

МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ
ОДЕСЬКИЙ ДЕРЖАВНИЙ ЕКОЛОГІЧНИЙ УНІВЕРСИТЕТ

Факультет комп'ютерних наук,
управління та адміністрування
Кафедра інформаційних
технологій

Кваліфікаційна робота бакалавра

на тему: «Розробка web-застосунку для управління замовленнями
на платформі ASP.NET»

Виконав студент групи К-19і
спеціальності 122 Комп'ютерні науки
Осадченко Олександр
Олександрович

Керівник к.т.н., доцент
Фразе-Фразенко Олексій
Олексійович

Консультант _____

Рецензент Домаскін Олег
Михайлович, Начальник ЦІТ ОНЕУ

ЗМІСТ

СКОРОЧЕННЯ ТА УМОВНІ ПОЗНАКИ	5
ВСТУП	6
1 АНАЛІЗ ПРЕДМЕТНОЇ ОБЛАСТІ І ПОСТАНОВА ЗАВДАННЯ.....	7
1.1 Бази даних	7
1.2 Системи керування базами даних	122
1.3 Веб-сервіси.....	144
1.4 Опис предметної області	18
1.5 Функціональні вимоги	19
1.6 Не функціональні вимоги	19
2 ВИБІР ПРОГРАМНИХ ЗАСОБІВ	21
2.1. Обґрунтування вибору платформи для програмування.....	21
2.2 Вибір середовища розробки.....	29
2.3 Принципи та підходи розробки інтерфейсів	30
3 ПРОЕКТУВАННЯ ІНФОРМАЦІЙНОЇ СИСТЕМИ	33
3.1 Проектування інформаційної системи за допомогою методології функціонального моделювання SADT.....	33
3.2 Проектування інформаційної системи за допомогою методології Workflow Diagramming	34
3.3 Структура БД.....	35
3.4 Реалізація функціональних можливостей.....	37
4 ПРАКТИЧНА РЕАЛІЗАЦІЯ ТЕХНІЧНОГО ПРОЕКТУ	41
4.1 Схема сайту.....	41
4.2 Опис роботи програми.....	42
4.3 Тестування web-додатку.....	47
ВИСНОВКИ.....	48
СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ	49

СКОРОЧЕННЯ ТА УМОВНІ ПОЗНАКИ

БД – база даних.

SQL – structured query language.

СКБД – система керування базою даних.

IBM – international business machines.

CLI – common Language Infrastructure.

HTML – hyperText Markup Language.

PHP – PHP: Hypertext Preprocessor.

JS – java script.

ІС – інформаційна система.

CRUD – create, read, update, delete.

SADT – Structured Analysis and Design Technique – методологія структурного аналізу і проектування.

GUI – Graphical User Interface – графічний користувацький інтерфейс.

JVM - Java Virtual Machine – віртуальна машина Java.

ВСТУП

Стрімкий розвиток інформаційних технологій зобов'язує дотримуватися їх, щоб бути в курсі того, що відбувається навколо. Спочатку радіо, потім телебачення, а тепер глобальна мережа Інтернет відіграють роль джерела інформації. Причому Інтернет випередив і радіо, і телебачення. Нові сайти по різних тематик з'являються щодня. Такі специфічні професії, як Web-дизайнер, Web-програміст, верстальник, затребувані на ринку праці, що говорить про популярність Інтернету.

Актуальність виконаної роботи очевидна. В даний час у кожній великій фірми, у деяких найбільш успішних середніх і малих фірм є свої Інтернет-ресурси, за допомогою яких вони надають інформацію про себе та свою діяльність. Подібно засобам масової інформації Інтернет-ресурси доносять інформацію до зацікавлених осіб, але вартість донесення інформації до зацікавлених осіб виходить нижче, а часом і швидше. Якщо зайти на пошукову систему Google і в вікні пошуку написати доставка їжі система знайде тисячі сайтів. Із цього випливає, що багато компаній, що працюють в цій сфері вже мають власні сайти. Значить і фірмі FoodStore, що займає одне з лідируючих місць на споживчому ринку, необхідно мати власний Інтернет-ресурс.

Компанія FoodStore займається продажем різноманітних страв та їх доставкою до адресату. Необхідно відразу помітити, що діяльність будь-якої компанії в сучасній економіці не є статичною, тобто компанія може перестати продавати будь-які продукти, може навпаки продавати нові продукти і надавати нові послуги. Все це говорить про те, що обов'язково треба продумати можливість додавання нових видів товару і видалення застарілих даних. Таким чином, головною метою випускної кваліфікаційної роботи є розробка інформаційно-програмного забезпечення автоматизації обслуговування клієнтів для FoodStore.

1 АНАЛІЗ ПРЕДМЕТНОЇ ОБЛАСТІ І ПОСТАНОВА ЗАВДАННЯ

1.1 Бази даних

1.1.1 Поняття бази даних

База даних – сукупність даних, організованих відповідно до концепції, яка описує характеристику цих даних і взаємозв'язки між їх елементами; ця сукупність підтримує щонайменше одну з областей застосування (за стандартом ISO/IEC 2382:2015). В загальному випадку база даних містить схеми, таблиці, подання, збережені процедури та інші об'єкти. Дані у базі організовують відповідно до моделі організації даних. Таким чином, сучасна база даних, крім саме даних, містить їх опис та може містити засоби для їх обробки.

В загальному випадку базою даних можна вважати будь-який впорядкований набір даних. Наприклад, паперову картотеку з формулярами про працівників підприємства у відділі кадрів. Але дана стаття зосереджена на використанні баз даних в інформаційних системах. На даний час застосунки для роботи з базами даних є одними з найпоширеніших прикладних програм.

В даний час існує декілька типів організацій баз даних, які поділяються, перш за все, ставленням до управління одиницями записів.

Модель ієрархічної бази даних є найстарішою моделлю бази даних і представлена у вигляді ієрархічної деревоподібної структури.

Ієрархічна деревоподібна структура робить доступ до читання надзвичайно швидким. Недолік деревовидних посилань полягає в тому, що вони самі і зв'язку між ними повинні бути прописані спочатку.

За основними принципами, мережева БД схожа на ієрархічну модель, але дозволяє створювати різнобічні вузли.

При з'єднанні вузлів один з одним, створюється мережа, через яку користувач може більш безпосередньо отримати доступ до шуканої записи даних. Мережеві бази даних добре підходять для відображення таких

структур, як географічного розташування, семантичних мереж, персональних мереж і мереж компаній.

У цьому типі БД інформація зберігається в стосунках (таблицях), які все взаємно зв'язуються. Оптимізація даними в таблиці здійснюється за допомогою скорочення рядків і / або стовпців. Запис будь-якої інформації є рядком і містить всі поля даних в стовпчиках. Цей тип БД є найбільш вживаною в бізнес-рішеннях.

У цьому різновиді БД застосовується принцип зберігання інформації разом з їх функціями в об'єкті. Вона використовує парадигми ООП і, зокрема, написання на відповідній мові (таких як, C ++, Java або C #).

Існують також об'єктні БД, функціональні, об'єктно-реляційні, які поєднують властивості перерахованих вище типів.

Автономні бази даних

Це новітній тип організації БД, в якому використовуються принципи машинного навчання для управління структурою даних. Це самоврядні БД, які можуть підтримувати повну функціональність без контролю адміністраторів і сам відновлювати втрачені або пошкоджені ресурси.

Еволюція баз даних

Принципи створення баз даних зазнали серйозних змін, з тих пір, як в 1962 році була реалізована перша з них. Спочатку застосовувалися навігаційні БД, до яких відносяться типи з деревовидної і мережевою структурою.

