміщені у БД.

Для роботи у розробленому web -додатку для клієнтів та працівників фітнес-клубу потреби у додаткових навиках немає. Для організації роботи у системі слід вміти працювати із ОС Windows чи будь-якою іншою ОС та браузером. Для виконання своїх посадових обов'язків працівники мають бути ознайомлені із своїми посадовими інструкціями.

Для працівників, які вдосконалюватимуть роботу програми та розроблятимуть нові функціональні можливості повинні мати знання у роботі із мовою програмування C#, вміння розробляти SQL запити для доступу до БД (отримання даних), практичні навички у роботі із платформою NET. Крім вище перерахованих основних теоретичних знань розробник повинен вміти працювати із програмним продуктом Visual Studio 2012, вміти налаштовувати сервер IIS 7.5.

Завантаження програми для організації роботи користувачами системи (клієнти та працівники фітнес-клубу) відбувається наступним чином: відкриття браузера Google Chrome (версія 35.0) та у адресному рядку вводиться адрес сайту фітнес-клубу. Для не ідентифікованого користувача системи на сайті відображається загально доступна інформація. Для отримання можливості працювати зі всіма реалізованими функціями, що визначені для ролі користувача – має відбутися вхід у систему, що передбачає визначення допустимих операцій. Дані, які затребує користувач передаються у відображення (html сторінка) автоматично із метода, який визиває користувач. Обробка даних із БД здійснюється з допомогою технології LINQ.

Організація роботи із системою для адміністратора програми починається із налагодження сервера та публікації скомпільованого проекту. Перш ніж розглянути процес інсталяції слід розглянути основні складові частини публікаційного сайту, з чого безпосередньо складатиметься працюючий веб-додаток.

Статичні ресурси в даному випадку – це файли стилів `css`, скрипти `JavaScript`, файли зображень. У стандартній структурі проекту `ASP.NET MVC` ці файли розташовуються в каталогах `Content`, `Scripts` і `Images` відповідно. Ці файли компілюються в динамічні збірки, тому є статичними ресурсами і переносяться на сайт при публікації як є.

Збірки додатку представляють по суті ключову частину додатку - саме в них компілюються класи та контролери. При компіляції збірки додатку потрапляють в проект в папку `bin`, де їх власне можна знайти. Так, якщо додаток називається `BusinessLogic`, то після компіляції в папці `bin` виявиться файл `BusinessLogic.dll`, який і представлятиме ключовий файл програми.

При розгортанні додатку важливо, щоб він мав доступ до бібліотек фреймворка `ASP.NET MVC`. Для цього необхідно або встановити на сервері даний фреймворк, або включити всі потрібні бібліотеки фреймворка в пакет програми. В останньому випадку всі потрібні нам бібліотеки також опиняються в папці `bin`. Якщо при розробці програмного продукту використовувалось програмне забезпечення `Visual Studi 2012` (у тому числі в `Express`-версії) копіювання бібліотек в папку `bin` відбувається автоматично). Тому всі динамічні бібліотеки фреймворка вже зможете побачити в папці `bin` після компіляції.

Процес публікації конструйованої аплікації на `web` -сервері `IIS` передбачає виконання наступних дій [44]:

1. Розміщення системи відбувається на вузлі по замовчуванню (`Default Web Site`).

2. На вузлі " `Default Web Site` " із контекстного меню вибираємо пункт "Додати додаток" та у вікні, що з'явиться встановлюємо Псевдонім та

Фізичний шлях.

3. Перейшовши до додатка в Visual Studio. У контекстному меню проекту вибираємо Publish.

4. У вікні, що появилось вибираємо профіль, якщо не визначено, то створюємо новий натиснувши на посилання New... і вибираємо назву.

5. Після створення профілю натиснемо на Next і переходимо до наступного етапу - Connection. На цьому етапі для опції Publish Method виберемо File System. Для опції Target Location визначимо фізичний шлях до каталогу нашого сайту (наприклад: C:\inetpub\wwwroot\SportClub). А для поля Destination URL вказуємо url, за яким буде доступно додаток (наприклад: <http://localhost/SportClub>). Після встановлення всіх властивостей тиснемо на кнопку Publish.

