

Міністерство освіти і науки України
Харківський національний університет радіоелектроніки

Факультет Комп'ютерних наук
(повна назва)

Кафедра Системотехніки
(повна назва)

КВАЛІФІКАЦІЙНА РОБОТА Пояснювальна записка

рівень вищої освіти другий (магістерський)

(позначення документа)
Розробка та аналіз інформаційного забезпечення систем надання послуг
(тема)

Виконав: здобувач групи ІТПМ-21-2
Клепиков Р. В.
(прізвище, ініціали)


Спеціальність 122 Комп'ютерні науки
(код і повна назва спеціальності)

Освітня програма ОПП Інформаційні технології проектування
(повна назва освітньої програми)

Керівник проф. Гребеннік І.В.
(посада, прізвище, ініціали)

Допускається до захисту

Зав. кафедри


(підпис)

Гребеннік І. В.
(прізвище, ініціали)

Я як студент ХНУРЕ розумію і підтримую політику закладу із академічної доброчесності. Я не надавав і не одержував недозволену допомогу під час підготовки кваліфікаційної роботи. Використання ідей, результатів і текстів інших авторів мають посилання на відповідне джерело.

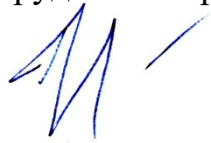
(дата, підпис, прізвище ін. студента/-ки)

Кваліфікаційна робота не містить відомостей заборонених до відкритого опублікування.

Кваліфікаційна робота виконана у відповідності до стандартів, що діють в Україні.

Попередній захист проведено 19 грудня 2022 р.

Керівник кваліфікаційної роботи



проф . Гребеннік І.В

Харківський національний університет радіоелектроніки

Факультет _____ Комп'ютерних наук _____

Кафедра _____ Системотехніки _____

Рівень вищої освіти _____ другий (магістерський) _____

Спеціальність _____ 122 «Комп'ютерні науки» _____
(код і повна назва)Освітня програма _____ ОПІ Інформаційні технології проектування _____
(повна назва)

ЗАТВЕРДЖУЮ:

Зав. кафедри _____
(підпис)

« ____ » _____ 20 __ 22 р.

ЗАВДАННЯ
НА КВАЛІФІКАЦІЙНУ РОБОТУстудентові _____ Клепікову Ростиславу Володимировичу _____
(прізвище, ім'я, по батькові)

1. Тема роботи _____ Розробка та аналіз інформаційного забезпечення систем надання послуг _____
затверджена наказом університету від 21 листопада 2022р. №1504Ст _____
2. Термін подання студентом роботи до екзаменаційної комісії 20 __ 12 _____ 2022 р.
3. Вихідні дані до роботи Розробити рекомендації що до розробки інформаційного забезпечення систем надання послуг. Використовуючи вказанні рекомендації створити інформаційне забезпечення для системи надання професійних навчальних послуг із самооборони. Серверна частина являє собою реалізацію бази даних розроблена для платформи MySQL. Клієнтська частина має такі бізнес-функції, як реєстрація у системі, автоматизація, можливість переглянути список секцій, та список викладачів, придбати абонемент
4. Перелік питань, що потрібно опрацювати в роботі Провести аналіз предметної області, виділити основні недоліки існуючих інформаційних систем надання послуг, сформулювати загальні рекомендації що до створення інформаційних систем надання послуг, на практиці реалізувати створенні рекомендації на прикладі системи надання професійних навчальних послуг із самооборони. В ході реалізації рекомендацій необхідно провести аналіз бізнес-процесів та виділити які потребують автоматизації; сформулювати та оформити вимоги до інформаційної системи: створити функціональну модель використовуючи стандарт IDEF0, провести функціональне моделювання, визначити та уточнити вимоги до інформаційної системи: провести логічне та фізично моделювання БД, обґрунтувати вибір платформи СУБД для реалізації БД, провести UML-моделювання проектованої клієнтської частини інформаційної системи, провести аналіз і виділити основні бізнес-процеси з поділом бізнес-функцій, реалізувати фізичну модель БД.

5. Перелік графічного матеріалу із зазначенням креслеників, схем, плакатів, комп'ютерних ілюстрацій (п.5 включається до завдання за рішенням випускової кафедри) Контекстна діаграма IDEF0, Декомпована діаграма IDEF0, Декомпозиція другого рівня IDEF0, Схематичне представлення БД, Use Case діаграма, діаграма послідовності, діаграма класів.

6. Консультанти розділів роботи

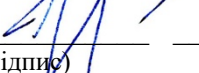
Найменування розділу	Консультант (посада, прізвище, ім'я, по батькові)	Позначка консультанта про виконання розділу	
		підпис	дата
Основна частина	проф. Гребеннік І.В.		

Календарний план

№	Назва етапів роботи	Терміни виконання етапів роботи	Примітка
1	Отримання завдання атестаційної роботи	<u>21.11.2022</u>	Виконано
2	Аналіз завдання, літератури та аналогів з теми атестаційної роботи.	<u>22.11.2022</u>	Виконано
3	Створення рекомендацій що до розробки інформаційного забезпечення систем надання послуг.	<u>24.11.2022</u>	Виконано
4	Імплементация розроблених рекомендацій на прикладі системи надання послуг	<u>26.11.2022</u>	Виконано
5	Вибір середовища розробки програми.	<u>28.11.2022</u>	Виконано
6	Розробка програми	<u>30.11.2022</u>	Виконано
7	Тестування програми	<u>02.12.2022</u>	Виконано
8	Розробка «посібника користувача»	<u>04.12.2022</u>	Виконано
9	Оформлення пояснювальної записки та програмної документації	<u>08.12.2022</u>	Виконано
10	Оформлення графічної частини та презентаційних матеріалів комп'ютерного захисту.	<u>10.12.2022</u>	Виконано
11	Представлення на рецензування	<u>14.12.2022</u>	Виконано
	Представлення атестаційної роботи в ЕК	<u>15.12.2022</u>	Виконано

Дата видачі завдання 21.11. _____ 2022_ р.

Студент _____
(підпис)

Керівник роботи _____
(підпис)  І.Гребеннік
(посада, прізвище, ініціали)

РЕФЕРАТ

Кваліфікаційна робота : 48 с., 19 рис., 16 джерел.

СЕРВІС-ОРІЄНТОВАНА АРХІТЕКТУРА, СИСТЕМА ОБЛІКУ ЗАМОВЛЕНЬ, СЕРВІС, РЕЄСТР СЕРВІСІВ, РОЗРОБКА, БАЗА ДАНИХ, JAVA, SPRING, SQL, JAVASCRIPT, HTML, CSS, SPRING, MYSQL. СИСТЕМИ НАДАННЯ ПОСЛУГ, СФЕРА ПОСЛУГ.

Об'єкт розробки – інформаційне забезпечення систем надання послуг

Предметом дослідження є інформаційні технології створення систем надання послуг різного призначення.

Мета роботи – підвищення ефективності систем надання послуг шляхом розробки та аналізу інформаційного забезпечення.

Метод розробки – гнучка методологія розробки (Agile), сервіс орієнтований підхід проектування програмного забезпечення.

Результатом кваліфікаційної роботи є: рекомендації щодо розробки інформаційного забезпечення систем надання послуг, реалізація на прикладі системи інформаційного забезпечення спортивного комплексу.

Розроблювана система дозволить значно спростити процес адміністрування абонементів в спортивному комплексі.

Сфера застосування – представлені рекомендації щодо розробки та запровадження систем надання послуг можуть бути використані при розробці будь-якої системи надання послуг.

ABSTRACT

Qualification work: 48 pages, 19 figures, 17 sources.

SERVICE-ORIENTED ARCHITECTURE, ORDER ACCOUNTING SYSTEM, SERVICE, SERVICE REGISTRY, DEVELOPMENT, DATABASE, JAVA, SPRING, SQL, JAVASCRIPT, HTML, CSS, SPRING, MYSQL. SERVICE PROVISION SYSTEMS, FIELD OF SERVICES.

The object of development is information support of service provision systems

The subject of research is information technologies for the creation of systems for the provision of services for various purposes.

The purpose of the work is to improve the efficiency of service delivery systems through the development and analysis of information support.

The development method is a flexible development methodology (Agile), a service-oriented approach to software design.

The result of the qualification work is: recommendations for the development of information support systems for providing services, implementation on the example of the information support system of a sports complex.

The developed system will significantly simplify the process of administration of season tickets in the sports complex.

Scope - the presented recommendations for the development and implementation of service delivery systems can be used in the development of any service delivery system.

ЗМІСТ

ПЕРЕЛІК УМОВНИХ ПОЗНАЧЕНЬ, СИМВОЛІВ, ОДИНИЦЬ, СКОРОЧЕНЬ І ТЕРМІНІВ	8
ВСТУП	9
1 АНАЛІЗ ПРЕДМЕТНОЇ ГАЛУЗІ	10
1.1 Аналіз предметної галузі	10
1.2 Аналіз існуючих систем аналогів	14
2 ПОСТАНОВКА ЗАДАЧІ ДОСЛІДЖЕННЯ	19
3. ОГЛЯД МЕТОДІВ ТА ТЕХНОЛОГІЙ ЩО ВИКОРИСТОВУЮТЬСЯ	23
3.1 Обґрунтування вибору програмних засобів для розробки.....	23
3.2 Модель бази даних	32
3.3 Розробка інтерфейсу клієнтської частини	34
4.РОЗРОБКА ІНФОРМАЦІЙНОЇ ТЕХНОЛОГІЇ.....	39
4.1 Розробка системних вимог до системи надання послуг.....	39
4.2 Опис архітектури розроблюваної системи	39
4.2 Розробка функціональних вимог	41
4.3 Розробка діаграми варіантів використання	44
ВИСНОВКИ.....	46
ПЕРЕЛІК ДЖЕРЕЛ ПОСИЛАННЯ.....	48
ДОДАТОК А Графічний матеріал атестаційної роботи.....	50
ДОДАТОК Б Текст програми	56
ДОДАТОК В Посібник користувача	65
ДОДАТОК Г Відомість атестаційної роботи	794

ПЕРЕЛІК УМОВНИХ ПОЗНАЧЕНЬ, СИМВОЛІВ, ОДИНИЦЬ, СКОРОЧЕНЬ І ТЕРМІНІВ

JDK – java development kit, безкоштовний розповсюджуваний Oracle комплект розробника застосунків на мові Java, який включає до себе компілятор Java стандартні бібліотеки класів Java, приклади, документацію, різноманітні утиліти і виконавчу систему Java (JRE). В склад JDK не входить інтегроване середовище розробки на Java (IDE), тому розробник, що використовує тільки JDK, повинен використовувати текстовий редактор і компілювати та виконувати свої програми через утиліти командного рядка.

JVM – java virtual machine, набір комп'ютерних програм та структур даних, що використовують модель віртуальної машини для виконання інших комп'ютерних програм чи скриптів.

CD – Continuous delivery, підхід у програмній інженерії, суть якого полягає в тому, що команди розробляють програмне забезпечення протягом коротких періодів часу, забезпечуючи надійний випуск версії у будь-який час. Його метою є створення, тестування та випуск програмного забезпечення швидше та частіше. Підхід допомагає зменшити вартість, час та ризик доставки змін, дозволяючи додаткові поповнення додатків у виробництві. Простий та повторюваний процес розгортання має важливе значення для безперервної доставки.

API – Application Programming Interface, набір визначень підпрограм, протоколів взаємодії та засобів для створення програмного забезпечення. Спрощено - це набір чітко визначених методів для взаємодії різних компонентів. API надає розробнику засоби для швидкої розробки програмного забезпечення.

Сфера послуг — частина економіки, яка включає в себе всі види комерційних і некомерційних послуг, зведена узагальнююча категорія, що включає виробництво різноманітних видів послуг, наданих підприємствами, організаціями, а також фізичними особами.

ВСТУП

Сфера послуг є однією з основних галузей громадського виробництва, покликана чуйно реагувати на потреби та попит населення.

Ринок послуг розглядається у єдності з товарним ринком як одна з його різновидів, що підпорядковується загальним законам ринкової економіки. При цьому ринок послуг має ряд особливостей: високий динамізм, територіальна сегментація та локальний характер, висока швидкість обороту капіталу внаслідок короткого виробничого циклу, переважання у виробництві послуг малих та середніх підприємств, індивідуальність і нестандартність послуг та технологій, що надаються багатьох галузях, невизначеність результату під час надання послуг.

Управління підприємством сфери послуг та прийняття необхідних управлінських рішень вимагає наявності великого обсягу інформації у цій сфері. Розвиток ринків послуг, посилення конкурентної боротьби за клієнтів надають новий, якісно інший, ніж раніше, сенс діяльності підприємств, вимагають їхньої орієнтації не тільки на громадські запити, а й, передусім, особисті потреби людини. В цій зв'язку виробництво послуг, покликане за своєю суттю обслуговувати клієнта, задовольняючи його нагальні особисті чи корпоративні потреби, стає домінантою для менеджменту сучасних організацій, його концепцій, методів та прийомів та визначає конкурентні стратегії, засновані на навичках і вміннях надавати високоякісний сервіс.