З плином часу реляційні бази даних замінили мережеві БД, які тепер використовуються тільки для спеціальних завдань.

Графові системи, в свою чергу, є свого роду подальшим розвитком реляційних БД. Вони використовують графіки для відображення та зберігання мережевої інформації.

Протягом декількох десятиліть на ринку домінували три основні виробники СУБД (RDBMS): Oracle, IBM і Microsoft. Однак поточна динаміка ринку привела до змін, які привели в рух ринок і створили більш широку конкуренцію.

За інформацією на перший квартал 2020 року, в світі налічується більше 250 різних продуктів, які пропонують реляційні або графові типи організації БД. Причому більше 50 продуктів використовують нові типи – NoSQL, куди відносяться як хмарні рішення, так і БД з автономної структурою і підтримкою машинного інтелекту.

Відмінність між базою даних і електронною таблицею

У програмі Excel ви, наприклад, можете ввести дані в вільно визначається сітку. Вони можуть бути відформатовані за бажанням, наприклад, числа можуть відображатися у вигляді валюти або осередкам може бути призначений колір фону. Існує також концепція формул, які обчислення можуть виконувати і оновлювати при зміні вхідних даних. Це створює певну інтерактивність.

Основний принцип полягає в тому, що дані беруться з так званих областей і зміщуються один щодо одного. Недоліком обробки таблиці є те, що ці області мають бути визначені заздалегідь.

Організація ж баз даних включає в себе широкий спектр термінів, концепцій і програмного забезпечення. У разі баз даних ми говоримо, перш за все, про системи, які реалізують концепцію реляційної алгебри.

Обробка даних в базах даних також проводиться в табличній формі. Різниця в тому, що в базі даних схема даних є фіксованою. Схема відносин точно визначає, в якому форматі які дані повинні бути доступні. Неможливо відформатувати дані. Хоча елементи дизайну доступні при обробці таблиць, бази даних працюють виключно на рівні даних.

Замість цього ви працюєте з так званими запитами. Запит може фільтрувати, обчислювати, групувати і сортувати, і він набагато більш гнучкий, ніж в Excel. На додаток до таблиць в базах даних є так звані уявлення, які мають структуру, аналогічну таблиць, але складаються тільки з обчислених або відфільтрованих даних.

1.1.2 Мова запитів SQL

SQL — декларативна мова програмування для взаємодії користувача з базами даних, що застосовується для формування запитів, оновлення і керування реляційними БД, створення схеми бази даних та її модифікації, системи контролю за доступом до бази даних. Сама по собі SQL не є ані системою керування базами даних, ані окремим програмним продуктом. На відміну від дійсних мов програмування (C або Pascal), SQL може формувати інтерактивні запити або, будучи вбудованою в прикладні програми, виступати як інструкції для керування даними. Окрім цього, стандарт SQL містить функції для визначення зміни, перевірки та захисту даних.

SQL — це діалогова мова програмування для здійснення запиту і внесення змін до бази даних, а також керування базами даних. Багато баз даних підтримує SQL з розширеннями до стандартної мови. Ядро SQL формує командна мова, яка дозволяє здійснювати пошук, вставку, оновлення і вилучення даних за допомогою використання системи керування і адміністративних функцій. SQL також включає CLI (Call Level Interface) для доступу і керування базами даних дистанційно.

SQL – це діалогова мова програмування для здійснення запиту і внесення змін до бази даних, а також управління базами даних. Багато баз даних підтримує SQL з розширеннями до стандартної мови.

Ядро SQL формує командна мова, яка дозволяє здійснювати пошук, вставку, оновлення, і вилучення даних, використовуючи систему управління і адміністративні функції. SQL також включає CLI (Call Level Interface) для доступу і управління базами даних дистанційно.

Переваги SQL:

– Незалежність від конкретної СУБД. Незважаючи на наявність діалектів і відмінностей в синтаксисі, в більшості своїй тексти SQL-запитів, що містять, DDL і DML, можуть бути досить легко перенесені з однієї СУБД в іншу. Існують системи, розробники яких спочатку закладалися на

застосування щонайменше кількох СУБД (наприклад: система електронного документообігу Documentum може працювати як з Oracle, так і з Microsoft SQL Server та IBM DB2). Природно, що при застосуванні деяких специфічних для реалізації можливостей такої переносимості добитися вже дуже важко.

– Наявність стандартів. Наявність стандартів і набору тестів для виявлення сумісності також відповідності конкретній реалізації загальноприйнятому SQL стандарту тільки сприяє «стабілізації» мови. Правда, варто звернути увагу, що сам по собі стандарт місцями занадто формалізований і роздутий в розмірах, наприклад, Core-частину стандарту SQL:2003 включає понад 1300 сторінок тексту.

– Декларативність. За допомогою SQL програміст описує тільки те, які дані потрібно витягнути або модифікувати. Те, яким чином це зробити, вирішує СУБД безпосередньо при обробці SQL-запиту. Проте не варто думати, що це повністю універсальний принцип – програміст описує набір даних для вибірки або модифікації, проте йому при цьому корисно уявляти, як СУБД розбиратиме текст його запиту. Особливо критичні такі моменти стають при роботі з великими базами даних і зі складними запитамі – чим сконструйований складніше запит, тим більше він допускає варіантів написання, різних за швидкістю виконання, але тих самих за набором даних.

Недоліки SQL:

– Складність. Хоча SQL і замислювався, як засіб роботи кінцевого користувача, врешті-решт він став настільки складним, що перетворився на інструмент програміста.

– Складність роботи з ієрархічними структурами. Раніше SQL не пропонував стандартного способу маніпуляції деревовидними структурами. Деякі постачальники СУБД пропонували свої рішення. Наприклад, Oracle використовує вираз CONNECT BY. В наш час як стандарт прийнята рекурсивна конструкція WITH.

Як і з багатьма стандартами в IT-індустрії, з мовою SQL виникла проблема: багато розробників ПЗ вирішили, що функціональність стандартної

версії недостатня (що, в принципі, для ранніх версій SQL було певною мірою справедливо) і їх бажанням стало її розширення. Це призвело до того, що у різних виробників СУБД застосовуються різні діалекти SQL, здебільшого, несумісні між собою.

До 1996 року питаннями відповідності комерційних реалізацій SQL стандарту займався переважно інститут NIST, який і встановлював рівень відповідності стандарту. Відтоді, як останній підрозділ, що займався СУБД, був розформований, всі зусилля з перевірки СУБД на відповідність стандарту були покладені на її виробника.

Вперше поняття «Рівня відповідності» було запропоноване в стандарті SQL-92. ANSI і NIST визначали чотири рівні відповідності реалізації цьому стандарту:

- Entry (базовий)
- Transitional (перехідний) — перевірку на відповідність цьому рівню проводив тільки інститут NIST
- Intermediate (проміжний)
- Full (повний)

Легко можна зрозуміти, що кожен наступний рівень відповідності свідомо передбачав відповідність попередньому рівню. Далі, згідно з цими рівнями стандартів, будь-яка СУБД, що відповідала рівню Entry, могла бути охарактеризована як «SQL-92-відповідна», хоча насправді її здатність до перенесення і відповідність стандарту обмежувалися набором можливостей, що входять до вказаного рівня.

1.2 Системи керування базами даних

У сучасних інформаційних системах для забезпечення роботи з базами даних використовують системи керування базами даних (СКБД). Система керування базами даних — це система, заснована на програмних та технічних засобах, яка забезпечує визначення, створення, маніпулювання, контроль,

керування та використання баз даних (за стандартом ISO/IEC 2382:2015). Застосунки для роботи з базою даних можуть бути частиною СКБД або автономними. Найпопулярнішими СКБД є MySQL, PostgreSQL, Microsoft SQL Server, Oracle, Sybase, Interbase, Firebird та IBM DB2. СКБД дозволяють ефективно працювати з базами даних, обсяг яких робить неможливим їх ручне опрацювання.