6. Якщо ви перенесли на IIS додаток з бази даних, що використовує движок LocalDB, то при зверненні до додатка може виникати помилка "provider: SQL Network Interfaces, error: 50 - Local Database Runtime error occurred. Cannot create an automatic instance". У цьому випадку нам потрібно налаштувати конфігурацію веб-сервера. Визначити пул сайту. В консолі IIS відкрити налаштування сайту. І в налаштуваннях знайти пул.

7. Перейти в каталог C:\Windows\System32\inetsrv\config. Відкрити файл applicationHost.config і знайти в ньому рядок пулу додатків, під яким працює сайт:

```
<add name="DefaultAppPool" managedRuntimeVersion="v4.0" />
```

Замінити цей рядок такими рядками:

```
<add name="DefaultAppPool" managedRuntimeVersion="v4.0" >
    <processModel
        identityType="ApplicationPoolIdentity"
        loadUserProfile="true" setProfileEnvironment="true" />
</add>
```

8. Зберегти файл. Перезапустити пул додатків для цього в консолі IIS виберемо праворуч Пули додатків. Знайти в центрі потрібний пул і зліва натиснемо на Перезапуск. Все готово.

3.3. Технологічний процес збору та обробки даних на периферійних пристроях при централізованій обробці даних

Конструйована система розроблена на основі клієнт-серверної архітектури та реалізована у вигляді «тонкого» клієнта. Враховуючи сказане, процес обробки даних є централізованим та реалізований на стороні сервера. На стороні клієнта винесена бізнес-логіка, що відповідає за валідацію даних, тобто перевірка правильності вводу інформації користувачем. Процес валідації даних відбувається за допомогою мови JavaScript та реалізується за допомогою javascript-фреймворка jQuery. Відправка даних на сервер відбувається за рахунок http протоколу прикладного рівня. У даному процесі передачі даних використовуються методи GET та POST.

Http протокол, що використовує методом GET запрошує вміст вказаного ресурсу, що визначається url адресою. Запитаний ресурс може приймати параметри (наприклад, пошукова система може приймати, як параметр ідентифікатор об'єкта). Запити типу GET є ідемпотентними, тобто багатократне повторення одного і того ж запиту GET приводить до однакових результатів. Формат запиту від клієнта до сервера формується згідно стандарту HTTP та будується наступним чином: <http://localhost/Controller/Action/parameters>, де вказується контролер і дія, що реалізуватимуть обробку на сервері даних переданих через об'єкт «parameters».

Використання методом POST у http протоколах здійснюється для передачі даних на сервер із відповідних HTML-форм. Даний метод не є ідемпотентним, отже, при багаточисельному повторенні тих самих запитів POST можуть надходити різні результати, зокрема, при відправленні певного коментаря може з'являтися одна копія цього коментаря.

Бізнес-логіка обробки даних реалізована на сервері. Процес обробки даних відбувається при виклику методу користувачем. Отримані дані, що передані протоколом прикладного рівня від користувача системи, сервер

використовує для формування контенту HTML сторінки, яку отримує кінцевий споживач. Отримання даних від БД здійснюється з використання технології LINQ. Технологія LINQ для SQL (колишній Dlinq) перетворює LINQ-вирази в SQL-запити до бази даних, використовує можливості компілятора для побудови дерева виразів, ґрунтуючись на контексті програми, а не створюючи делегати функцій. Отримавши дерево вирази, що описує запит, спеціалізований провайдер бази даних може його проаналізувати і перетворити в запит на відповідному мовою для бази даних, наприклад Microsoft SQL Server, Jet (яка використовується в Microsoft Access) або будь-який інший. Таким чином організовується обмін інформацією між БД, сервером та клієнтом системи.

Висновки за розділом 3

Проблема складності є головною проблемою, яку доводиться вирішувати при створенні систем будь-якої природи, в тому числі й ЕОС. Жоден розробник не в змозі вийти за межі людських можливостей і зрозуміти всю систему в цілому. Однак, можна знайти вихід з цієї ситуації, якщо створити (побудувати) складну систему, що містить кілька частин, а останні, у свою чергу, складаються з частинок ще меншого розміру і т.п., поки процес не набуде зворотного шляху, зокрема: побудова невеликих частин з наявного матеріалу. Цей процес (підхід) має назву ієрархічна декомпозиція. По відношенню до проектування програмної системи це означає, що складну систему маємо розділити на невеликі підсистеми, а тоді, кожну новостворену підсистему у подальшому розробляти незалежно від інших. Це дозволяє нам при розробці підсистеми обліку послуг спортивного клубу тримати в розумі інформацію тільки про неї, а не про всіх інших частинах системи. «Правильна система» по відношенню до декомпозиції має відповідати наступним вимогам: насамперед, мінімальними має бути кількість зв'язків між окремими підсистемами і навпаки, максимальною має бути зв'язність окремих частин в середині кожної підсистеми.