1 АНАЛІЗ ПРЕДМЕТНОЇ ГАЛУЗІ

1.1 Аналіз предметної галузі

Система надання послуги визначає, коли, де і як послуга надається клієнту. Ця підсистема охоплює як видимі елементи підсистеми сервісних операцій - будівлі, устаткування й персонал, а й впливає інших клієнтів. Розміщення підприємства сервісу, мабуть, важливіше у сфері послуг, ніж у виробництві. Воно пов'язане з купівельним попитом: а чи взагалі буде клієнт відвідувати офіс сервісної фірми? "Servuction" модель дає таку відповідь: місцезнаходження підприємства з надання послуг переважно визначається місцезнаходженням споживачів, а чи не будь-якими іншими чинниками. Там, де обслуговування потребує участі клієнта (у сфері соціально-культурного сервісу, туризму, розваг) розумна близькість до споживачів просто необхідна. Насамперед це стосується розміщення фронт- офісів підприємств та фірм. Бек-офіс цілком може бути винесений поза зонами перебування споживачів, тобто на околицю міста чи селища чи промисловий район. Зазвичай сервісні компанії завжди взаємодіяли зі своїми клієнтами безпосередньо. Однак для досягнення багатьох цілей, що стоять перед ними: від зниження вартості послуг і підвищення якості обслуговування клієнтів до підвищення зручності послуг для клієнтів, при наданні багатьох послуг, які не потребують фізичної присутності споживачів, - сьогодні йде активний пошук шляхів скорочення прямих контактів клієнта з сервісною організацією. У міру того як використовуються нові електронні технології та модернізовані фізичні потоки, що дозволяють перетворити послуги з високим ступенем взаємодії з клієнтом на послуги з низьким рівнем контакту, видимий компонент системи сервісних операцій сьогодні скорочується в багатьох галузях. У цьому сенсі система самообслуговування зазвичай зручніша для клієнтів, ніж система, заснована на безпосередньому контакті з обслуговуючим персоналом сервісної організації. Звісно ж, система самообслуговування

застосовується як на сервісних підприємствах, віддалених від персоналу. Так, кафе самообслуговування та відвідування музею без екскурсовода є прикладами того, як клієнт бере на себе виконання тих завдань, які в іншому випадку лягли б на обслуговуючий персонал компанії. У цьому змінюється роль клієнта. Клієнт стає учасником процесу створення послуги, що діє у співпраці із сервісною компанією. Клієнти, які обслуговуються з низьким ступенем взаємодії, зазвичай ніколи не бачать місця, де виконується робота; крім того, найчастіше вони спілкуються з представником компанії, який допомагає їм вирішити ту чи іншу проблему телефоном. Не бачачи ні офісу, ні обстановки в ньому, ні навіть співробітника фірми, з яким вони контактують, клієнти змушені формувати свою думку про якість обслуговування на основі того, наскільки швидко їм вдалося додзвонитися до компанії, наскільки приємний і ввічливий голос співрозмовника та наскільки швидко співробітник фірми вирішує проблему. Якщо послуга надається через загальні електронні канали, вводиться певна імітація традиційного обслуговування людини людиною. Деякі фірми уособлюють автомати самообслуговування або автоматичні телефонні дзвінки на центральний комп'ютер офісу або домашній комп'ютер клієнта, шляхом присвоєння апаратурі імені, програючи музику або анімуючи графічне зображення, додаючи різні звукові ефекти та інші інтерактивні засоби, що дозволяють певною мірою відтворити традиційні відчуття людини у процесі. Для цього, наприклад, деякі web-сайти оформляються так, щоб їхній вигляд максимально нагадував людям вітрину магазину. Опосередковані новими технічними засобами, контакти, хоч, і видозмінилися, проте зберегли свою безпосередність та участь у них фахівців, які мають спеціальні знання з профілю послуг. У сервісних підприємствах, у яких клієнти вступають у взаємодію один з одним, - наприклад, у готелях, ресторанах, авіакомпаніях або автозаправних станціях, - їх поведінкою необхідно обережно керувати для того, щоб їхні дії не суперечили стратегії компанії, включаючи такі найважливіші елементи, як зручність та безпека інших клієнтів. Чинники, що впливають на систему надання послуг:

- місце розташування підприємства з надання послуг;

- потреби та бажання споживачів зазвичай передують ефективності;
- календарне планування робіт залежить переважно від споживачів;
- визначення та вимірювання якості послуг;
- персонал повинен мати добрі навички спілкування з клієнтами;
- виробнича потужність розраховується за “піковим” попитом, а чи не за середнім рівнем;
- ефективність роботи службовців важко піддається виміру;
- великі підприємства у сфері послуг нетипові, крім банків, авіакомпаній;
- маркетинг та виробництво у сфері послуг часто важко поділити.

Кваліфікація персоналу - встановлення якісної потреби у навчанні.

Розмір підприємств у сфері послуг поділяється на мікропідприємства, малі підприємства, великі підприємства.

Маркетинг послуг - процес розробки, просування та реалізації послуг, орієнтований виявлення специфічних послуг. Придбання будь-якого товару чи послуг завжди пов'язане покупця з певним ризиком. У сфері послуг рівень ризику різко зростає. Це з характеристиками послуг, невловимістю, мінливістю. Ступінь купівельних ризиків у сфері послуг залежить від невизначеності послуги: чим вищий ступінь невизначеності, тим вищі ризики.

Крім цього, рівень ризику залежить від:

- значущості послуги;
- вартості послуги;
- ступеня самовпевненості клієнта, яка визначиться його знаннями
- та досвідом.

Для зниження ризиків у сфері послуг продавець має визначити очікування клієнта до отримання послуги та підготувати його до результатів. Необхідно також допомогти клієнтам зрозуміти зміст послуги та те, що вони мають отримати протягом та після отримання послуги. Крім цього, для зменшення ризику та залучення клієнтів, якщо можливо, доцільно надавати пробні послуги, наприклад, перше заняття у спортивному комплексі може бути

безкоштовним. Придбання будь-якого товару чи послуг завжди пов'язане покупця з певним ризиком. Одна з можливостей зменшити сумніви клієнта і знизити ризик – надання гарантій на послуги. Для зменшення мінливості послуг та зменшення ризиків доцільно розробляти стандарти обслуговування. Стандарт обслуговування – це комплекс обов'язкових до виконання правил обслуговування клієнтів, які мають гарантувати встановлений рівень якості всіх операцій. Розробка стандарту обслуговування включає формування товарного знака, фірмового кольору, логотипу, які поширюються на всю рекламно-сувенірну продукцію.

Існує два основні підходи до розробки веб-додатків: односторінкові (SPA), багатосторінкові (MPA). Вони виділяються серед інших підходів простотою розробки, зручністю для користувачів та широкими можливостями для розвитку бізнесу.

Розповідаємо, чим відрізняються компоненти MPA, SPA, які у них переваги та недоліки, що з них вибрати і для яких завдань.

SPA або Single Page Application — це односторінковий веб-додаток, який завантажується на одну HTML-сторінку. Завдяки динамічному оновленню за допомогою JavaScript під час використання не потрібно перезавантажувати або підвантажувати додаткові сторінки. На практиці це означає, що користувач бачить у браузері весь основний контент, а при прокручуванні або переходах на інші сторінки замість повного перезавантаження потрібні елементи просто підвантажуються.

У процесі роботи користувачеві може здатися, що він запустив не веб-сайт, а десктопний додаток, оскільки він миттєво реагує на всі його дії, без затримок та «підвисань».

Такого ефекту вдається досягти за допомогою просунутих фреймворків JavaScript: Angular, React, Ember, Meteor, Knockout.

Приклади динамічних програм: Gmail, Google Maps, Facebook, GitHub, Meduza.

Переваги:

- Висока швидкість - всі ресурси завантажуються за одну сесію, а під час дій на сторінці дані просто змінюються, що дуже економить час;
- гнучкість і чуйність інтерфейсу користувача - за рахунок того, що веб-сторінка всього одна, простіше побудувати насичений інтерфейс, зберігати відомості про сеанс, керувати станами уявлень і анімацією;
- спрощена розробка – код можна починати писати з файлу `file://URL`, не використовуючи сервер, не потрібен окремий код для рендеру сторінки на стороні сервера;
- кешування даних — програма надсилає лише один запит, збирає дані, а після цього може функціонувати в `offline`-режимі.

Недоліки:

- Seo оптимізація вимагає рішень у вигляді серверного рендерингу - через те, що контент завантажується за допомогою технології AJAX, яка передбачає динамічну зміну змісту станиці, а для оптимізації важлива стійкість;
- навантаження на браузер - через те, що клієнтські фреймворки важкі, вони досить довго завантажуються;
- потрібна підтримка JavaScript - без JS не можна повноцінно користуватися повним функціоналом програми;
- витік пам'яті в Java Script - через поганий захист, SPA більше схильна до дій зловмисників і витоку пам'яті.

1.2 Аналіз існуючих систем аналогів

У наш час будь-яке підприємство чи компанія, яка надає послуги, зобов'язана мати інформаційну систему для розширення своєї клієнтської бази. Інформаційна система може бути як у вигляді мобільного додатка, так і у вигляді веб-застосунку, спочатку я розгляну два веб-застосунки. Основним завданням даних веб-додатків є продаж товару, але я розгляну ці два веб-додатки з системи

надання послуг оскільки дані майданчики це місце для розповсюдження будь якого власного товару, обидва обрані веб-застосунки надають можливість усім бажаючим продавати свої товари на їх майданчиках. Спочатку я розгляну веб-додаток, який пропонує компанія розетка. Цей веб-додаток має особливий дизайн, який відрізняється від тих, який рекомендує більшість фахівців з дизайну. На ньому відсутній футер. Футер – це область, яка розташована внизу кожної сторінки сайту під основним контентом. Замість нього внизу знаходиться пусте біле місце. Найчастіше користувачі щоб дізнатися якусь додаткову інформацію про веб-додаток або підприємство, яке використовує веб-додаток, шукають інформацію у футері. Але в даному веб-додатку футер є повністю. Він винесений у ліву частину сайту там де і знаходиться каталог товарів. Це рішення безпосередньо впливають на оцінку користувача даного веб-додатку. Також у веб-додатку немає розділу в якому б знаходилися відповіді на найчастіші питання, відсутність цього розділу є істотним мінусом даної системи. Люди, які вперше користуються подібними веб-додатками, мають питання, і найчастіше в даному розділі знаходяться основні відповіді на їхні запитання, а також є можливість поставити питання, яке цікавить потенційного клієнта. Відсутність цього розділу – це дуже великий мінус для даної системи. Оскільки зворотний зв'язок з клієнтами та грамотний дизайн є одним із ключових факторів успіху веб-додатку та популярності серед користувачів.

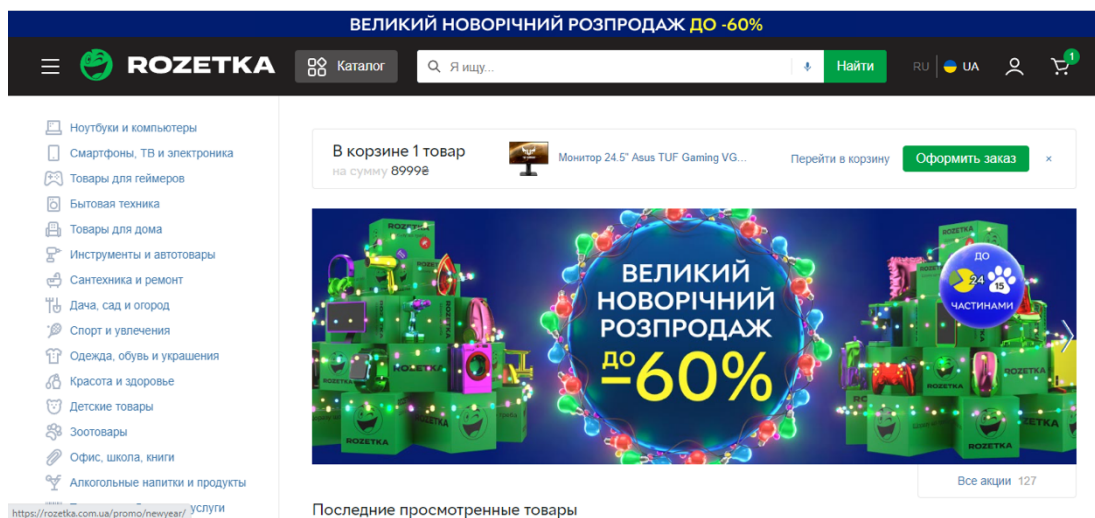


Рисунок 1.1 Клієнтський інтерфейс веб-додатку Rozetka

Наступним суттєвим мінусом даної системи є неможливість здійснити передоплату на сайті. Клієнту надається можливість лише сплатити повну вартість товару. Так як на даному майданчику продавати свої товари можуть всі, то якщо продавцем товару є не сама розетка, а сторонній продавець то він не відправить товар поштою доки не буде здійснено передоплату найчастіше це 100 гривень. Продавець товару зв'язується в вами по телефону і пропонує здійснити передоплату на номер рахунку, який він скидає по смс, таке рішення не є оптимальним, оскільки бувають випадки, коли продавець не виходить на зв'язок і таким чином посилка не відправляється. В наш час бувають ситуації, коли у продавця може бути відсутній зв'язок та інтернет, і він не може зв'язатися з клієнтом, що також затримує відправку посилки. Вирішенням цієї проблеми могла б стати можливість внести передоплату прямо на сайті. Але такої функції у цій системі немає.