Через тісний зв'язок баз даних з СКБД під терміном «база даних» інколи необґрунтовано та неточно мають на увазі систему керування базами даних. Але варто розрізняти базу даних — сховище даних, та СКБД — засоби для роботи з базою даних. СКБД з інформаційної системи може бути видалена, але база даних продовжить існувати. І навпаки: СКБД може функціонувати без жодної бази даних.

У сучасних інформаційних системах для забезпечення роботи з базами даних використовують системи керування базами даних (СКБД). Система керування базами даних — це система, заснована на програмних та технічних засобах, яка забезпечує визначення, створення, маніпулювання, контроль, керування та використання баз даних (за стандартом ISO/IEC 2382:2015). Застосунки для роботи з базою даних можуть бути частиною СКБД або автономними. Найпопулярнішими СКБД є MySQL, PostgreSQL, Microsoft SQL Server, Oracle, Sybase, Interbase, Firebird та IBM DB2. СКБД дозволяють ефективно працювати з базами даних, обсяг яких робить неможливим їх ручне опрацювання.

Через тісний зв'язок баз даних з СКБД під терміном «база даних» інколи необґрунтовано та неточно мають на увазі систему керування базами даних. Але варто розрізняти базу даних — сховище даних, та СКБД — засоби для роботи з базою даних. СКБД з інформаційної системи може бути видалена, але база даних продовжить існувати. І навпаки: СКБД може функціонувати без жодної бази даних.

У сучасних інформаційних системах для забезпечення роботи з базами даних використовують системи керування базами даних (СКБД). Система

керування базами даних — це система, заснована на програмних та технічних засобах, яка забезпечує визначення, створення, маніпулювання, контроль, керування та використання баз даних (за стандартом ISO/IEC 2382:2015). Застосунки для роботи з базою даних можуть бути частиною СКБД або автономними. Найпопулярнішими СКБД є MySQL, PostgreSQL, Microsoft SQL Server, Oracle, Sybase, Interbase, Firebird та IBM DB2. СКБД дозволяють ефективно працювати з базами даних, обсяг яких робить неможливим їх ручне опрацювання.

Через тісний зв'язок баз даних з СКБД під терміном «база даних» інколи необґрунтовано та неточно мають на увазі систему керування базами даних. Але варто розрізняти базу даних — сховище даних, та СКБД — засоби для роботи з базою даних. СКБД з інформаційної системи може бути видалена, але база даних продовжить існувати. І навпаки: СКБД може функціонувати без жодної бази даних.

1.3 Веб-сервіс

Веб-сервіс — ідентифіцируемая унікальним веб-адресою (URL-адресою) програмна система зі стандартизованими інтерфейсами, а також HTML-документ сайту, який відображається браузером користувача.

Веб-служби можуть взаємодіяти один з одним і зі сторонніми додатками за допомогою повідомлень, заснованих на певних протоколах (SOAP, XML-RPC і т. Д.) І угодах (REST). Веб-служба є одиницею модульності при використанні сервіс-орієнтованої архітектури додатку.

У побуті веб-сервісами називають послуги, що надаються в Інтернеті. У цьому вживанні термін вимагає уточнення, чи йде мова про пошук, веб-поштою, зберігання документів, файлів, закладок і т. П. Такими веб-сервісами можна користуватися незалежно від комп'ютера, браузера або місця доступу в Інтернет.

Переваги

- Веб-служби забезпечують взаємодію програмних систем незалежно від платформи. Наприклад, Windows-C #-клієнт може обмінюватися даними з Java-сервером, що працює під Linux.
- Веб-служби засновані на базі відкритих стандартів і протоколів. Завдяки використанню XML досягається простота розробки і налагодження веб-служб.
- Використання інтернет-протоколу забезпечує HTTP-взаємодія програмних систем через міжмережевий екран. Це значна перевага, порівняно з такими технологіями, як CORBA, DCOM або Java RMI. З іншого боку, веб-служби не прив'язані намертво до HTTP - можуть використовуватися і інші протоколи.

Недоліки

- Менша продуктивність і більший розмір мережевого трафіку в порівнянні з технологіями RMI, CORBA, DCOM за рахунок використання текстових XML-повідомлень. Однак на деяких веб-серверах можлива настройка стиснення мережевого трафіку.
- Аспекти безпеки. Відповідальні веб-служби повинні використовувати кодування, можливо - вимагати аутентифікації користувача. Чи достатньо тут застосування HTTPS, або кращі такі рішення, як XML Signature, XML Encryption або SAML - має бути вирішено розробником.

1.3.1 Веб-сервер

Веб-сервер — це сервер, що приймає HTTP-запити від клієнтів, зазвичай веб-браузерів, видає їм HTTP-відповіді, зазвичай разом з HTML-сторінкою, зображенням, файлом, медіа-потокком або іншими даними.

Веб-сервером називають як програмне забезпечення, що виконує функції веб-сервера, так і комп'ютер, на якому це програмне забезпечення працює.

Клієнти дістаються веб-сервера за URL-адресою потрібної їм веб-

сторінки або іншого ресурсу.

Додатковими функціями багатьох веб-серверів є:

- Ведення журналу серверу про звернення користувачів до ресурсів
- Автентифікація користувачів
- Підтримка сторінок, що динамічно генеруються
- Підтримка HTTPS для захищених з'єднань з клієнтами

Часто на комп'ютері разом з веб-сервером встановлюється також і поштовий сервер.

Існує багато веб-серверів. Найпоширенішими є:

- NCSA HTTPd — один із перших веб-серверів, розроблений Робертом Маккулом (англ. Robert McCool) та іншими у компанії NCSA.

- Apache HTTP-Server — найпоширеніший у світі веб-сервер з відкритим сирцевим кодом

- IIS — веб-сервер компанії Microsoft, розповсюджується з ОС сімейства Windows NT

lighttpd — open-source веб-сервер.

- Google Web Server — веб-сервер, створений на основі Apache компанією Google.

- Resin — open-source сервер для застосувань java.

- Cherokee — вільний багатоплатформовий веб-сервер, написаний на C.

Клієнти для звернення до веб-серверів можуть використовуватися абсолютно різні пристрої:

- Веб-браузер — найпоширеніший спосіб

- Спеціальне програмне забезпечення може самостійно звертатися до веб-серверів для отримання оновлень або іншої інформації

- Мобільний телефон може дістатися до ресурсів веб-сервера за допомогою протоколу WAP або HTTP

- Інші інтелектуальні пристрої або побутова техніка

1.3.2 Web-додаток

Веб-додаток – клієнт-серверний додаток, в якому клієнт взаємодіє з веб-сервером за допомогою браузера. Логіка веб-додатки розподілена між сервером і клієнтом, зберігання даних здійснюється, переважно, на сервері, обмін інформацією відбувається по мережі. Одним з переваг такого підходу є той факт, що клієнти не залежать від конкретної операційної системи користувача, тому веб-додатки є межплатформенними службами.

Суттєва перевага побудови веб-додатків для підтримки стандартних функцій браузера полягає в тому, що функції повинні виконуватися незалежно від операційної системи даного клієнта. Замість того, щоб писати різні версії для Microsoft Windows, Mac OS X, GNU / Linux та інших операційних систем, додаток створюється один раз для довільно обраної платформи і на ній розгортається. Однак різна реалізація, CSS, DOM і інших специфікацій в браузерах може викликати проблеми при розробці веб-додатків і подальшої підтримки. Крім того, можливість користувача налаштовувати багато параметрів браузера (наприклад, розмір шрифту, кольору, відключення підтримки сценаріїв) може перешкоджати коректної роботи програми.