Успіх будь-якого проекту з розробки ПЗ визначається трьома складовими: методологія проектування, система позначень (мова, нотація) та інструмент.

У третьому розділі ми дали характеристику двом підходам програмної інженерії щодо розробки програмного забезпечення ЕІС, де зазначили принципові відмінності обумовлені різними способами декомпозиції систем. Перший підхід має назву – функціонально – модульний (структурний). Суть його полягає у застосуванні принципу функціональної декомпозиції, коли структура системи подається в термінах ієрархії її функцій та передачі інформації між певними функціональними елементами. Об'єктно – орієнтований підхід містить об'єктну декомпозицію. При даному підході, в термінах об'єктів та зв'язків між ними описується структура системи. Що стосується поведінки системи зазначеного підходу – опис відбувається в термінах обміну повідомленнями між об'єктами.

При розробці підсистеми обліку послуг будемо спиратися на структурний підхід.

Переваги структурного підходу в тому, що розроблена нами підсистема, має здатність зберігати цілісне уявлення про неї, крім того, всі складові компоненти в ній взаємопов'язані.

Інструментальна база структурного підходу представлена – широкою номенклатурою CASE-засобів, що підтримують різні методології і види проектної діяльності.

Для роботи у розробленому web -додатку для клієнтів та працівників фітнес-клубу потреби у додаткових навиках немає. Для організації роботи у системі слід вміти працювати із ОС Windows чи будь-якою іншою ОС та браузером. Для виконання своїх посадових обов'язків працівники мають бути ознайомлені із своїми посадовими інструкціями.

Для працівників, які вдосконалюватимуть роботу програми та розроблятимуть нові функціональні можливості повинні мати знання у роботі із мовою програмування C#, вміння розробляти SQL запити для доступу до БД (отримання даних), практичні навики у роботі із платформою NET. Крім вище

перерахованих основних теоретичних знань розробник повинен вміти працювати із програмним продуктом Visual Studio 2012, вміти налаштовувати сервер ІІS 7.5.

ВИСНОВКИ

Застосування та розвиток інформаційних продуктів є новим основним засобом організації послуг клієнтам спортивного клубу, а тому потребують узагальнень та розробки практичних рекомендацій на всіх рівнях.

Розробка веб – додатків, на сьогодні, є досить актуальним, оскільки зазначена технологія розвивається швидкими темпами, має суттєві переваги щодо економії часу, ресурсів, не обмежує користувача у виборі місцезнаходження для застосування своїх можливостей, можна працювати з будь – яких девайсів не залежно від операційної системи чи її технічних характеристик.

В результаті проведеного дослідження був розроблений веб – додаток для управління процесом діяльності фітнес – клубу.

З метою реалізації даної роботи був обраний в якості інструменту програмний продукт BPwin фірми Platinum technology/Logic Works.

Під час моделювання підсистеми використані нотації SADT (Strucured Analysis and Design Technique) і DFD (Data Flow Diagrams), завдяки яким найбільш адекватно відображено логіку предметної області та сприйняття для проведення аналізу бізнес – процесів.

Відповідно до стандарту IDEF1X була побудована інформаційна модель предметної області, фізична модель даних, яка містить інформацію щодо конкретних об'єктів баз даних (таблиця, індекси, поля, процедури та ін.). Що стосується фізичної моделі, вона орієнтована на СУБД Microsoft SQL Server.

Програмне забезпечення розроблено з дотриманням основним принципам створення інтерфейсу, в роботі подана інструкція користувача, а також умови застосування програмної системи

Програмне забезпечення може бути використане при наданні послуг різноманітних спортивних секцій (клубів), зокрема фітнес – клубами, футбольними клубами, волейбольними та ін. Розглянутий програмний продукт застосовується з метою покращення процесу управління надання послуг, автоматизації дій працівників, створення оперативного реагування на бажання

клієнтів. У зручний спосіб відбувається формування звітності щодо діяльності фітнес – клубу відповідно до розроблених вимог та даних, які розміщені у базі даних.