Рисунок 1.2 Клієнтський інтерфейс оплати

Наступним веб-додатком, який я розгляну, буде веб-додаток від компанії алло. На відміну від попередньої програми, тут немає проблем з дизайном. Він виконаний за класичним шаблоном. Але водночас тут також є питання щодо деяких розділів. При спробі переглянути інформацію про компанію, програма

відкриває нову вкладку і відкриває в ній сайт. Більшість користувачів вкрай негативно ставляться до додаткових дій, які потрібно здійснювати у веб-додатку, щоб отримати бажаний результат. Одним з основних факторів успіху великих систем надання послуг є мінімальна кількість дій, яку здійснює клієнт, щоб отримати бажаний результат. Переходи на інші сторінки викликають невдоволення у багатьох користувачів. Що стосується купівлі товарів, то тут такий самий недолік, як і в попередній системі. На відміну від попередньої системи в цьому додатку є розділ у якому знаходяться відповіді на найчастіші питання.

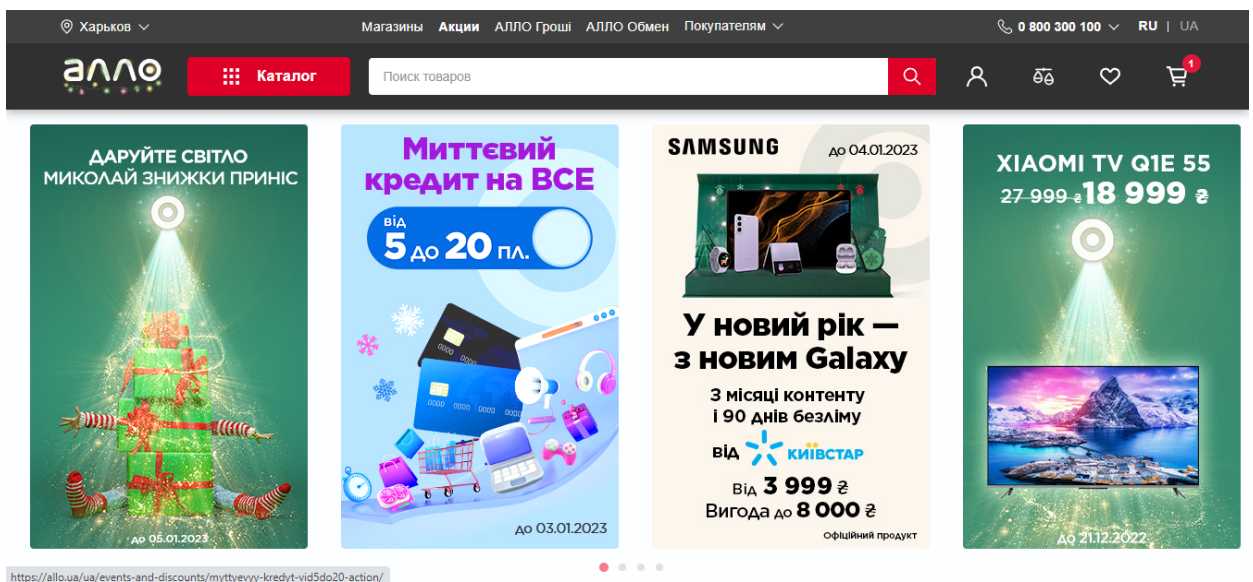


Рисунок 1.3 Клієнтський інтерфейс веб-додатку Allo

Загальним недоліком двох систем є жахлива взаємодія з іншими системами надання послуг. Оформляючи покупку в одному з двох веб-додатків, користувач стикається з тим, що він не має інформації про те, що відбувається з його посилкою і де вона знаходиться. Раніше такі системи як нова пошта автоматично розсилали смс повідомлення в якому було вказано час відправки та номер посилки, завдяки якому клієнт міг відстежити свою посилку. В даний момент при покупці товару відповідальність за відправку смс повідомлення з даними про покупку лежить на продавці, а не на системі нової пошти. Нерідко трапляються

випадки коли продавці не відправляють смс повідомлення з номером посилки, і клієнт не має жодної інформації про те, де знаходиться його посилка і коли вона прибуде.

У зв'язку з цим вважаю, що доцільно також у контексті систем, що надають послуги, розглянути веб-додаток нової пошти. Дизайн програми дуже простий, але в той же час інтуїтивно зрозумілий. В той же час у ньому є свої недоліки. Інформація про тариф написана дуже дрібним шрифтом, і для людей, які мають проблеми із зором, він не є читабельним. Основним недоліком даної системи є те, що одна з найголовніших функцій цієї системи це відстеження посилки знаходиться поза основним веб-додатком. У той же час, коли у всіх програм конкурентів можливість відстеження посилки знаходиться всередині основного веб-додатку. Також під час відстеження посилки системою чомусь використовується англійська часова градація. Більшість дорослого населення не володіє англійською та використання англійської позначення часу вводить в оману частину клієнтів. Також важливою проблемою є використання англійських слів. Послугами поштових систем користуються всі вікові категорії, тому краще під час найменування розділів використовувати державну мову.

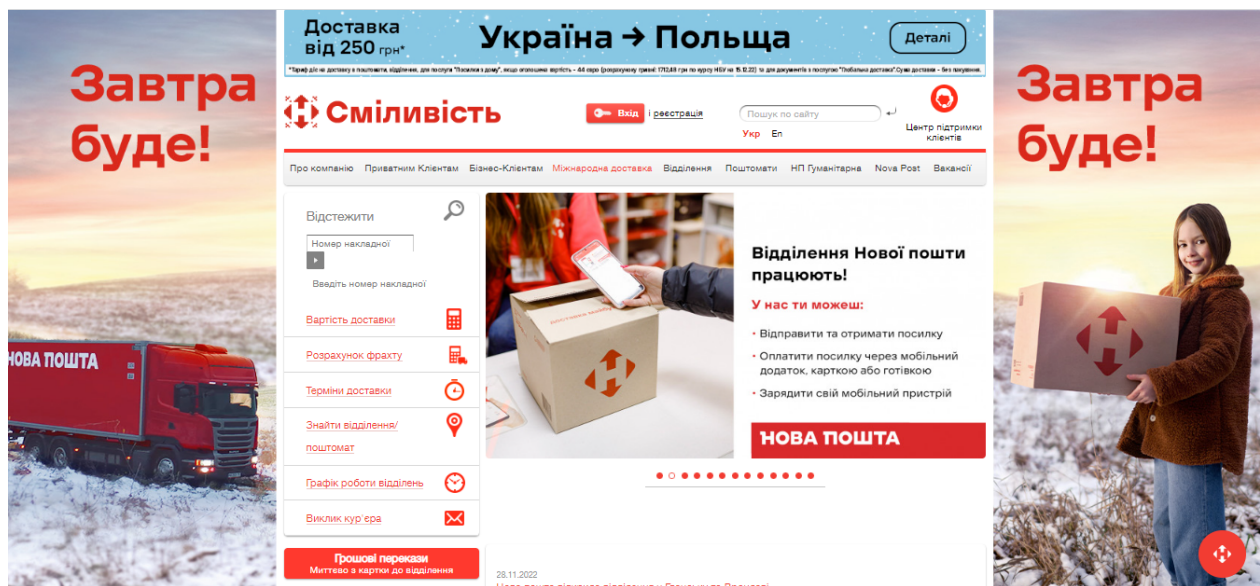


Рисунок 1.4 Клієнтський інтерфейс веб-додатку Нової пошти

2 ПОСТАНОВКА ЗАДАЧІ ДОСЛІДЖЕННЯ

Проаналізувати основні проблеми та засоби створення інформаційного забезпечення систем надання послуг. Виявити загальні риси та потреби систем вказаного класу щодо інформаційного забезпечення. Порівняти існуючі підходи та технології, виробити рекомендації щодо розробки інформаційного забезпечення систем надання послуг.

Реалізувати розроблені рекомендації на прикладі розробки інформаційного забезпечення системи надання професійних навчальних послуг із самооборони з дотриманням вимог клієнтів. В якості основного фреймворка виростати Spring. Для захисту системи від різних злочинців рекомендується використати Spring Security. Під час демонстрації сторінок використовувати Thymeleaf.

Дизайн веб-сервісу має бути простим і не нав'язливим. Додаток повинен мати грамотний інтерфейс користувача. Найбільш зручний, орієнтований на вашу цільову аудиторію. Програма повинна мати логічну навігацію. Основні розділи сайту краще розмістити зверху, корисні посилання, дружні сайти тощо – у бічному меню. Розділ "Про проект". Ви не повинні бути для читача абстрактною фігурою. Це відштовхує. Це обов'язкова умова: зв'язок з вами має бути зручним і найпрямішим. Реєстрація має бути максимально спрощена. Реєстрація та підписка має у своєму розпорядженні будь-якого читача. Веб-додаток повинен мати карту сайту. Вона допомагає скласти читачеві повне представлення вашого ресурсу, а також сприяє індексації сторінок. Відокремте контент від дизайну. Html не дає такої можливості. Використовуйте CSS-файли для того, щоб пошукові системи могли, минаючи надлишкові коди, ідентифікувати контент. Використовуйте XHTML/CSS. Крім того, коди мають бути точними. Два відмінні ресурси перевірки коду: W3C Markup Validation Service, W3C CSS Validation Service. Оптимізація зображень. Зменшуйте зображення, намагаючись максимально зберегти якість.

Методологія побудови баз даних виходить з теоретичних основ їх проектування. Для розуміння концепції методології наведемо основні її ідеї у вигляді двох етапів, що послідовно реалізуються на практиці:

1-й етап - обстеження всіх функціональних підрозділів фірми з метою:

- зрозуміти специфіку та структуру її діяльності;
- побудувати схему інформаційних потоків;
- проаналізувати існуючу систему документообігу;
- визначити інформаційні об'єкти та відповідний склад реквізитів (параметрів, характеристик), що описують їх властивості та призначення.

2-й етап - побудова концептуальної інформаційно-логічної моделі даних для обстеженої на 1-му етапі сфери діяльності. У цій моделі мають бути встановлені та оптимізовані всі зв'язки між об'єктами та їх реквізитами. Інформаційно-логічна модель є фундаментом, на якому буде створено базу даних.

Для створення інформаційного забезпечення необхідно:

- ясне розуміння цілей, завдань, функцій усієї системи управління організацією;
- виявлення руху інформації від моменту виникнення та до її використання на різних рівнях управління, що представлена для аналізу у вигляді схем інформаційних потоків;
- удосконалення системи документообігу;
- наявність та використання системи класифікації та кодування;
- володіння методологією створення концептуальних інформаційно-логічних моделей, що відбивають взаємозв'язок інформації;
- створення масивів інформації на машинних носіях, що потребує наявності сучасного технічного забезпечення.

На основні розроблених рекомендацій побудувати систему надання професійних навчальних послуг із самооборони на основі вимог цільової аудиторії, яка включатиме такі основні параметри:

- кросплатформність (підтримка різними браузерями);
- потреби кожного клієнта, що до вибору напрямку;
- багатий вибір кваліфікованих спеціалістів;
- детальний опис кожного напрямку який викладається в спортивному клубі.

Побудувати базу даних яка включає в себе інформацію про клієнтів, інформацію про тренерів, інформацію про інвентар який використовується нашим спортивним клубом, загальний опис спортивного комплексу, а також розділів які в спортивному комплексі викладаються.

Вимоги до інформації баз даних повинні містити особливості забезпечення її надійності, актуальності (достовірності), ефективності використання обчислювальних ресурсів та прийняттого рівня супроводу. Зміст та сутність цих конструктивних характеристик як базових понять та характеристик якості доцільно використовувати при проектуванні інформаційних систем. Заходи та шкали для оцінювання конструктивних характеристик значною мірою можуть застосовуватися ті самі, що при аналізі якості програмних засобів. Особливо виділяються характеристики достовірності даних та захищеності інформації.

Окремо відзначимо таку характеристику як захищеність інформації БД, яка реалізується, в основному, програмними засобами СУБД, однак у поєднанні з засобами організації та захисту даних, що їх підтримують. Цілі, призначення та функції захисту тісно пов'язані з особливостями функціональної придатності кожної БД. При системному проектуванні властивості захищати інформацію баз даних від негативних впливів описуються зазвичай складом і номенклатурою методів і засобів, що використовуються для захисту від зовнішніх і внутрішніх загроз. Непрямим показником її якості може бути відносна частка обчислювальних ресурсів, використовуваних безпосередньо засобами захисту БД.

Основна увага в практиці забезпечення безпеки застосування БД зосереджена на захисті від зловмисних руйнувань, спотворень та розкрадань інформації баз даних. Основою такого захисту є аудит доступу, а також контроль організації та ефективності обмежень доступу. У реальних системах баз даних повинні враховуватися наслідки реалізації загроз, джерелами яких є випадкові, непередбачувані, дестабілізуючі фактори або дефекти і відсутні безпосередньо зацікавлені особи в подібних порушеннях. Якість захисту систем баз можна характеризувати величиною потенційної шкоди, ризик виникнення якого при прояві дестабілізуючих факторів та реалізації конкретних загроз безпеці вдається

запобігти або знизити. Також можливою характеристикою якості може бути середній час між можливими проявами загроз, що долають захист даних.

3. ОГЛЯД МЕТОДІВ ТА ТЕХНОЛОГІЙ ЩО ВИКОРИСТОВУЮТЬСЯ

3.1 Обґрунтування вибору програмних засобів для розробки програмних засобів

Java - це мова програмування загального призначення, який слід парадигмі об'єктно-орієнтованого програмування і підходу «Написати один раз і використовувати скрізь». Java використовується для настільних, мережевих, мобільних і корпоративних додатків

Java - це не тільки мова програмування, а й екосистема інструментів, що охоплює майже всі, що може знадобитися при програмуванні на Java. У неї входять:

Java Development Kit (JDK) - комплект розробника Java. За допомогою JDK і стандартного блокнота можна писати і запускати / компілювати код на Java.