Використовуються підходи до архітектурного проектування веб - додатків. Веб-додаток складається з клієнтської і серверної частин, тим самим реалізуючи технологію «клієнт-сервер».

Клієнтська частина реалізує користувальницький інтерфейс, формує запити до сервера і обробляє відповіді від нього.

Серверна частина отримує запит від клієнта, виконує обчислення, після цього формує веб-сторінку і відправляє її клієнту через мережу з використанням протоколу HTTP.

Для створення веб-додатків на стороні сервера використовуються різноманітні технології та будь-які мови програмування, здатні здійснювати вивід в стандартну консоль.

В результаті такої універсальності, постійний користувач веб-додатка може абсолютно без проблем працювати з програмою на будь-якому зі своїх пристроїв, починаючи з офісного стаціонарного комп'ютера, закінчуючи планшетом і смартфоном.

1.4 Опис предметної області

Інтернет-магазин - це сайт, що містить докладний каталог товарів з описом і зображенням. Основна відмінність від звичайного інтернет-каталогу полягає в тому, що товари представлені в інтернет-магазині можна не тільки побачити, а й замовити, не встаючи з місця і не перериваючи захоплюючої подорожі по просторах Інтернет.

Існує кілька типів інтернет-магазинів:

- магазин, що торгує певним видом товарів;
- магазин, в якому торгують товарами вузьконаправленої категорії;
- електронні супермаркети
- торговельні мережі інтернет-магазинів,

Для всіх інтернет-магазинів властивий певний обов'язковий набір елементів, таких як:

- Спеціалізований каталог з підрозділами, в яких представлені всі наявні в наявності товари. Зовнішній вигляд каталогу може бути різним - дерево, що випадають або вкладені списки меню.

- Система реєстрації користувача, яка створює для кожного нового клієнта його власну «кошик», в яку можна «покласти» обраний товар і згодом замовити. У міру пересування клієнта по каталогу система також відстежує переваги клієнта, на основі яких в майбутньому може будуватись не лише асортимент магазину, а й структура видачі супутньої інформації каталогу. Наприклад, якщо людина замовив книгу, система може ненав'язливо запропонувати ознайомитись з іншими книги з тієї ж серії або тієї ж тематики

або запропонувати почитати відгуки читачів. Система може «доповісти», що люди, які замовляли подібний товар, також цікавилися супутніми товарами - наприклад, при покупці картини пропонується вибрати рамку для неї.

- Система оплати товару: покупцеві пропонується використовувати різні способи оплати - кредитні карти, електронні гроші, оплата готівкою (кур'єру або при отриманні на пошті).

- Система доставки товару: тут теж широкий вибір можливостей: пересилання по електронній пошті (програмне забезпечення), доставка кур'єрською службою, звичайна пошта.

1.5 Функціональні вимоги

Розроблюване програмне забезпечення повинно виконувати наступні функції:

- Оброблювати запити користувачів;
- Можливість додавання, видалення та редагування бази даних товарів;
- Збереження запитів користувачів у БД;

1.6 Не функціональні вимоги

Нижче перераховані не функціональні вимоги до системи:

- Представляти з себе веб-сервісне застосування. Веб-сервіс повинен бути доступним через будь-який браузер.
- Веб-сервіс має бути адаптивним під різні пристрої: десктопи, планшети, мобільні телефони.
- Інтерфейс користувача повинен бути інтуїтивно зрозумілим для швидкого навчання нових користувачів.
- Система повинна розгортатись на популярних ОС: Windows, Linux, macOS.

- Система повинна розгортатись на процесорах з архітектурою x86 і ARM (RISC).
- Забезпечувати одночасний доступ до системи до 20 учасників без втрати продуктивності, якщо користувач обирає план на платній основі.
- Містити керівництво користувача для розгортання проекту і його налаштування.
- Бути захищеною від несанкціонованого доступу.
- Зберігати усі дані у реляційній базі даних.
- Система повинна бути масштабована і відмовостійка.

2 ВИБІР ПРОГРАМНИХ ЗАСОБІВ

2.1. Обґрунтування вибору платформи для програмування

ASP (англ. Active Server Pages - «активні серверні сторінки») - технологія, запропонована компанією Microsoft в 1996 році для створення Web-додатків. Ця технологія заснована на впровадженні в звичайні веб-сторінки спеціальних елементів управління, що допускають програмне керування.

За своєю суттю, ASP - це технологія динамічного створення сторінок на стороні сервера, яка наблизила проектування і реалізацію Web-додатків до тієї моделі, за якої проектуються і реалізуються звичайні додатки.

Для реалізації програм ASP використовуються мови сценаріїв (VBScript або JScript). Також допускається застосування COM-компонентів.

Технологія ASP розроблена для операційних систем з сімейства Windows NT і функціонує під управлінням веб-сервера Microsoft IIS.

Технологія ASP отримала свій розвиток у вигляді ASP.NET - технології створення веб-додатків, заснованої вже на платформі Microsoft .NET.

ASP.NET (Active Server Pages для .NET) - платформа розробки веб-додатків, до складу якої входять: веб-сервіси, програмна інфраструктура, модель програмування, від компанії Майкрософт. ASP.NET входить до складу платформи .NET Framework і є розвитком старішої технології Microsoft ASP.

ASP.NET зовні багато в чому зберігає схожість із старішою технологією ASP, що дозволяє розробникам відносно легко перейти на ASP.NET. У той же час внутрішній устрій ASP.NET істотно відрізняється від ASP, оскільки вона заснована на платформі .NET і, отже, використовує всі нові можливості, що надаються цією платформою.

Оскільки ASP.NET ґрунтується на Common Language Runtime (CLR), яка є основою всіх додатків Microsoft .NET, розробники можуть писати код

для ASP.NET, використовуючи мови програмування, що входять в комплект .NET Framework (C #, Visual Basic.NET, J # і JScript .NET).

Програмна модель ASP.NET ґрунтується на протоколі HTTP і використовує його правила взаємодії між сервером і браузером. При формуванні сторінки закладена абстрактна програмна модель Web Forms і на ній заснована основна частина реалізації програмного коду.

У ASP.NET, для управління системою аутентифікації користувачів на сайті, був реалізований за рахунок Membership API, який представляє засоби для входу, зберігання і управління обліковими записами користувачів. Архітектура Membership API вибудована таким чином, що дозволяє управляти користувачами зберігаються в різних джерелах: Microsoft SQL Server, Microsoft Active Directory або спеціальне сховище (реалізоване самим, але тоді необхідно самому реалізовувати постачальник, щоб підключитися до узагальненої бази постачальників). У ASP.NET аутентифікація може здійснюється за допомогою форм, або за допомогою Windows через IIS.