Програмне забезпечення написане за допомогою застосування мови програмування С# із використанням .NET, що дає можливість суттєво покращити ефективність діяльності фітнес – клубу.

СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ

1. Адам Фримен. ASP.NET MVC 4 с примерами на C# 5.0 для профессионалов. [Текст] / Адам Фримен – М.: Вильямс. - 2013 – 688 с.
2. Адам Фримен. LINQ. Язык интегрированных запросов в C# 2010 для профессионалов. [Текст] Адам Фримен, Джозеф С. Раттц. – М.: Вильямс. - 2013 – 656 с.
3. Абдрахманова Г. И. Использование ИКТ в экономике: тенденции развития в 2009-10 годах [Текст] / Г. И. Абдрахманова, Г. Г. Ковалева // Вопросы статистики. - 2018. - № 11. - С. 53-57.
4. Балабанова Л. В. Информационное обеспечение обоснование управленческих решений в условиях маркетинговой ориентации предприятия : [Монография] / Л. В. Балабанова, Т. И. Алачева. – Донецк: ДонГУЭТ им. М. Туган-Барановского. - 2003. – 143 с.
5. Барсегян А. А. Методы и модели анализа данных: OLAP и Data Mining. [Текст] / А. А. Барсегян, М. С. Куприянов, В. В. Степаненко // – СПб.: БХВ-Петербург. - 2004. – 336 с.
6. Біляченко О. Л. Особливості відображення програмного забезпечення на рахунках бухгалтерського обліку [Текст] / О. Л. Біляченко // Вісник Житомирського державного технологічного університету. - 2010. - № 4. - (54). - С. 22-26.
7. Бердтис А. Структуры данных [Текст] / А. Бердтис - М.: Статистика . – 2018. - 408 с.
6. Бормашов Д. А. “Кластерный анализ текстов”: Дипломная работа [Электронный ресурс] / Д. А. Бормашов. Режим доступа: <http://inf.tsu.ru/library/DiplomaWorks/CompScience/2006/bormashov/diplom.pdf>.
8. Босак І. П. Інформаційне забезпечення управління підприємством: економічний аспект / І. П. Босак, Є. М. Палига // Регіональна економіка. – 2017. – № 4. – С. 193–195.

9. Буров Є. В. Комп'ютерні мережі: підручник. [Текст] / Євген Вікторович Буров. — Львів: «Магнолія 2006», 2010. — 262 с. ISBN 966-8340-69-8
10. Бхуптани М. RFID - технологии на службе вашего бизнеса [Текст] / М. Бхуптани, Ш. Морадпур ; пер.с англ. - М. : Альпина Бизнес Букс. - 2017. - 281 с.
11. Вікіпедія [Електронний ресурс]: вікіпідія <https://uk.wikipedia.org/wiki> - веб - [додаток](#)
12. Вовчак І.С. Інформаційні системи та комп'ютерні технології в менеджменті: Навч. посібник. [Текст] //І. С. Вовчак. – Тернопіль: Карт-бланш. - 2001. – 354с.
13. Гафіяк А.М. Автоматизована система формування проектної звітності / А.М. Гафіяк, Ю.В. Кириченко // Тези 69-ї наукової конференції професорів, викладачів, наукових працівників, аспірантів та студентів університету. Т. 2. (Полтава, 19 квітня–19 травня 2017 р.). – Полтава : ПолтНТУ. - 2017. – С. 167–168.
14. Гафіяк А.М. Автоматизована інформаційна система управління організаційно-технологічними процесами на підприємстві / А.М. Гафіяк, Р.М. Костирко // Тези 68-ї наукової конференції професорів, викладачів, наукових працівників, аспірантів та студентів університету. Т. 2. (Полтава, 15 квітня – 15 травня 2016 р.) – Полтава : ПолтНТУ. - 2016. – С. 167–168.
15. Гуцайлюк З. В. Інформаційна система бухгалтерського обліку в управлінні промисловим підприємством: рівні формування та напрямки використання / З. В. Гуцайлюк, Р. І. Мачуга // Вісн. ЖДТУ. – № 3. – 2010. – С. 70–74.
16. Закон України «Про основні засади розвитку інформаційного суспільства в Україні на 2007-2015 роки» від 9 січня 2007 року, N 537-У [Електронний ресурс]. - Режим доступу: <http://zakon1.rada.gov.ua/cgi-bin/laws/main.cgi?nreg=537-16>.
17. Інформаційне суспільство. Дефініції... / [В. М. Брижко, А. А. Орехов, В. С. Цимбалюк та ін.] ; під ред. Р. А. Калюжний, М. Я. Швець. – К. : Інтеграл. - 2002. – 220 с.