Java Runtime Environment (JRE) - виконуюча система Java. Механізм розповсюдження програмного забезпечення, складається з автономної віртуальної машини Java, стандартної бібліотеки Java (Java Class Library) і інструментів настройки.

Integrated Development Environment (IDE) - інтегроване середовище розробки. Інструменти, які допомагають запускати, редагувати і компілювати код. Найпопулярніші з них - IntelliJ IDEA, Eclipse і NetBeans.

Java можна знайти всюди. Це основна мова розробки для Android. Він використовується в веб-додатках, урядових веб-сайтах і технологіях обробки великих даних, таких як Hadoop і Apache Storm. Java підходить і для наукових проєктів, особливо в області обробки природної мови. Мова Java переважав і в програмуванні для мобільних пристроїв, задовго до появи смартфонів - перші мобільні ігри на початку 2000-х років були написані на Java. Java, завдяки своїй довгій історії, заробив своє місце в Залі слави програмування. Індекс TIOBE, один з найавторитетніших індексів популярності програм в світі, при складанні рейтингу використовує результати пошукової видачі. Незважаючи на зростаючу

популярність Go і Python, Java залишається на вершині списку вже понад десятиліття.

Java - вже не єдиний офіційно підтримувана мова для розробки на Android. Java далеко не єдиний вибір в веб-програмуванні. Проте, Java йде в ногу з часом. Давайте розглянемо, які переваги пропонує Java.

Java включає в себе об'єктно-орієнтоване програмування (ООП) - концепцію, в якій ви не тільки визначаєте тип даних і його структуру, а й набір функцій, що застосовуються до нього. Таким чином, структура даних стає об'єктом, яким можна управляти для створення відносин між різними об'єктами.

При іншому підході - процедурному програмуванні - потрібно слідувати чітким інструкціям, використовувати змінні і функції. При ООП можна групувати ці змінні і функції за допомогою контексту, маркувати їх і посилатися на функції в контексті кожного конкретного об'єкта. Порівняння процедурного та об'єктно-орієнтованого програмування показано на рисунку 2.1

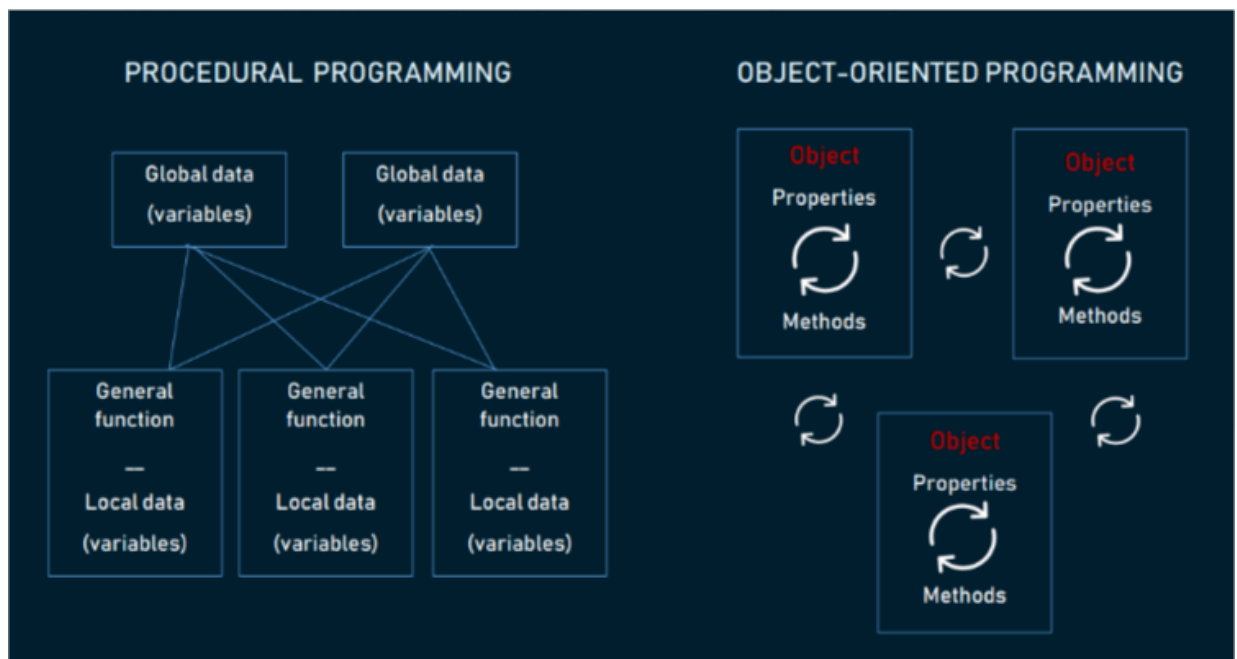


Рисунок 3.1 – Порівняння процедурного та об'єктно-орієнтованого стилю

При ООП можна повторно використовувати об'єкти в інших програмах.

ООП запобігає помилки, оскільки об'єкти приховують інформацію, до якої не повинно бути доступу.

ООП більш ефективно організовує структуру програм, в тому числі великих ООП спрощує обслуговування і модернізацію старого коду.

Java - мова високого рівня з простим синтаксисом і плавною кривою навчання.

Java - це мова високого рівня, тобто він схожий на людську мову. На відміну від мов низького рівня, які нагадують машинний код. Мови високого рівня перетворюється за допомогою компіляторів або інтерпретаторів. Це спрощує розробку, роблячи мову більш легким для написання, читання і обслуговування.

Синтаксис Java заснований на C ++, тому Java схожа на C. Проте, синтаксис Java простіше, що дозволяє новачкам швидше вчитися і ефективніше використовувати код для досягнення конкретних результатів.

Java не так доброзичливий до новачків, як Python, проте досить простий для будь-якого розробника з базовим розумінням фреймворків, пакетів, класів і об'єктів. Він простий, типізований і передбачуваний, що дозволяє вчитися мислити в правильному напрямку. Крім того, новачок завжди може звернутися до безлічі безкоштовних онлайн-уроків і курсів.

Стандарт для корпоративних обчислювальних систем

Корпоративні програми - головна перевага Java з 90-х років, коли організації почали шукати надійні інструменти програмування не на C. Java підтримує безліч бібліотек - будівельних блоків будь-якої корпоративної системи. Бібліотеки допомагають розробникам створювати будь-які функції, які можуть знадобитися компанії. Java широко поширений - це мова, яку викладають в рамках введення в програмування в більшості шкіл і університетів. Можливості інтеграції Java вражають: більшість хостинг-провайдерів підтримують Java. Більш того, Java - мова, дешевий в обслуговуванні: працювати з Java можна з будь-якого комп'ютера, незалежно від конкретної апаратної інфраструктури.

Існує думка, що Java - безпечна мова, проте це не зовсім так. Сама мова не захищає вас від вразливостей, але деякі його функції усувають поширені уразливості. По-перше, на відміну від C, в Java немає покажчиків. Покажчик - це об'єкт, який зберігає адресу осередки пам'яті іншого значення, що може викликати несанкціонований доступ до пам'яті. По-друге, в Java є Security Manager, створена для кожної програми політика безпеки, в якій можна вказати правила доступу. Це дозволяє запускати додатки Java в «пісочниці» і усувати таким чином уразливості.

«Написати один раз і використовувати скрізь» (WORA) - популярна в IT-сфері фраза, за допомогою якої Sun Microsystems описує крос-платформні можливості Java. Можна створити Java-застосунок на Windows, скомпілювати його в байт-код і запустити його на будь-якій іншій платформі, яка підтримує віртуальну машину Java (JVM). Таким чином, JVM служить рівнем абстракції між кодом і обладнанням. Кроссплафтоменість Java подана на рисунку 2.2

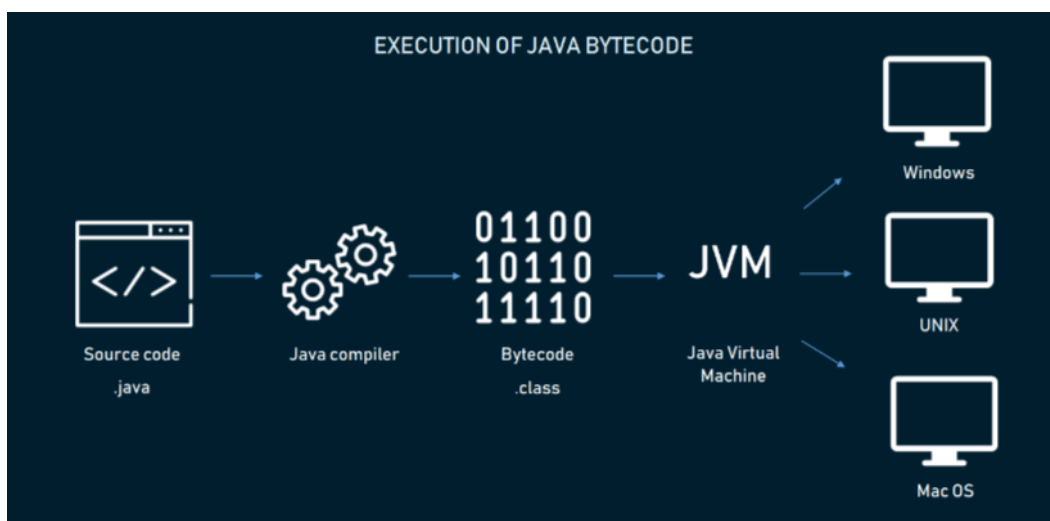


Рисунок 3.2 – Кроссплафтоменість Java.

Всі основні операційні системи, включаючи Windows, Mac OS и Linux, підтримують JVM. Если ваша програма НЕ спірається на специфічні для платформи Функції и призначення для користувача інтерфейс, ее можна з легкістю перенести: прінаймні, більшу ее часть.

Мова для розподіленого програмування и комфортно віддаленої комунальної роботи

Java створювалася як мова для розподіленого програмування: Він має вбудований Механізм Спільного використання Даних и програм декількома комп'ютерами, що підвищує продуктивність и ефективність праці.

В інших мовах потрібно використовувати зовнішній API для дистрибуції. В Java ця технологія вбудована. Специфічна для Java методологія розподілених обчислень називається Remote Method Invocation (RMI). RMI дозволяє використовувати всі переваги Java: безпека, незалежність від платформи і об'єктно-орієнтоване програмування для розподілених обчислень. Крім того, Java також підтримує програмування сокетів і методологію розподілу CORBA для обміну об'єктами між програмами, написаними на різних мовах.

Розробникам Java не потрібно вручну писати код для управління пам'яттю завдяки автоматичному управлінню пам'яттю (АММ). АММ також використовується в мові програмування Swift і при очищенні пам'яті додатками, які автоматично обробляють розподіл і звільнення пам'яті. Що саме це означає?

Ефективність програми безпосередньо пов'язана з пам'яттю. При цьому обсяг пам'яті обмежений. При написанні програми на мовах з ручним керуванням пам'яттю, розробники ризикують забути виділити пам'ять, що призведе до збільшення обсягу займаної додатком пам'яті і проблем з продуктивністю. Програми очищення пам'яті шукають об'єкти, які більше не використовуються програмою, і видаляють їх. Це впливає на роботу процесора, однак розумна оптимізація та налаштування дозволяють знизити цей вплив.

Потік - найменша одиниця обробки в програмуванні. Щоб максимально ефективно використовувати час процесора, Java дозволяє запускати потоки одночасно, що називається багатопоточність.

Потоки використовують одну і ту ж область пам'яті, тому між ними можна швидко перемикатися. Потоки незалежні один від одного: один потік не впливає на роботу інших потоків. Це особливо корисно в іграх і програмах з великим об'ємом анімації.

Уже багато років розвитку Java сприяють співтовариство, підтримка Oracle і достаток додатків і мов на JVM. Крім того, постійно випускаються нові версії Java з новими цікавими функціями.

Спільнота розробників Java не має собі рівних. Близько 45% респондентів опитування StackOverflow 2018 використовують Java. У Java надзвичайно велика екосистема добре протестованих бібліотек і фреймворків для будь-яких завдань. Початківець розробник, швидше за все, вибере Java: на тему Java-програмування існує понад 1000 курсів на Udemy і більше 300 на Coursera.

Нещодавно Oracle оголосила, що з 2019 року компанія почне стягувати плату за використання Java Standard Edition 8 в «комерційних цілях». За все нові оновлення та виправлення помилок доведеться заплатити. Плата залежить від кількості користувачів або комп'ютерів.

Поточна версія Java безкоштовна для простого використання. Таким чином, кожна використовує Java компанія повинна оцінити, наскільки ефективно вона використовує Java. Компанія повинна зрозуміти, що вигідніше: шукати альтернативне рішення або продовжувати користуватися Java.

У будь-якої мови високого рівня досить низька продуктивність через компіляції та абстракції за допомогою віртуальної машини. Однак це не єдина причина низької швидкості Java. Наприклад, застосунок очищення пам'яті: це корисна функція, яка, на жаль, призводить до значних проблем з продуктивністю, якщо вимагає більше 20 відсотків часу процесора. Погана настройка кешування може викликати надмірне використання пам'яті. Існує також взаємне блокування потоків: так відбувається, коли кілька потоків намагаються отримати доступ до одного і того ж ресурсу. В цьому випадку відбувається кошмар кожного Java-розробника - помилка через брак пам'яті. Проте вміле планування може вирішити всі ці проблеми.