Переваги ASP.NET перед ASP

- Компільований код виконується швидше, більшість помилок отлавлюється ще на стадії розробки;
- Значно поліпшена обробка помилок під час виконання запущеної готової програми, з використанням блоків try..catch;
- Призначені для користувача елементи управління (controls) дозволяють виділяти часто використовувані шаблони, такі як меню сайту;
- Використання метафор, вже застосовуються в Windows-додатках, наприклад, таких як елементи управління і події;
- Розширюваний набір елементів управління і бібліотек класів дозволяє швидше розробляти додатки;
- ASP.NET спирається на багатомовні можливості .NET, що дозволяє писати код сторінок на VB.NET, Delphi.NET, Visual C #, J # і т. Д. ;
- Можливість кешування всієї сторінки або її частини для збільшення продуктивності;

- Можливість кешування даних, що використовуються на сторінці;
- Можливість поділу візуальної частини та бізнес-логіки по різних файлах («code behind»);
- Розширювана модель обробки запитів;
- Розширена подієва модель;
- Розширювана модель серверних елементів управління;
- Наявність master-сторінок для завдання шаблонів оформлення сторінок;
- Підтримка CRUD-операцій при роботі з таблицями через GridView;
- Вбудована підтримка AJAX;

PHP (англ. PHP: Hypertext Preprocessor - «PHP: препроцесор гіпертексту»; спочатку PHP / FI (Personal Home Page / Form Interpreter), а пізніше названий Personal Home Page Tools - «Інструменти для створення персональних веб-сторінок») - скриптова мова загального призначення, інтенсивно застосовується для розробки веб-додатків. В даний час підтримується переважною більшістю хостинг-провайдерів і є одним з лідерів серед мов, що застосовуються для створення динамічних веб-сайтів.

Мова і його інтерпретатор (Zend Engine) розробляються групою ентузіастів в рамках проекту з відкритим кодом. Проект поширюється під власною ліцензією, несумісною з GNU GPL.

В області веб-програмування, зокрема серверної частини, PHP - один з популярних сценарних мов (разом з JSP, Perl і мовами, використовуваними в ASP.NET).

Популярність в області побудови веб-сайтів визначається наявністю великого набору вбудованих засобів і додаткових модулів для розробки веб-додатків. Основні з них:

- автоматичне вилучення POST- і GET-параметрів, а також змінних оточення веб-сервера в зумовлені масиви;

- взаємодія з великою кількістю різних систем управління базами даних через додаткові модулі (MySQL, MySQLi, SQLite, PostgreSQL, Oracle Database (OCI8), Microsoft SQL Server, Sybase, ODBC, mSQL, IBM DB2, Cloudscape і Apache Derby, Informix, Ovrimos SQL, Lotus Notes, DB ++, DBM, dBase, DBX, FrontBase, FilePro, Ingres II, SESAM, Firebird і InterBase, Paradox File Access, MaxDB, інтерфейс PDO, Redis);
 - автоматизована відправка HTTP-заголовків;
 - робота з HTTP-авторизацією;
 - робота з cookies і сесіями;
 - робота з локальними і віддаленими файлами, сокетами;
 - обробка файлів, що завантажуються на сервер;
 - робота з XForms.

Java - строго типізований об'єктно-орієнтована мова програмування загального призначення, розроблений компанією Sun Microsystems (в подальшому придбаній компанією Oracle). Розробка ведеться співтовариством, організованим через Java Community Process; мова і основні реалізують його технології поширюються за ліцензією GPL. Права на торговельну марку належать корпорації Oracle.

Програми Java зазвичай транслуються в спеціальний байт-код, тому вони можуть працювати на будь-якій комп'ютерній архітектурі, для якої існує реалізація віртуальної Java-машини.

Програми на Java транслуються в байт-код Java, який виконується віртуальною машиною Java (JVM) - програмою, обробній байтовий код і передавальній інструкції обладнанню як інтерпретатор.

Перевагою подібного способу виконання програм є повна незалежність байт-коду від операційної системи і устаткування, що дозволяє виконувати Java-додатки на будь-якому пристрої, для якого існує відповідна віртуальна машина. Іншою важливою особливістю технології Java є гнучка система безпеки, в рамках якої виконання програми повністю контролюється віртуальною машиною. Будь-які операції, які перевищують встановлені

повноваження програми (наприклад, спроба несанкціонованого доступу до даних або з'єднання з іншим комп'ютером), викликають негайне переривання.

Часто до недоліків концепції віртуальної машини відносять зниження продуктивності. Ряд удосконалень кілька збільшив швидкість виконання програм на Java:

- застосування технології трансляції байт-коду в машинний код безпосередньо під час роботи програми (JIT-технологія) з можливістю збереження версій класу в машинному коді,
- широке застосування переносних орієнтованого коду (native-код) в стандартних бібліотеках,
- апаратні засоби, що забезпечують прискорену обробку байт-коду (наприклад, технологія Jazelle, підтримувана деякими процесорами архітектури ARM).

За даними сайту shootout.alioth.debian.org, для семи різних завдань час виконання на Java становить в середньому в півтора-два рази більше, ніж для C / C ++, в деяких випадках Java швидше, а в окремих випадках в 7 разів повільніше. З іншого боку, для більшості з них споживання пам'яті Java-машиною було в 10-30 разів більше, ніж програмою на C / C ++. Також примітно дослідження, проведене компанією Google, згідно з яким відзначається істотно нижча продуктивність і більше споживання пам'яті в тестових прикладах на Java в порівнянні з аналогічними програмами на C ++.

Ідеї, закладені в концепцію і різні реалізації середовища віртуальної машини Java, надихнули безліч ентузіастів на розширення переліку мов, які могли б бути використані для створення програм, що виконуються на віртуальній машині. Ці ідеї знайшли також вираз в специфікації загальномовна інфраструктури CLI, закладеної в основу платформи .NET компанією Microsoft.

Основні властивості

- Автоматичне управління пам'яттю.
- Розширені можливості обробки виняткових ситуацій.

- Багатий набір засобів фільтрації введення-виведення.
- Набір стандартних колекцій: масив, список, стек і т. П.
- Наявність простих засобів створення мережевих додатків (у тому числі з використанням протоколу RMI).
 - Наявність класів, що дозволяють виконувати HTTP-запити і обробляти відповіді.
 - Вбудовані в мову засоби створення багатопоточних додатків, які потім були перенести на багато мов (наприклад Python).
 - Уніфікований доступ до баз даних

Hypertext Markup Language (HTML) (укр. мова розмітки гіпертекстових документів) – стандартна мова розмітки для створення веб-сторінок і веб-додатків.

Веб-браузери отримують HTML-документи з веб-сервера або з локальної пам'яті і передають документи в мультимедійні веб-сторінки. HTML описує структуру веб-сторінки семантично і спочатку включені сигнали для зовнішнього вигляду документа.

Елементи HTML є будівельними блоками сторінок HTML. За допомогою конструкцій HTML, зображення та інші об'єкти, такі як інтерактивні форми, можуть бути вбудовані у візуалізовану сторінку. HTML надає засоби для створення структурованих документів, позначаючи структурну семантику тексту, наприклад заголовки, абзаци, списки, посилання, цитати та інші елементи. Елементи HTML окреслені тегами , написаними з використанням кутових дужок .

HTML може вбудовувати програми, написані на мові сценаріїв, наприклад JavaScript, що впливає на поведінку та вміст веб-сторінок. Включення CSS визначає вигляд і компоновання вмісту.

HTML впроваджує засоби для:

- створення структурованого документа шляхом позначення структурного складу тексту: заголовки, абзаци, списки, таблиці, цитати та інше;

- отримання інформації із Всесвітньої мережі через гіперпосилання;
- створення інтерактивних форм;
- включення зображень, звуку, відео, та інших об'єктів до тексту.

CSS (англ. Cascading Style Sheets, укр. Каскадні таблиці стилів) – спеціальна мова, що використовується для опису зовнішнього вигляду сторінок, написаних мовами розмітки даних.

Найчастіше CSS використовують для візуальної презентації сторінок, написаних HTML та XHTML(Extensible Hypertext Markup Language).

CSS використовується авторами та відвідувачами веб-сторінок, щоб визначити кольори, шрифти, верстку та інші аспекти вигляду сторінки. Одна з головних переваг – можливість розділити зміст сторінки (або контент, наповнення, зазвичай HTML, XML або подібна мова розмітки) від вигляду документу (що описується в CSS).