18. Інформаційні технології та моделювання бізнес-процесів : навч. посіб. [Текст] / О. М. Томашевський, Г. Г. Цигелик, М. Б. Вітер, В. І. Дудук. – К. : Центр учбової літератури, 2012. – 296 с.
19. Калянов Г. Н. CASE-технологии. Консалтинг в автоматизации бизнес процессов. –3-е изд [Текст] / Калянов Г. Н. Горячая линия — М.: Телеком, 2012. – 320с.
20. Клімушин П. С. Електронне урядування в інформаційному суспільстві : [монографія] / П. С. Клімушин, А. О. Серенок. – Х. : Вид-во ХарПІ НАДУ “Магістр”. - 2010. – 312 с.
21. Клімушин П. С. Програмні комплекси автоматизації управлінських технологій підприємств / П. С. Клімушин // Державне будівництво : електр. наук. фах. вид. – Х. : Вид-во ХарПІ НАДУ “Магістр”. - 2007. – № 1. – Ч. 2. – Режим доступу : <http://www.nbu.gov.ua>, <http://www.nhlu.gov.ua>, <http://www.kbuara.kharkov.ua>
22. Клімушин П. С. Інформаційні системи та технології в економіці : навч. посіб. [Текст] / П. С. Клімушин, О. В. Орлов, А. О. Серенок. – Х. : Вид-во ХарПІ НАДУ “Магістр”. - 2011. – 448 с.
23. Клименко О. В. Інформаційні системи і технології в обліку : навч. посіб. [Текст] / Клименко О. В. – К. : ЦУЛ. - 2008. – 320 с.
24. Клушин Ю. С. Підвищення швидкості роботи веб – додатків//Ю.С. Клушин, Ю.Б. Захарчин. Комп’ютерні системи і мережі. [Електронний ресурс]: Режим доступу : <https://science.lpnu.ua/sites/default/files/journal-paper/2021/may/23469/csnv2n12020-37-47.pdf>
25. Комп’ютерні мережі: [навчальний посібник] [Текст] / А. Г. Микитишин, М. М. Митник, П. Д. Стухляк, В. В. Пасічник. — Львів: «Магнолія 2006». - 2013. — 256 с.
26. Корзаченко О. В. Технології управління бізнес-процесами як інструмент підвищення ефективності бізнесу / О. В. Корзаченко // Моделювання та інформ. системи в економіці [Електронний ресурс] : зб. наук. праць / М-во освіти і

науки України, ДВНЗ “Київ. нац. екон. ун-т ім. В. Гетьмана” ; відп. ред. В. К. Галіцин. – 2010. – Вип. 82. – С. 228–239.

27. Коряк С. Ф. Комп’ютерні системи обробки та передачі фінансової інформації [Текст] / С. Ф. Коряк, Л. Д. Самофалов ; за ред. Л. Д. Самофалова. – Х. : СМІТ. - 2014. – 290 с.

29. Ландэ Д. В. Интернетика: Навигация в сложных сетях: модели и алгоритмы / Д.В. Ланлэ, А.А. Снарский, И.В. Безсуднов. — М.:Либроком (Editorial URSS), 2009. — 264 с.

29. Лодон Дж. Управление информационными системами [Текст] / Дж. Лодон, К. Лодон ; пер. с англ. под ред. Д. Р. Третнева. – 7-е изд. – СПб. : Питер. - 2005. – 912 с.

30. Мацанюк Б.М. Використання ІТ – технологій для оптимізації управління бізнес – процесами організацій та підприємств /Б.М. Мацанюк //Математичні методи, моделі та інформаційні технології в управлінні підприємством . Матеріали ІV студентської вузівської наукової конференції. – Вінниця : ВТЕІ КНТЕУ, 2019. – С. 180 - 182

31. Методичні вказівки для проведення практичних (лабораторних) занять з дисципліни «Інформаційні системи в управлінні» /укладачі І.В. Данилюк, Л.В. Дума, А.Я. Мушак. – Тернопіль. - 2022. – 80 с.