Для створення графічного інтерфейсу користувача (GUI) розробники використовують різні інструменти, орієнтовані для конкретної мови. Для Android-додатків є Android Studio, яка допомагає створювати додатки з нативним

дизайном. Однак, коли справа доходить до призначеного для користувача інтерфейсу на ПК, Java-інструменту для створення нативного дизайну немає.

Є кілька інструментів для розробки GUI для Java: найпопулярніші з них - Swing, SWT, JavaFX, JSF. Бібліотека Swing - це старий, але надійний крос-платформний інструмент, інтегрований в різні Java-IDE, в тому числі Eclipse і NetBeans. Однак, якщо ви не використовуєте шаблони, ви помітите невідповідності інтерфейсу. SWT використовує власні компоненти, але не підходить для складного інтерфейсу. JavaFX - лаконічний і сучасний, але занадто новий. В цілому, перед створенням GUI на Java потрібно докладніше вивчити інструменти.

Багатослівність коду може здатися перевагою, яке допоможе при вивченні мови. Однак, довгі, надмірно складні речення ускладнюють читання і перегляд коду. Як і природні мови, багатьма мовами програмування високого рівня містять зайву інформацію. Java - це легша версія неприступного C ++, яка змушує програмістів прописувати свої дії словами з англійської мови. Це робить мову більш зрозумілим для неспеціалістів, але менш компактним.

Обґрунтування вибору платформи СУБД

Раніше для довготривалого зберігання інформації ми працювали з файлами: поміщали в них деяку кількість рядків, а потім витягували їх для подальшої роботи. Завдання тривалого зберігання інформації дуже часто зустрічається в програмуванні Web-додатків: підрахунок відвідувачів в лічильнику, зберігання повідомлень у форумі, віддалене управління змістом інформації на сайті і т.д.

Тим часом, професійні прийоми роботи з файлами дуже трудомісткі: необхідно піклується про приміщенні в них інформації, про її сортування, добування, при цьому не потрібно забувати, що всі ці дії будуть відбуватися на сервері хост-провайдера, де з дуже великою ймовірністю варто один з варіантів Unix - отже, потрібно так само піклується про права доступу до файлів і їх розміщенні. При цьому обсяг коду значно зростає, і зробити помилку в програмі дуже просто.

Всі ці проблеми вирішує використання бази даних. Бази даних самі дбають про безпеку інформації та її сортування і дозволяють отримувати і розміщувати інформацію за допомогою одного рядка. Код з використанням бази даних виходить більш компактним, і налагоджувати його набагато легше. Крім того, не потрібно забувати і про швидкість - вибірка інформації з бази даних відбувається значно швидше, ніж з файлів.

База даних - це спеціально створене сховище важливої інформації, невід'ємним атрибутом якого є зручний доступ до всіх даних, що зберігаються. Для забезпечення цього доступу використовуються так звані системи управління базами даних або просто СУБД, що представляють собою особливі додатки на базі певної програмної архітектури

SQL - мова запитів, призначений для організації управління реляційними БД (з пов'язаними між собою даними). Він багатofункціональний і з його допомогою можна коригувати, створювати і прибирати дані з таблиць, запитувати з них інформацію, створювати самі таблиці і т.д.

Однією з найпопулярніших СУБД на сьогоднішній день є MySQL, яка поширюється вільно (з деякими обмеженнями). Ця серверна система здатна ефективно функціонувати у взаємодії з інтернет-сайтами та веб-додатками. При цьому вона проста в освоєнні, що лише збільшує її популярність, особливо серед «айтішників» -аматорів.

Важливим фактором є її безкоштовність. MySQL поширюється на умовах загальної ліцензії GNU (GPL, GNU Public License).

Примітно, що результатом величезної популярності MySQL стала поява в інтернеті безлічі посібників з освоєння системи, а також величезної кількості всіляких плагінів і розширень, що спрощують роботу з цією системою. Це в свою чергу лише додало системі ще більшої популярності.

Незважаючи на відсутність деякого функціоналу, наявного у інших СУБД, MySQL володіє досить великим розмаїттям доступних інструментів для створення додатків.

Крім універсальності і поширеності СУБД MySQL володіє цілим комплексом важливих переваг перед іншими системами. Зокрема слід відзначити такі якості як:

Простота у використанні. MySQL досить легко інсталується, а наявність безлічі плагінів і допоміжних додатків спрощує роботу з базами даних.

Великий функціонал. Система MySQL володіє практично всім необхідним інструментарієм, який може знадобитися в реалізації практично будь-якого проекту.

Безпека. Система спочатку створена таким чином, що безліч вбудованих функцій безпеки в ній працюють за замовчуванням.

Масштабованість. Будучи досить універсальною СУБД, MySQL в рівній мірі легко може бути використана для роботи і з малими, і з великими об'ємами даних.

Швидкість. Висока продуктивність системи забезпечується за рахунок спрощення деяких використовуваних в ній стандартів.

Як і будь-який програмний продукт, система MySQL має певні обмеження в своєму функціоналі, що не дозволяє використовувати її для роботи з додатками, що мають деякі специфічні вимоги. До недоліків цієї СУБД відносяться:

Недостатня надійність. У питаннях надійності деяких процесів по роботі з даними (наприклад, зв'язок, транзакції, аудит) MySQL поступається деяким іншим СУБД.

Низька швидкість розробки. Як і багатьом іншим програмним продуктам з відкритим кодом, MySQL бракує деякого технічної досконалості, що часом позначається на ефективності процесів розробки.

Сам факт того, що MySQL є найпопулярнішою СУБД на ринку, говорить про її універсальності і перевагу в більшості ситуацій. Зокрема її використовують в наступних ситуаціях:

при розподілених операціях, коли функціоналу SQLite (інша популярна система) не вистачає;

коли потрібно забезпечити високий рівень безпеки, що MySQL робить з успіхом;

для роботи з інтернет-сторінками та веб-додатками, оскільки MySQL є найбільш зручною СУБД для цієї сфери застосування;

при роботі зі специфічним проектом, де функціонал MySQL дає оптимальний результат.

Проте, існують ситуації, коли від MySQL все ж слід відмовитися на користь інших СУБД. Наприклад, вона не годиться якщо:

- потрібно відповідати стандарту SQL, який дана система підтримує лише частково;
- проект передбачає багатопоточність даних, оскільки при здійсненні паралельних операцій читання / запису у MySQL можуть виникати проблеми;
- наявний функціонал MySQL не здатний забезпечити весь набір можливостей роботи з базою даних.

Системні адміністратори блокують персоналу доступ до деяких сайтів, на кшталт соцмереж, торрент-треккерів або ігор.

При розробці сайтів, додатків та інших програмних продуктів використовується досить багато відмінних.

Без інтернету сьогодні складно уявити повноцінне життя сучасної людини: навчання, робота, бізнес, дозвілля, об.

Висновки: в ході виконання 2-го пункту були описані способи, методи та рішення, які використовуються для розробки веб-сервісів. На мою думку були обрані найкращі з них: обширний функціонал розробки, проста мова, яку не важко освоїти, бібліотеки, що нівелюють недоліки мови та мають гарний функціонал, та спосіб, заснований на бізнес-правилах, котрий дасть можливість реалізувати задану ідею без зайвих затрат по часу.

3.2 Модель бази даних

Визначення посилальної цілісності подано в таблиці 3.1

Таблиця 3.1– Визначення посилальної цілісності

№	Ім'я таблиці 1, зовнішній ключ	Ім'я таблиці 2, первинний ключ	Тип посилальної цілісності
1	User, idrole	Role, idrole	Cascade
2	Inventory, idhall	Hall, idhall	Cascade
3	Subscription, idcoach	Coach idcoach	Restict
4	Subscription, iduser	User, iduser	Restict
5	Subscription, idhall	Hall, idhall	Restict

Схема бази даних подана на рисунку 3.6.

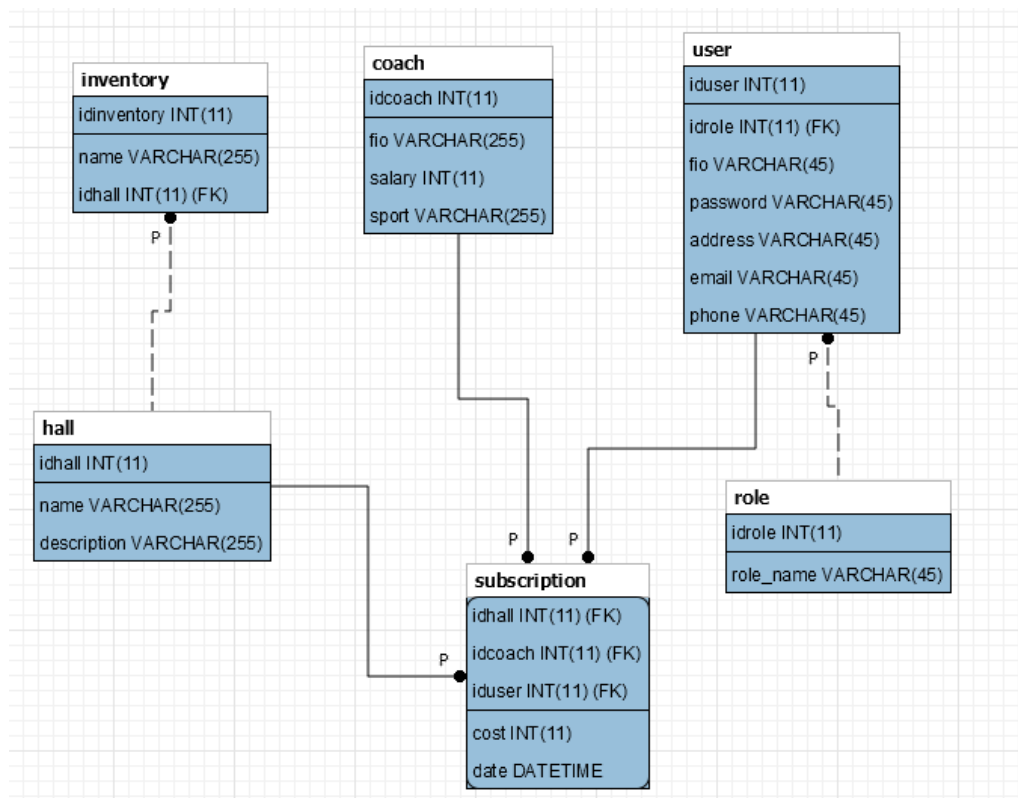


Рисунок 3.3 – Схема бази даних

3.3 Розробка інтерфейсу клієнтської частини

Однією з вимог до програмного засобу є простий і зрозумілий інтерфейс користувача. На рисунку 3.4 відображено макет головної сторінки сайту.

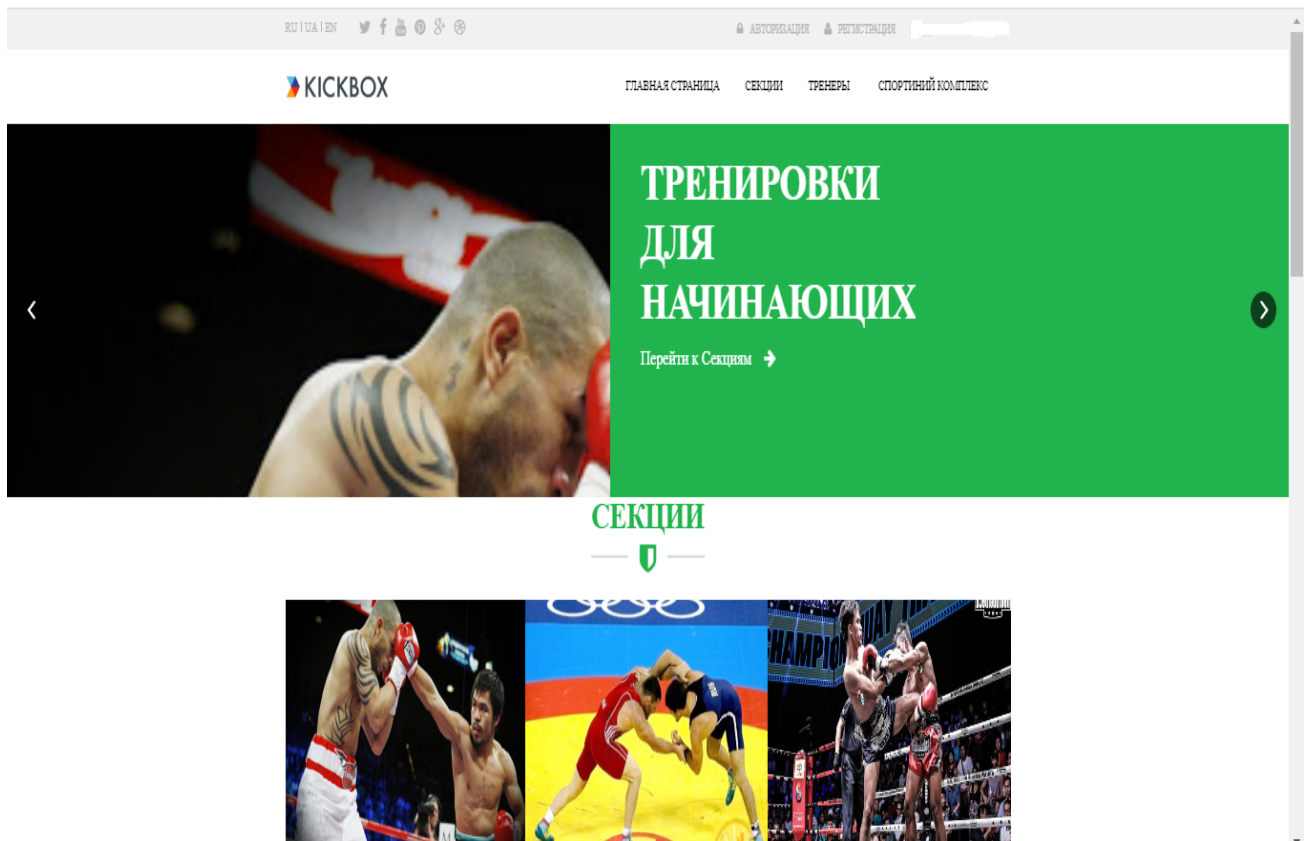



Рисунок 3.4 – Макет головної сторінки веб-додатку

Також на макеті головної сторінки знаходиться розклад занять для усіх напрямків, які викладаються у спортивному комплексі. На рисунку 3.4 відображено макет розкладу занять .