Стилі для відображення сторінки можуть бути:

1) Стилі автора (інформація надана автором сторінки):

- зовнішні таблиці стилів (англ. stylesheet), найчастіше окремий файл або файли .css;
- внутрішні таблиці стилів, включені як частина документу або блоку
- стилі для окремого елемента.

2) Стилі користувача:

- локальний .css-файл, вказаний користувачем для використання на сторінках і вказаний в налаштуваннях браузера (наприклад Opera).

1) Стилі переглядача (браузера):

- стандартний стиль переглядача, наприклад стандартні стилі для елементів, визначені браузером, використовуються коли немає інформації про стиль елемента або вона неповна.

CSS має порівняно простий синтаксис і використовує небагато англійських слів для найменування різних складових стилю.

Стилі складаються зі списку правил. Кожне правило має один або більше селектор (англ. selector) та блок визначення (англ. declaration block). Блок визначення складається із оточеного фігурними дужками списку властивостей.

Властивості в списку оформлюються у вигляді назва властивості, двокрапка (:), значення, крапка з комою (;).

JavaScript (JS) – динамічна, об'єктно-орієнтована прототипна мова програмування. Реалізація стандарту ECMAScript. Найчастіше використовується для створення сценаріїв веб-сторінок, що надає можливість на стороні клієнта (пристрої кінцевого користувача) взаємодіяти з користувачем, керувати браузером, асинхронно обмінюватися даними з сервером, змінювати структуру та зовнішній вигляд веб-сторінки.

JavaScript класифікують як прототипну (підмножина об'єктно-орієнтованої), скриптову мову програмування з динамічною типізацією. Окрім прототипної, JavaScript також частково підтримує інші парадигми програмування (імперативну та частково функціональну) і деякі відповідні архітектурні властивості, зокрема: динамічна та слабкатипізація, автоматичне керування пам'яттю, прототипне наслідування, функції як об'єкти першого класу.

Мова JavaScript використовується для:

- написання сценаріїв веб-сторінок для надання їм інтерактивності;
- створення односторінкових веб-застосунків (React, AngularJS, Vue.js);
- програмування на стороні сервера (Node.js);
- стаціонарних застосунків (Electron, NW.js);
- мобільних застосунків (React Native, Cordova);
- сценаріїв в прикладному ПЗ (наприклад, в програмах зі складу Adobe Creative Suite чи Apache JMeter);

– всередині PDF-документів тощо.

jQuery – набір функцій JavaScript, що фокусується на взаємодії JavaScript і HTML. Бібліотека jQuery допомагає легко отримувати доступ до будь-якого елементу DOM, звертатися до атрибутів і вмісту елементів DOM, маніпулювати ними. Також бібліотека jQuery надає зручний API для роботи з AJAX. Розробка jQuery ведеться командою добровольців на пожертвування.

Можливості:

- Движок крос-браузерних CSS-селекторів Sizzle, що виділився в окремий проект;
- Перехід по дереву DOM, включаючи підтримку XPath як плагіна;
- Обробка подій;
- Візуальні ефекти;
- AJAX-додатки;
- JavaScript-плагіни.

2.2 Вибір середовища розробки

При розробці і реалізації програмного продукту було використано середовище Visual Studio 2019

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

Visual Studio включає в себе редактор вихідного коду з підтримкою технології IntelliSense і можливістю найпростішого рефакторінга коду.

Вбудований відладчик може працювати як відладчик рівня вихідного коду, так і відладчик машинного рівня. Решта вбудовуються інструменти включають в себе редактор форм для спрощення створення графічного інтерфейсу додатку, веб-редактор, дизайнер класів і дизайнер схеми бази даних. Visual Studio дозволяє створювати і підключати сторонні додатки (плагіни) для розширення функціональності практично на кожному рівні, включаючи додавання підтримки систем контролю версій вихідного коду (як, наприклад, Subversion і Visual SourceSafe), додавання нових наборів інструментів (наприклад, для редагування і візуального проектування коду на предметно-орієнтованих мовах програмування) або інструментів для інших аспектів процесу розробки програмного забезпечення (наприклад, клієнт Team Explorer для роботи з Team Foundation Server).

2.3 Принципи та підходи розробки інтерфейсів

Дизайн-проект створювався завдяки інструментам Bootstrap.

Bootstrap – вільний набір інструментів для створення сайтів і веб-додатків. Включає в себе HTML- і CSS-шаблони оформлення для типографіки, веб-форм, кнопок, міток, блоків навігації та інших компонентів веб-інтерфейсу, включаючи JavaScript-розширення.

Bootstrap використовує сучасні напрацювання в області CSS і HTML, тому необхідно бути уважним при підтримці старих браузерів .

Користувачі – головний об'єкт уваги. Потрібно визначити що їм потрібно, використовувати це в дизайні, а потім сформуванати продукт так, щоб користувачі могли виконати дію якнайшвидше. Flat інтерфейс заснований на правилах типографіки та сітки і направлений на молодшу аудиторію, а більш дружелюбні скевоморфічні елементи підійдуть віковим користувачам, які не настільки просунуті.

Оточення – це не тільки платформа для якої робиться дизайн, але також і умови використання, та місце в якому він буде працювати. Наприклад

мобільний телефон суттєво відрізняється від телевізора, – на нього дивляться здалеку, він завжди використовується в приміщенні і керується на віддаленні. Це передбачає інші підходи до розміру тексту, кольору та контрастності інтерфейсу.

Можливості – здатність об'єкта висловлювати свої функції через органи чуття: кнопка своєю опуклістю натякає на її клікабельність, дверна ручка, завдяки формі та розташуванню натякає на те, що її можна потягнути. Все це може бути використано в цифровому дизайні, щоб спонукати користувачів взаємодіяти з об'єктами. Випуклі кнопки, текст який виходить за межі сторінки – це демонстрація можливостей.

Текст. Хороший текст не тільки робить застосунок зрозумілішим, але й створює зв'язок з користувачем, завдяки використовуваному тону. Коли ми говоримо з людьми рідною мовою, а не як з машиною, створюється емоційний зв'язок, який покращує досвід взаємодії з продуктом.

Типографіка. В цифровому дизайні типографіку часто незаслужовано обділяють увагою. Дуже важливо використовувати хорошу типографіку, адже більшість інформації передається у вигляді тексту. Існують базові фактори на які потрібно звертати увагу.

Анімації. З постійним збільшенням потужності комп'ютерів, ми можемо робити не тільки гарний дизайн, але й додавати в нього елегантну динаміку. Легка анімація може задати характер інтерфейсу, поліпшити його інтуїтивність, виділити функції та можливості. Наприклад, якщо ви хочете привернути увагу до кнопки – примусьте її пульсувати.

Тестування. Ніщо не порівняти з тестом дизайну на призначеному пристрої. Роздільна здатність, технологія екрану, кути перегляду та методи введення можуть істотно відрізнитись від вашого комп'ютера. У наш час існує безліч засобів для попереднього перегляду, які перенесуть дизайн з комп'ютера прямо у ваш пристрій, оновлюючись у реальному часі.

Прототипи. Тестування дуже тісно пов'язане з прототипуванням. Прототипи дозволяють нам швидко випробовувати ідеї, не витрачаючи часу на

фінальне «полірування». Починаючи з ескізів на папері, закінчуючи створенням простих застосунків в рідному середовищі розробки. Все залежить від того що саме потрібно: ескізи можна зробити і на папері, а от плавну анімацію доведеться показати на відео.

При розробці додатку використовувались такі компоненти:

- `Navbar` для створення навігаційного меню;
- `Card` – для викреслення кожного об'єкту;
- `Btn` – кнопка для взаємодії між користувачем та додатком
- `Label` – контейнер, що містить бідь-які написи.