32. Оксанич А. П. Інформаційні системи і технології маркетингу : навч. посіб. [Текст] / А. П. Оксанич, В. Р. Петренко, О. П. Костенко. – Х. : СМІТ. - 2015. – 240 с.

33. Писаревська Т. А. Інформаційні системи обліку та аудиту : навч. посіб. [Текст] / Писаревська Т. А. – К. : КНЕУ. - 2014. – 369 с.

34. Кен Хендерсон Професійне керівництво з SQL Server: структура та реалізація [Текст] /Кен Хендерсон . — М.: Издательский дом «Вильямс». - 2006. — С. 1056.

35. Яценко Р. М. Інформаційні системи в логістиці : навч. посіб. [Текст] / Р. М. Яценко, І. В. Ніколаєв – Х. : Вид-во ХНЕУ. - 2012. – 232 с.

36. James E. Hunton. Blending Information and Communication Technology with Accounting Research/ James E. Hunton // Accounting Horizons. - Vol. 16. - 2002.

37. Xie Shi-fen. Accounting information: Concepts, characteristics and significance of Hunan Finance and Accounting/ Xie Shi-fen. - 1999 - С.11.

Інформаційні ресурси

38. Кельдер Т. Л. Системи обробки економічної інформації / Т. Л. Кельдер [Електронний ресурс]. - Режим доступу: <http://www.zsu.zp.ua/lab/mathdep/тте/ГУ/soei/soei11.htm.1>

39. Клієнт-сервер архітектура [Електронний ресурс]. – Режим доступу: <http://buklib.net/books/23148/>

40. Створення веб – додатку для автоматизованої роботи кадрового агентства. Пояснювальна записка до дипломного проекту (роботи). Магістр,

Маньковський А.В. Житомирський державний університет імені Івана Франка, Фізико-математичний факультет, Кафедра прикладної математики та інформатики[Електронний ресурс]. – Режим доступу: <http://eprints.zu.edu.ua/18020/1/%D0%9C%D0%B0%D0%BD%D1%8C%D0%BA%D0%BE%D0%B2%D1%81%D1%8C%D0%BA%D0%B8%D0%B9.pdf>

41. Основні методи тестування безпеки веб-додатків. [Електронний ресурс]. – Режим доступу: <http://uastudent.com/osnovni-metody-testuvannja-bezpeky-veb-dodatkiv/>

42. Створення веб – додатку для автоматизованої роботи кадрового агентства [Електронний ресурс]. – Режим доступу: [:file:///D:/%D0%BA%D0%BE%D0%BF%D1%96%D1%8F](file:///D:/%D0%BA%D0%BE%D0%BF%D1%96%D1%8F)

43. Проект Web expert. [Електронний ресурс]. — Режим доступу: [https://webexpert.com.ua/ecommercein\\$ukraine\\$2016](https://webexpert.com.ua/ecommercein$ukraine$2016)

44. Програмування на C# [Електронний ресурс]. – Режим доступу: http://life-prog.ru/ukr/view_cat.php?cat=10

45. Українські підручники онлайн. Система управління базою даних SQL Server 7.0 [Електронний ресурс]. – Режим доступу:

http://pidruchniki.ws/15100827/bankivska_sprava/sistema_upravlinnya_bazoyu_dani_h_sql_server

46. Хмарні технології [Електронний ресурс]. – Режим доступу : <http://j.parus.ua/ua/358>. - Хмарні технології

47. Чураков Михаил. Муравьиные алгоритмы [Електронний ресурс] / Михаил Чураков, Андрей Якушев. // Режим доступу: <http://rain.ifmo.ru/cat/data/theory/unordered/ant-algo-2006/article.pdf>.

48. Шапаренко О.В. Вплив інформаційних технологій на бухгалтерський облік /О.В. Шапаренко [Електронний ресурс] / /О.В. Шапаренко – Режим доступу: <http://zt.knteu.kiev.ua>

49. WEB – додатки [Електронний ресурс]. – Режим доступу : <https://stud.com.ua/97610/informatika/dodatki>- WEB – додатки