РАСПИСАНИЕ

⌵

Время	Понедельник	Вторник	Среда	Четверг	Пятница	Суббота	Воскресенье
06:00-07:00 AM	БОКС	САМБО		ТАЙСКИЙ БОКС	САМБО	ТАЙСКИЙ БОКС	
08:00-09:00 AM	ВОЛЬНАЯ БОРЬБА		ВОЛЬНАЯ БОРЬБА			БОКС	
09:00-17:00 AM	БОКС		САМБО			САМБО	
09:00-07:00 AM			ВОЛЬНАЯ БОРЬБА	ТАЙСКИЙ БОКС	ДЖУДО	ДЖУДО	
08:00-09:00 AM					ВОЛЬНАЯ БОРЬБА		ВОЛЬНАЯ БОРЬБА
09:00-17:00 AM	ТАЙСКИЙ БОКС			БОКС		ДЖУДО	

Рисунок 3.4 – Макет головної сторінки веб-додатку

У якості прикладу для реалізації рекомендацій що до розробки інформаційних забезпечення систем надання послуг було обрано систему надання професійних навчальних послуг із самооборони . На рисунку 3.6 макет відображення секції.

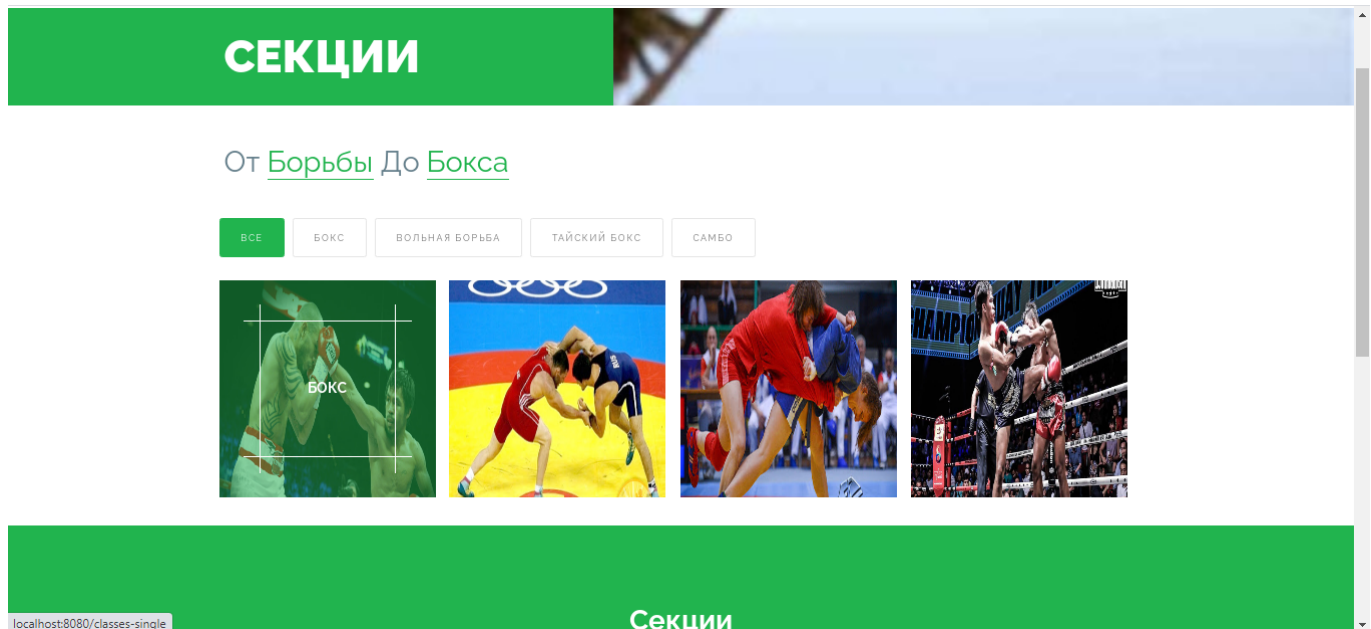


Рисунок 3.6 – Макет списка секцій

Клієнти можуть придбати абонемент який буде включати в себе персональні тренування з тренером, тому розроблена система надає змогу вибрати собі персонального тренера. На рисунку 3.7 макет списку тренерів.

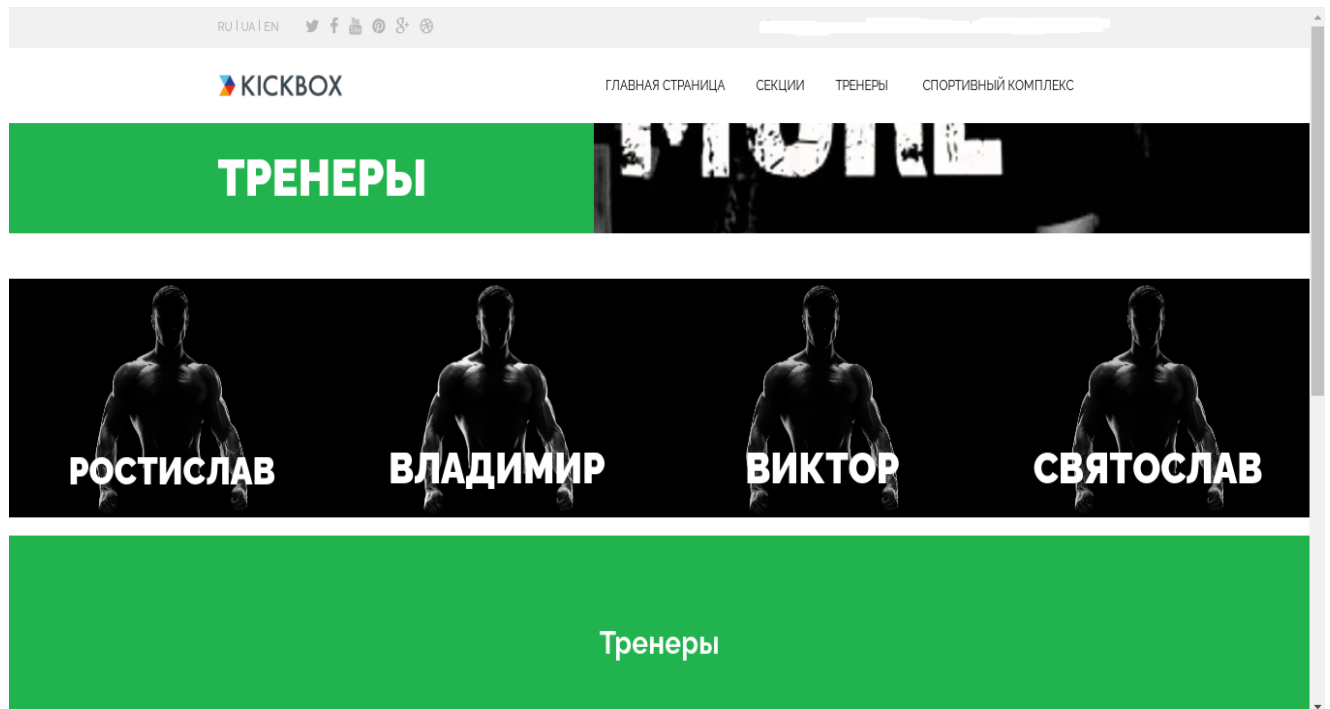


Рисунок 3.7 – Список тренерів

У кожного тренера є своя історія, свої досягнення, і свій напрямок, докладну інформацію, що до досягнень тренера клієнт може отримати в персональній сторінці тренера, а також відразу придбати абонемент який включає послуги тренера. Якщо клієнт має бажання займатись у групі він має можливість придбати абонемент який включає групові заняття з напрямку який обере клієнт. На рисунку 3.8. макет детальної інформації, що до тренера.

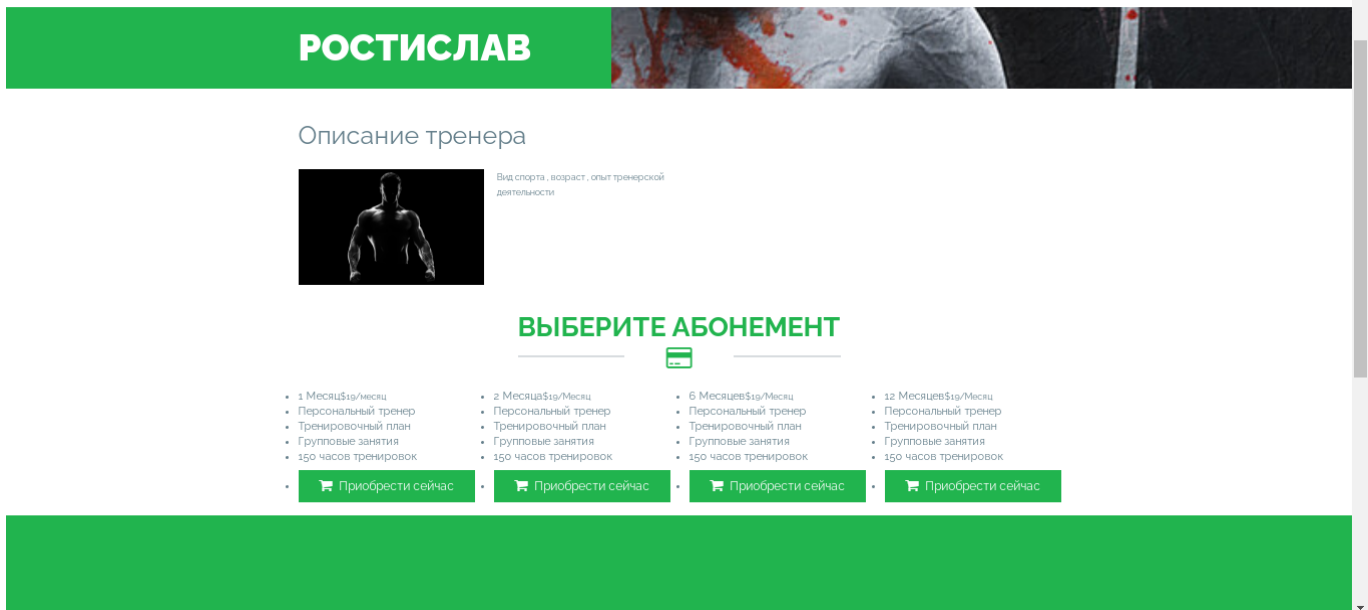


Рисунок 3.8 – Макет детальної інформації про тренера

На рисунку 3.9 макет сторінки групових занять.