3 ПРОЕКТУВАННЯ ІНФОРМАЦІЙНОЇ СИСТЕМИ

3.1 Проектування інформаційної системи за допомогою методології функціонального моделювання SADT

Методологія SADT – це сукупність методів, процедур, правил, які призначені для побудови функціональної моделі об'єкту якої-небудь предметної області. Функціональна модель SADT відображає функціональну структуру об'єкту, тобто дії, які над ним виконуються та зв'язки між цими діями. Основні елементи цієї методології ґрунтуються на таких концепціях:

1. Графічне представлення блокового моделювання. Графіка блоків і дуг SADT-діаграми відображає функцію у вигляді блоку, а інтерфейси входу/виходу представляються дугами, що, відповідно, входять у блок і виходять з нього. Взаємодія блоків між собою описується за допомогою інтерфейсних дуг, що виражають “обмеження”, котрі у свою чергу визначають, коли і яким чином функції виконуються й керуються;

2. Строгість і точність. Виконання правил SADT вимагає достатньої строгості й точності, не накладаючи в той же час надмірних обмежень на дії аналітика.

Правила SADT включають:

- обмеження кількості блоків на кожному рівні декомпозиції (як правило, 3-6 блоків);
- унікальність міток і найменувань (відсутність імен, що повторювалися б);
- синтаксичні правила для графіки (блоків і дуг);
- поділ входів і керувань (правило визначення ролі даних).
- відділення організації від функції, тобто виключення впливу організаційної структури на функціональну модель.



Рисунок 3.1 – Контекстна діаграма

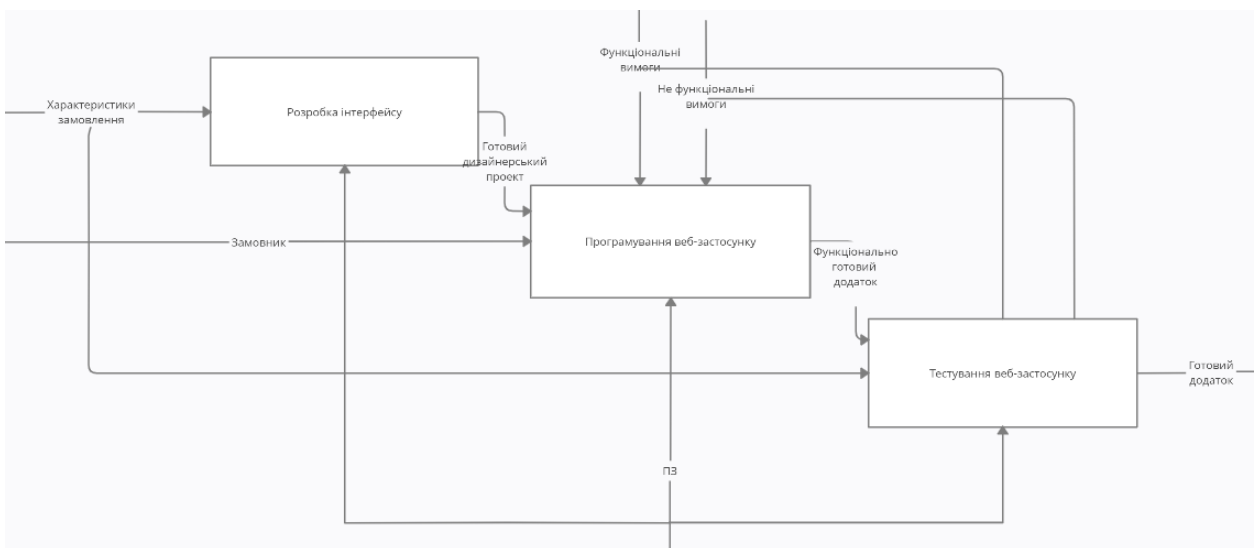


Рисунок 3.2 – Деталізована діаграма 1 рівня.

3.2 Проектування інформаційної системи за допомогою методології Workflow Diagramming

Діаграми методології IDEF3 (Workflow Diagramming) – методологія моделювання і стандарт документування процесів, що відбуваються в системі. Метод документування технологічних процесів є механізм документування та збору інформації про процеси. IDEF3 показує причинно-наслідкові зв'язки між ситуаціями і подіями в зрозумілій експерту формі, використовуючи структурний метод вираження знань про те, як функціонує система, процес або підприємство. Система описується як упорядкована послідовність подій з одночасним описом об'єктів, що мають відношення до моделюючого процесу.

IDEF3 складається з двох методів: Process Flow Description (PFD) - опис технологічних процесів, із зазначенням того, що відбувається на кожному етапі технологічного процесу. Object State Transition Description (OSTD) - опис переходів станів об'єктів, із зазначенням того, які існують проміжні стани у об'єктів в моделюється системі.



Рисунок 3.3 – Контекстна діаграма

3.3 Структура БД

В даному проєкті БД складається з 4 таблиць. 3 з яких відповідають за товари та замовлення, а остання за авторизацію робітників.

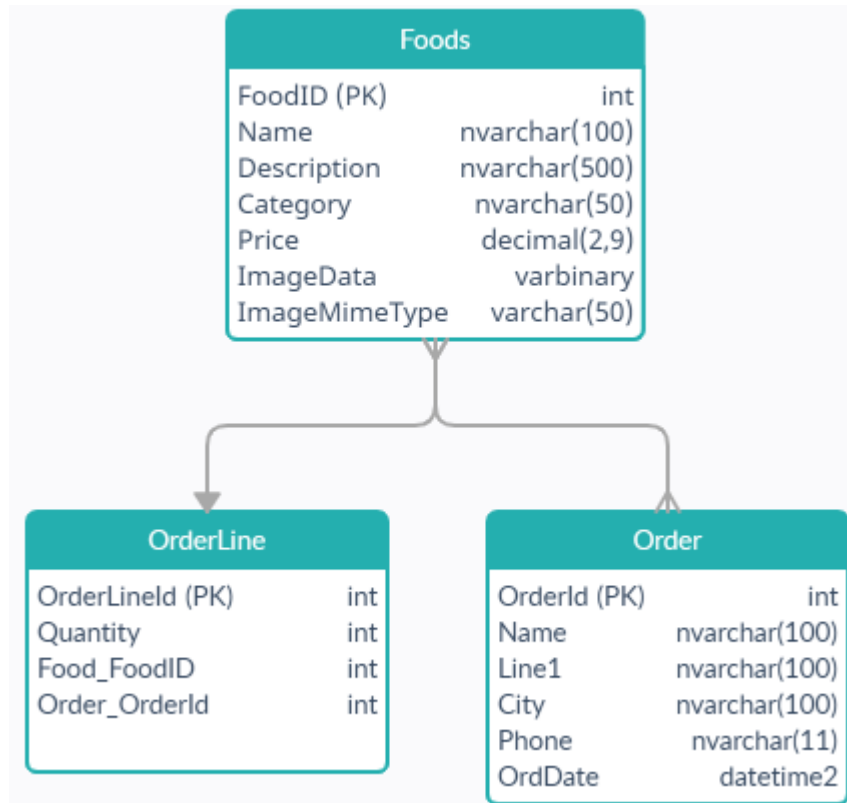


Рисунок 3.4 – Зв’язані таблиці БД

Таблиця OrderLines зв’язує таблицю Foods та Orders через первинні ключі Foods.FoodID та Orders.OrderID. Цей зв’язок потрібен для того, щоби ефективно опрацьовувати замовлення користувачів