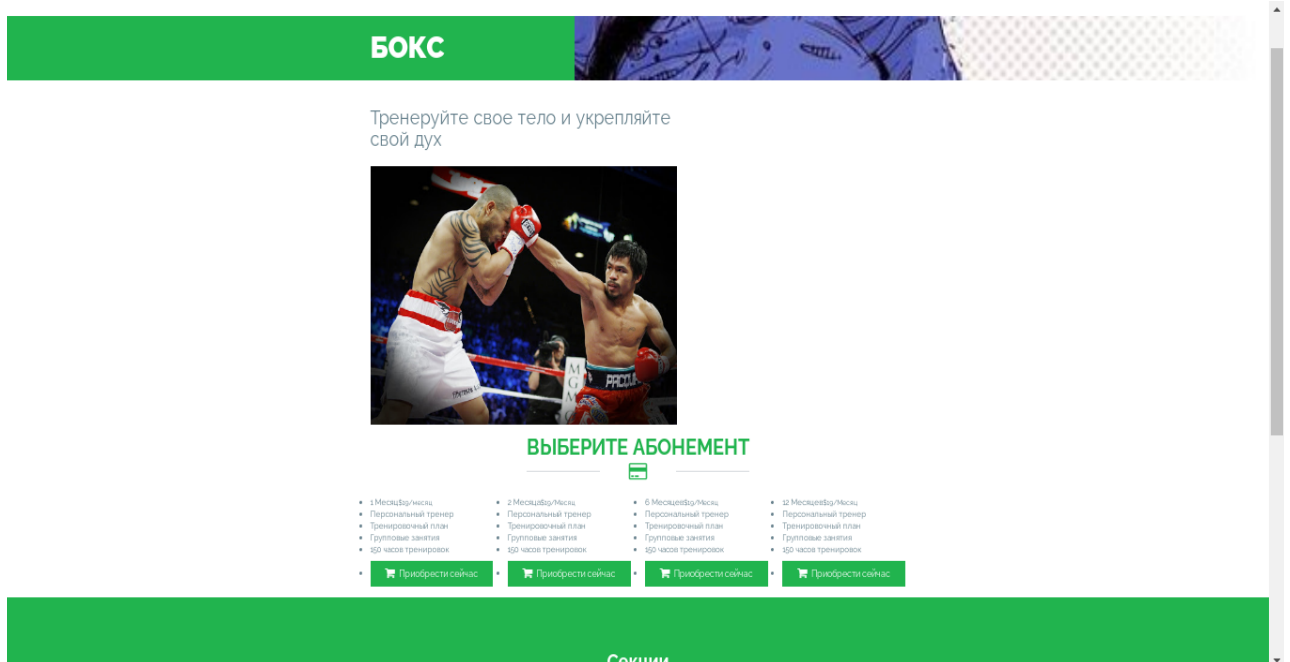


Рисунок 3.9 – Сторінка групових занять

Клієнт має можливість ознайомитись с інформацією, що до закладу який надає послуги. На рисунку 3.10 макет інформації про спортивний комплекс

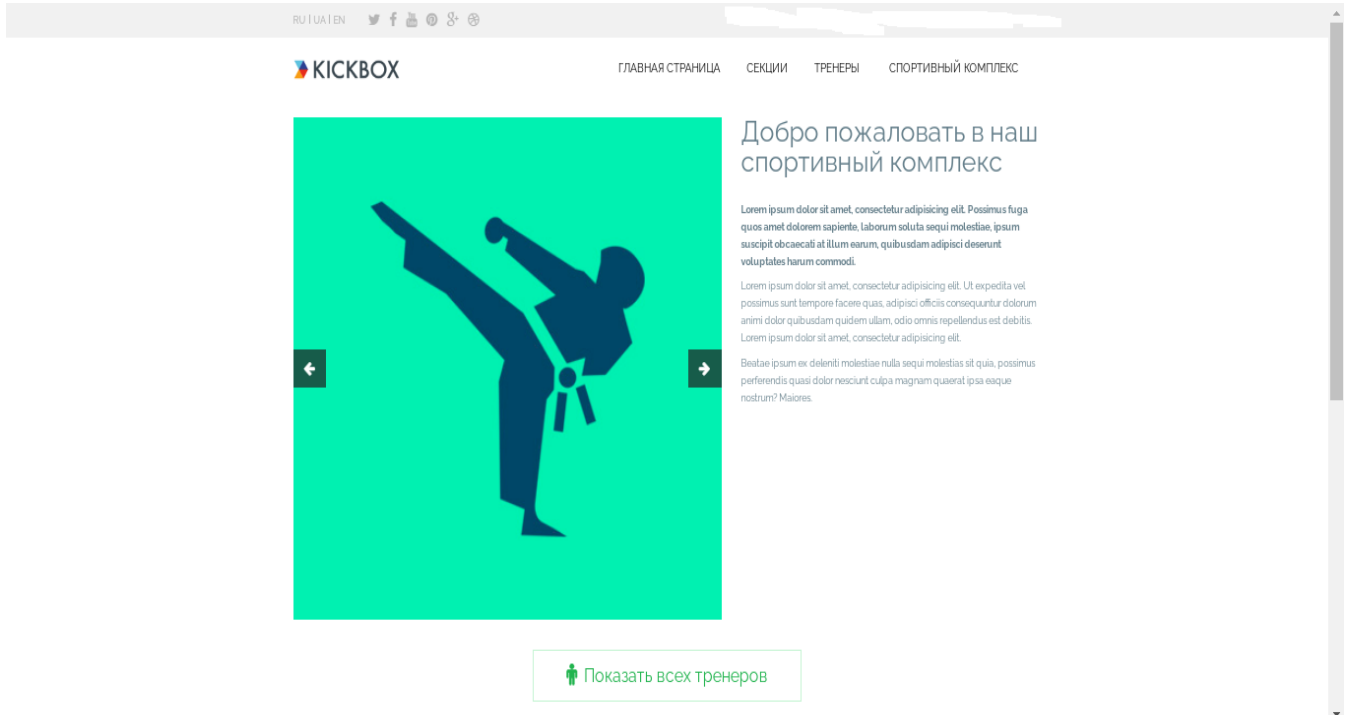


Рисунок 3.10 – Інформація про спортивний комплекс

4. РОЗРОБКА ІНФОРМАЦІЙНОЇ ТЕХНОЛОГІЇ

4.1 Розробка системних вимог до системи надання послуг

Для створення системи надання послуг використовується операційна система сімейства Microsoft Windows, а саме Windows 10. Дана система була обрана через специфіку пристрою на якому проводиться створення інформаційної технології. Однак, приймаючи до факту те, що під інформаційною технологією розуміється створення веб-додатку, тобто застосунок, який працює у браузері та обмеження на операційну систему при її використанні відсутні. Користуватися веб-додатком можна на будь-якому комп'ютерному пристрої, який надає доступ до мережі інтернет. Для реалізації клієнтської частини використовувався інструментарій JavaScript. А саме, бібліотека JQuery для створення користувальницьких інтерфейсів. Разом із бібліотекою JQuery були використані додаткові бібліотеки, які необхідні для коректної її роботи. Для реалізації серверної частини додатку було обрано мову програмування Java, а саме фреймворк Spring.

Дані веб-додатку зберігаються у вигляді бази даних типу MySQL. Для цього використовується реляційна база даних, що означає заздалегідь завдані колонки і таблиці даної БД зв'язані між собою. Дані в БД можна створювати, читати, змінювати та видаляти. Для позначення даних взаємодій можна зустріти аббревіатуру CRUD (Create Read Update Delete). Для запиту даних у БД використовується спеціальна мова SQL (structured query language). Для комфортної роботи з MySQL використано його внутрішній інструмент – Workbench, який надає зручний інтерфейс для взаємодії з БД.

4.2 Опис архітектури розроблюваної системи

Фреймворк Spring Web model-view-controller (MVC) або на нашу модель-уявлення-контролер побудований навколо DispatcherServlet, який розподіляє запити по обробника[1]. У ньому налаштовується меппінг запитів, локалі,

тимчасові зони і багато іншого. Оброблювач за замовчуванням будується на анотаціях `@Controller` і `@RequestMapping`, які надають широкий набір гнучких методів для обробки запитів. Після версії Spring 3.0. механізм `@Controller` так само дозволяє створювати RESTful веб сайти і додатки, використовуючи анотацію `@PathVariable` і інші можливості.

В Spring Web MVC ви можете використовувати будь-який об'єкт в якості команди або об'єкта зі зворотним зв'язком; вам немає необхідності реалізовувати будь-якої спеціальний інтерфейс фреймворка або базовий клас. Зв'язування даних в Spring є дуже гнучким: наприклад, воно розглядає невідповідність типів як помилки валідації і тому це може бути оброблено в додатку, а не в якості системних помилок. Таким чином, вам не потрібно дублювати властивості бізнес-об'єктів, як простих нетипізований рядків для ваших об'єктів форм. Тому можна легко обробляти неправильні підтвердження (сабміта) або правильно конвертувати їх в рядки. Замість цього, бажано пов'язувати такі об'єкти безпосередньо з об'єктами бізнес логіки.

Spring MVC побудований навколо центрального сервлету, який розподіляє запити по контролерам, а також надає інші широкі можливості при розробці веб додатків[11]. Насправді `DispatcherServlet` - повністю інтегрований сервлет в Spring IoC контейнер і таким чином отримує доступ до всіх можливостей Spring. Архітектура розроблюваного додатку подана на рисунку 4.1.

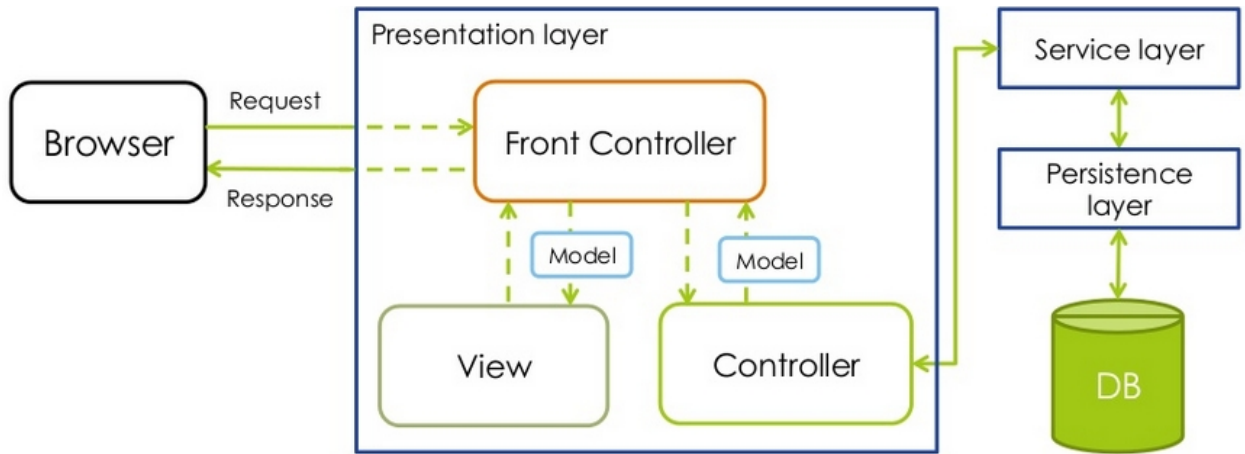


Рисунок 4.1 – Архітектура розробленої системи

4.2 Розробка функціональних вимог

Для визначання функціональних вимог до інформаційної системи, була розроблена її функціональна модель з використанням стандарту IDEF0. Відповідно до стандарту IDEF0 будь-який процес може бути описаний у вигляді блоку, який містить входи та виходи. Процес полягає в перетворенні входів виходи під впливом управління та за наявності необхідних ресурсів. Виходи цього процесу згодом можуть бути або введенням для наступного процесу, або ресурсами, або засобами управління.

На рисунку 4.2 представлена концептуальна діаграма.

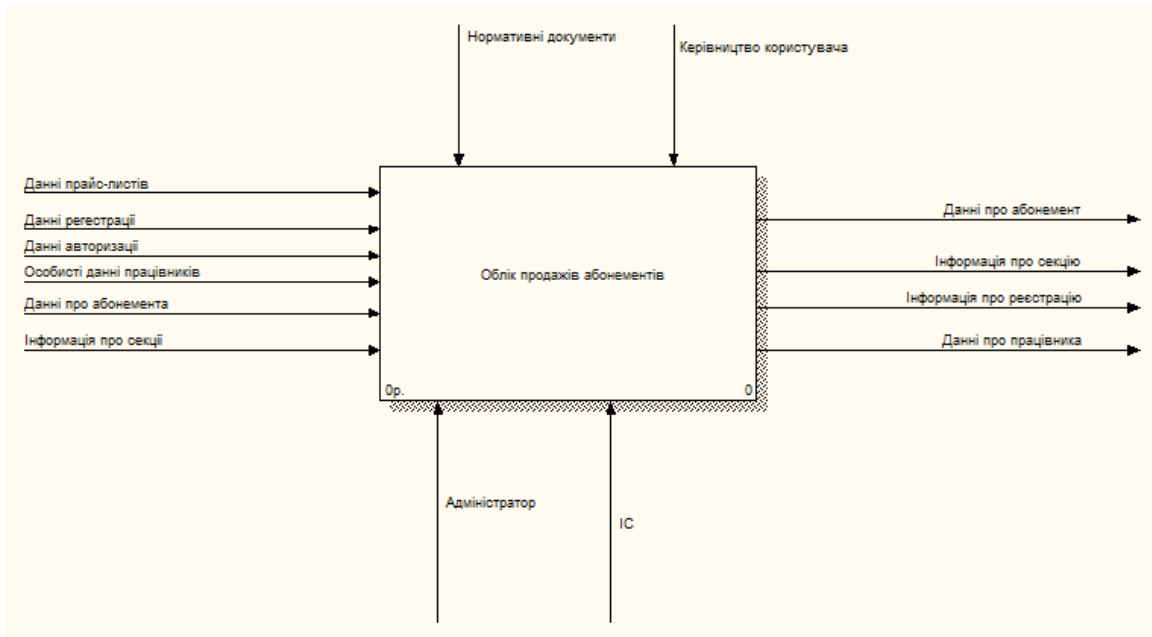


Рисунок 4.2 – Концептуальна діаграма інформаційної системи в нотації IDEF0

На вхід подаються дані користувача прайс-листи , інформації про секції. Стрілки управління це нормативні документи та керівництво користувача. На виході отримуємо данні про абонемент , дані про реєстрацію користувачів , данні про працівників.. У ролі механізму виступають: адміністратор та інформаційна система.

На рисунку 4.3 представлена декомпозиція головної функції системи – «Облік продажів абонементів ». Ця декомпозиція дозволяє краще деталізувати головну функцію. На діаграмі зображено функції:

- авторизація або реєстрація незареєстрованого користувача у системі, на цьому етапі користувач зможе зареєструватися як адміністратор так и клієнт ;
- адміністрування системи, надає можливість управління налаштуваннями системи;

На вхід функції реєстрації/авторизації подається інформація про користувача а на виході отримуємо акаунт користувача.

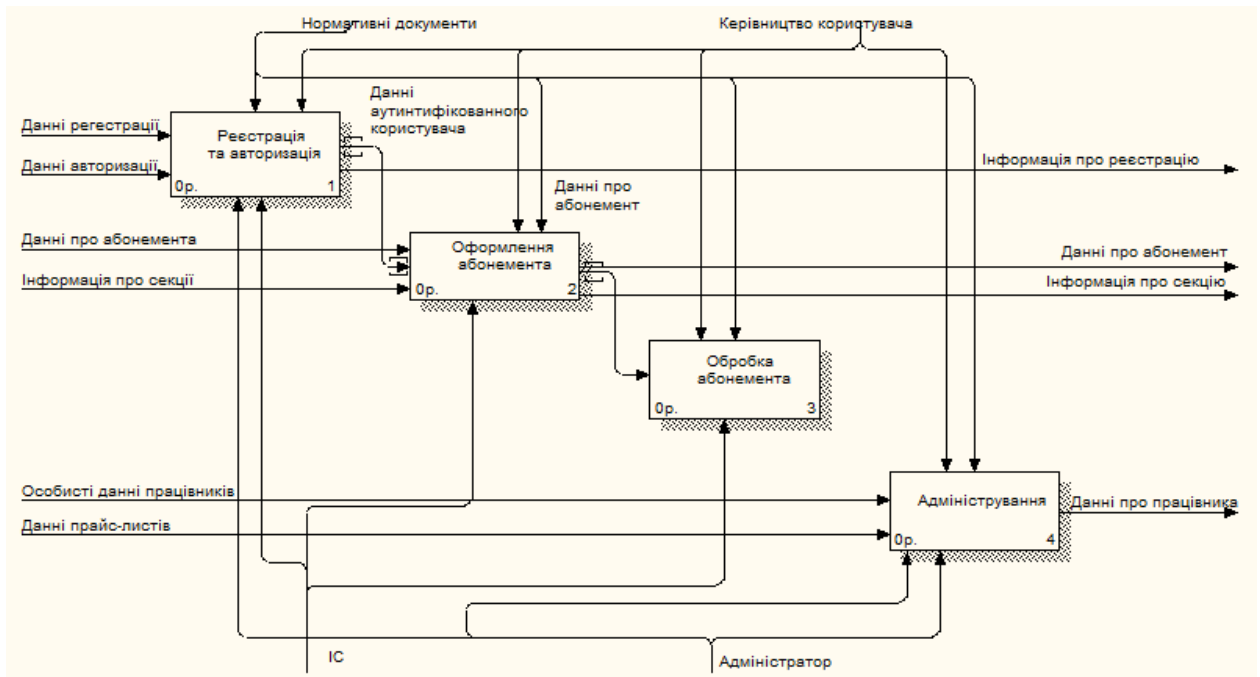


Рисунок 4.3 – Діаграма декомпозиції головної функції системи в нотації IDEF0

На рисунку 4.4 представлена декомпозиція функції «Реєстрація та авторизація» системи.

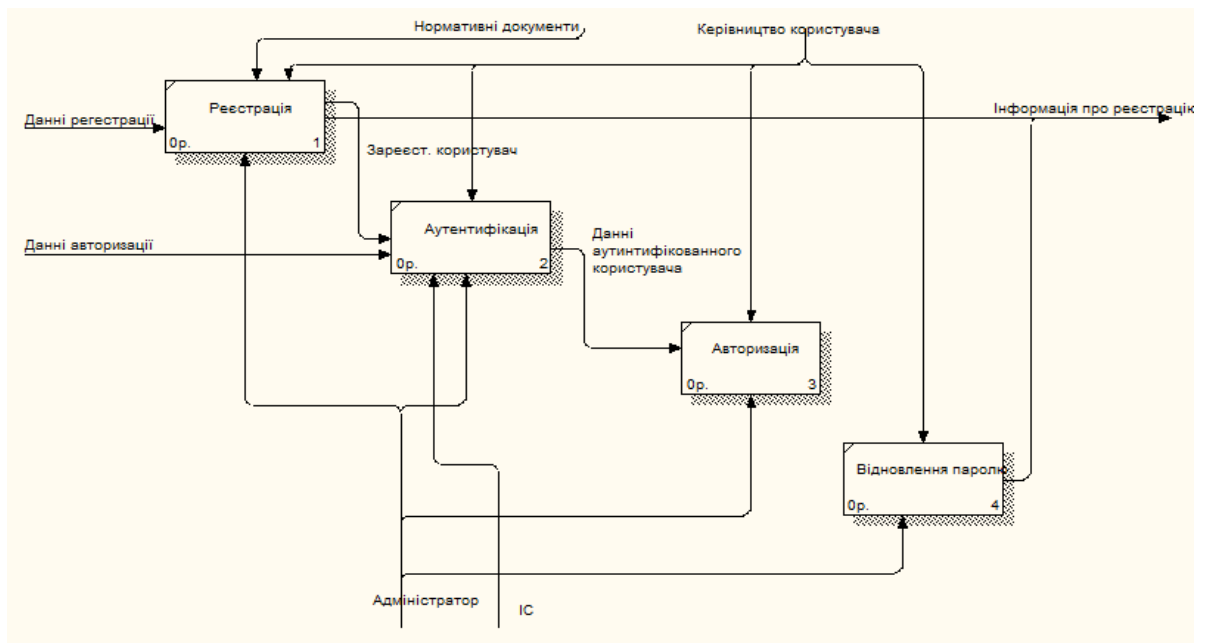


Рисунок 4.4 – Діаграма декомпозиції функції «Реєстрація та авторизація» системи

Ця функція необхідна для користувачам системи для авторизації . Ця діаграма показує, що користувач може:

- пройти реєстрацію;
- відновити пароль.

На вхід функції реєстрації подається інформацію яку ввів користувач. На виході отримуємо зареєстрованого користувача .

4.3 Розробка діаграми варіантів використання

Діаграма випадків використання розглядається для аналізу вимог високого рівня в системі. Коли аналізуються вимоги системи, функціональні можливості фіксуються у випадках використання. Можна сказати, що випадки використання – це не що інше, як функціональні можливості системи. Вікно системи відображається лише на діаграмі верхнього рівня і має містити овали використання, по одному для кожної послуги вищого рівня, які надає інформаційна система акторам, тобто користувачам системи. Будь-яка внутрішня поведінка, яку може мати система, яка використовується лише частинами системи, не повинна відобразитись у вікні системи .

Аби детально розглянути і виділити хто і як може використовувати систему, яка розробляється та які функції може виконувати система в залежності від ролі користувача, розробимо діаграму Use Case[5].

На рисунку 4.5 представлена діаграма варіантів використання, яка описує функціональність системи з точки зору користувачів.

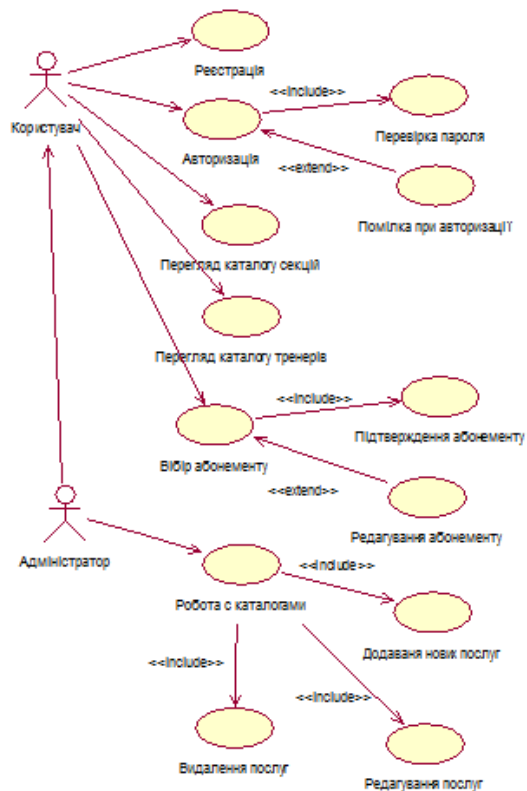


Рисунок 4.5 – Діаграма варіантів використання системи (Use Case)

На даній діаграмі можна побачити двох акторів:

- користувач – клієнт, користувач, який може придбати абонемент;
- адміністратор, який керує каталогом.

Дії які може виконувати учень:

- реєстрація та авторизація;
- перегляд інформації що до спортивного клубу;
- придбати абонемент.

Дії які може виконувати адміністратор:

- додавати, видаляти, або редагувати вже існуючі послуги;
- редагувати інформацію що до спортивного клубу;
- налаштовувати систему, за власними вподобаннями;
- створювати звіт.

Таким чином було детально представлено функціональні можливості системи в залежності від ролі користувача

ВИСНОВКИ

В рамках кваліфікаційної роботи проведенний аналіз різноматнітних систем надання послуг, були знайдені похибки в системах, завдяку цьому вдалось створити універсальні рекомендації щодо створення інформаційного забезпечення систем надання послуг, використовувачи створенні рекомендації був створений веб-додаток для системи надання професійних навчальних послуг із самооборони , який втілює у собі рекомендації описані у кваліфікаційній роботі.

Результати розробки є продукт, що дозволяє підвищити якість взаємодії з клієнтами і підвищити імідж організації, а також надає можливість зменшити витрати часу на роботу з потенційним клієнтом.

В процесі розробки рекомендацій були вивчені проблеми – сфери послуг, проаналізовано необхідність впровадження запропонованих рекомендацій , погоджено засіб розробки, розроблений сайт, який повністю відповідає рекомендаціям.

Веб-додаток який розроблений с урахуванням розроблених рекомендацій дає можливість підприємству вийти на високий рівень, який згодом окупиться і призведе до позитивний ефект з точки зору збільшення оборотів і прибутку підприємства за рахунок підвищеного попиту в геометричній прогресії.

В ході виконання випускної кваліфікаційної роботи були вирішені наступні завдання:

- проведено аналіз сфери послуг в цілом, а також різноманітних систем надання послуг;
- розроблені рекомендації що до створення інформаційного забезпечення систем надання послуг;
- розроблені рекомендації були втілені у життя на прикладі системи надання професійних навчальних послуг із самооборони;
- реалізований веб-додаток;
- сайт був наповнений контентом і протестований.

Використання розробленого веб-сайту для тренувального центру спрощує процеси просування компанії, ведення та видачі інформації, що призводить до значного зниження трудових витрат, забезпечує скорочення часу обробки інформації, підвищує достовірність видаваної інформації. На веб-сайті тренувального центру так само розроблений необхідний функціонал:

- особистий кабінет;
- оформлення замовлення абонементу;
- онлайн запис на заняття;
- інструменти для зворотного зв'язку;
- розробити кілька розділів з докладною інформацією про послуги.

Для кращого розуміння функціональних можливостей системи, були розроблені UML діаграми та діаграма IDEF0 з основною бізнес функцією та декомпозицією цієї функції. На кожній з діаграм продемонстровано як користувачі можуть взаємодіяти з інформаційною системою в залежності від ролі.

Таким чином, слід вважати, що завдання випускної кваліфікаційної роботи повністю виконані і мета досягнута.

ПЕРЕЛІК ДЖЕРЕЛ ПОСИЛАННЯ

1. Технічна документація Spring Boot : <https://spring.io/projects/spring-boot>.
2. Matthew B. Doar. Practical JIRA Administration. United States of America, 2012.
3. McGowan C. IDEF0 and SADT: A Modeler's Guide / Clement McGowan., 2005. – 392 с.
4. Alistair Cockburn. Writing Effective Use Cases / Alistair Cockburn, 2001. – 259 с.
5. UML Class Diagram Relationships [Електронний ресурс] – Режим доступу до ресурсу: <https://creately.com/blog/diagrams/class-diagram-relationships/>.
6. Ghinwa Jalloul. UML by Example / Ghinwa Jalloul, 2004. – 243с.
7. Діаграма станів [Електронний ресурс] – Режим доступу до ресурсу: <http://flash.retejo.info/cxefpagxo/uml/diagrama-staniv>.
8. Майк Кон. Scrum: гибкая разработка ПО – Succeeding with Agile: Software Development Using Scrum (Addison-Wesley Signature Series). – М.: «Вильямс», 2011. – С. 576.
9. J. Welles Wilder, Jr. — New Concepts in Technical Trading Systems — Trend Research. P.O. BOX 450. Greensboro, N.C. 27402. June 1978. 130 p.
10. Гайдамакин Н.А. Автоматизированные информационные системы, базы и банки знаний, 2002. – М.: Гелиос. – 368 с.
11. Spring in Action, Fifth Edition Craig Walls October 2018 ISBN 9781617294945 520 pages
12. Spring Boot in Action Craig Walls Foreword by Andrew Glover December 2015 ISBN 9781617292545 264 pages
13. Reactive Spring Paperback – September 10, 2020
14. Spring Security in Action Laurențiu Spilcă October 2020 ISBN 9781617297731 560 pages

15. Professional Java Development with the Spring Framework Rod Johnson, Jürgen Höller, Alef Arendsen, Thomas Risberg, Colin Sampaleanu ISBN: 978-0-764-57483-2 July 2005 676 Pages

16. Grebennik I., Hutsa O., Petrova R., Yelchaninov D., Morozova A. (2021) Information Technology Based on Qualitative Methods in Cyber-Physical Systems of Situational Disaster Risk Management. In: Murayama Y., Velev D., Zlateva P. (eds) Information Technology in Disaster Risk Reduction. ITDRR 2020. IFIP Advances in Information and Communication Technology, vol 622. Springer, Cham. https://doi.org/10.1007/978-3-030-81469-4_11

17. Сервісна діяльність <http://www.mainmarketing.ru/mcob-731.html>