Рисунок 3.5 – Таблиця для авторизації

Таблиця Workers потрібна лише для авторизації робітників, тому вона не має ніяких зв’язків з іншими таблицями

3.4 Реалізація функціональних можливостей

Для того щоб користувач мог побачити товари необхідно їх отримати з БД та присвоїти їх класам. Для цього використовується метод на рис. 3.6

```
private IFoodRepository repository;
public int pageSize = 50;
1 reference
public FoodController(IFoodRepository repo)
{
    repository = repo;
}
1 reference
public ActionResult List(string category, int page = 1)
{
    FoodListViewModel model = new FoodListViewModel
    {
        Foodes = repository.Foodes
            .Where(p => category == null || p.Category == category)
            .OrderBy(food => food.FoodId)
            .Skip((page - 1) * pageSize)
            .Take(pageSize),
        PagingInfo = new PagingInfo
        {
            CurrentPage = page,
            ItemsPerPage = pageSize,
            TotalItems = category == null ?
                repository.Foodes.Count() :
                repository.Foodes.Where(game => game.Category == category).Count()
        },
        CurrentCategory = category
    };
    return View(model);
}
0 references
public ActionResult GetImage(int foodId)
{
    Food food = repository.Foodes
        .FirstOrDefault(g => g.FoodId == foodId);

    if (food != null)
    {
        return File(food.ImageData, food.ImageMimeType);
    }
    else
    {
        return null;
    }
}
}
```

Рисунок 3.6 – Отримання об'єктів з БД

Отриманні дані потрібно вивести в задовільному вигляді. Метод на рис 3.7 дозволяє нам окреслити кожен об'єкт та вивести їх у вікно браузеру

```

@model Domain.Entities.Food
<div class="card h-100 border-primary">
  <div class="card-header">
    <strong>@Model.Name</strong>
  </div>
  @if (Model.ImageData != null)
  {
    <div class="card-img-top">
      
    </div>
  }
  <div class="card-body">@Model.Description</div>
  @using (Html.BeginForm("AddToCart", "Cart"))
  {
    <div class="pull-right pull-bottom">
      @Html.HiddenFor(x => x.FoodId)
      @Html.Hidden("returnUrl", Request.Url.PathAndQuery)
      <div class="card-footer">
        @Model.Price.ToString("# грн")
        <input type="submit" class="btn btn-sm btn-success"
          style="position:absolute; bottom:5px; right:3%"
          value="Добавить в корзину" />
      </div>
    </div>
  }
</div>

```

Рисунок 3.7 – Відображення об'єктів в браузері

Для адміністрування БД також необхідні методи видалення та додавання до БД записів. На рис. 3.8 показані методи для оновлення вмісту БД

```
2 references
public void SaveFood(Food food)
{
    if (food.FoodId == 0)
        context.Foodes.Add(food);
    else
    {
        Food dbEntry = context.Foodes.Find(food.FoodId);
        if (dbEntry != null)
        {
            dbEntry.Name = food.Name;
            dbEntry.Description = food.Description;
            dbEntry.Price = food.Price;
            dbEntry.Category = food.Category;
            dbEntry.ImageData = food.ImageData;
            dbEntry.ImageMimeType = food.ImageMimeType;
        }
    }
    context.SaveChanges();
}

2 references
public Food DeleteFood(int foodId)
{
    Food dbEntry = context.Foodes.Find(foodId);
    if (dbEntry != null)
    {
        context.Foodes.Remove(dbEntry);
        context.SaveChanges();
    }
    return dbEntry;
}
```

Рисунок 3.8 Методи видалення та додавання записів

Метод для авторизації робітників зображений на рис. 3.9

```
public class AccountController : Controller
{
    IAuthProvider authProvider;
    0 references
    public AccountController(IAuthProvider auth)
    {
        authProvider = auth;
    }
    0 references
    public ActionResult Login()
    {
        return View();
    }

    [HttpPost]
    0 references
    public ActionResult Login(models.LoginViewModel model, string returnUrl)
    {
        if (ModelState.IsValid)
        {
            if (authProvider.Authenticate(model.UserName, model.Password))
            {
                return Redirect(returnUrl ?? Url.Action("Index", "Admin"));
            }
            else
            {
                ModelState.AddModelError("", "Неправильний логин или пароль");
                return View();
            }
        }
        else
        {
            return View();
        }
    }
}
```

Рисунок 3.9 – Авторизація робітників

4 ПРАКТИЧНА РЕАЛІЗАЦІЯ ТЕХНІЧНОГО ПРОЕКТУ

4.1 Схеми сайту

Карта сайту представлена на рисунку 4.1 та 4.2.

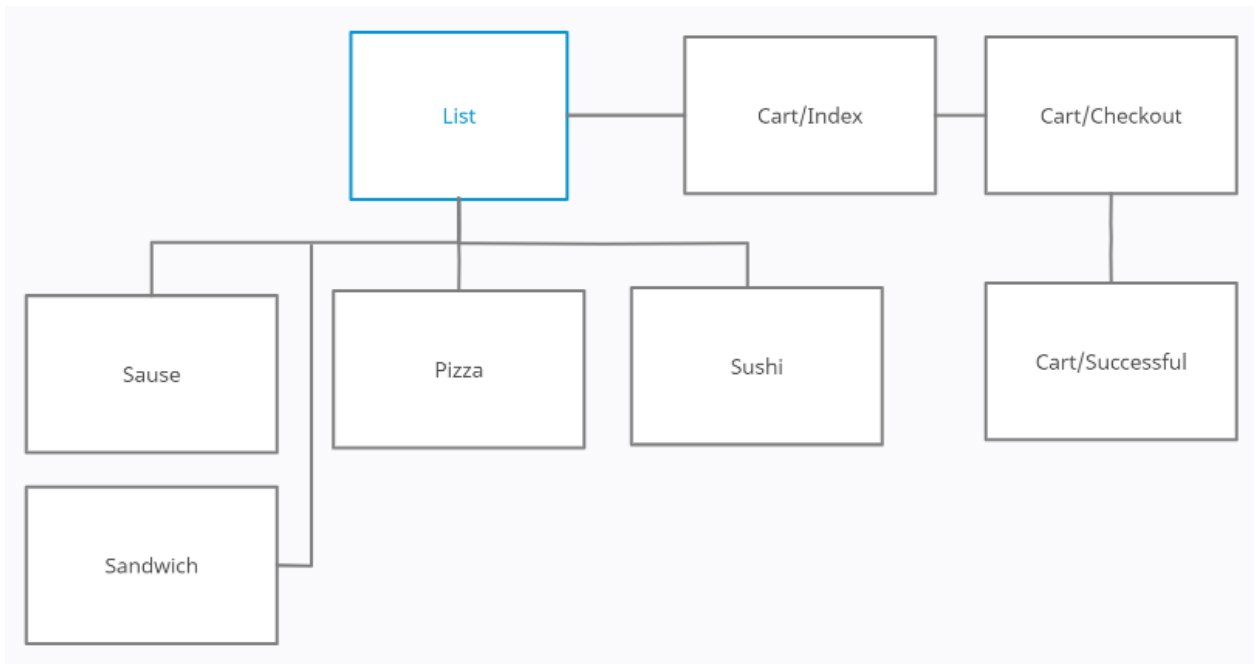


Рисунок 4.1 – Карта сайту для користувача.

При вході на сайт на головну сторінку(List),де можемо побачити весь асортимент страв з описом та ціною. Також зверху знаходиться навігаційна панель, яка виступає в ролі фільтру, та кошик. Після того як вибрали страву, є можливість оформити заказ або продовжувати обирати страви. Для того що би оформити замовлення потрібно ввести ім'я, телефон та адресу.

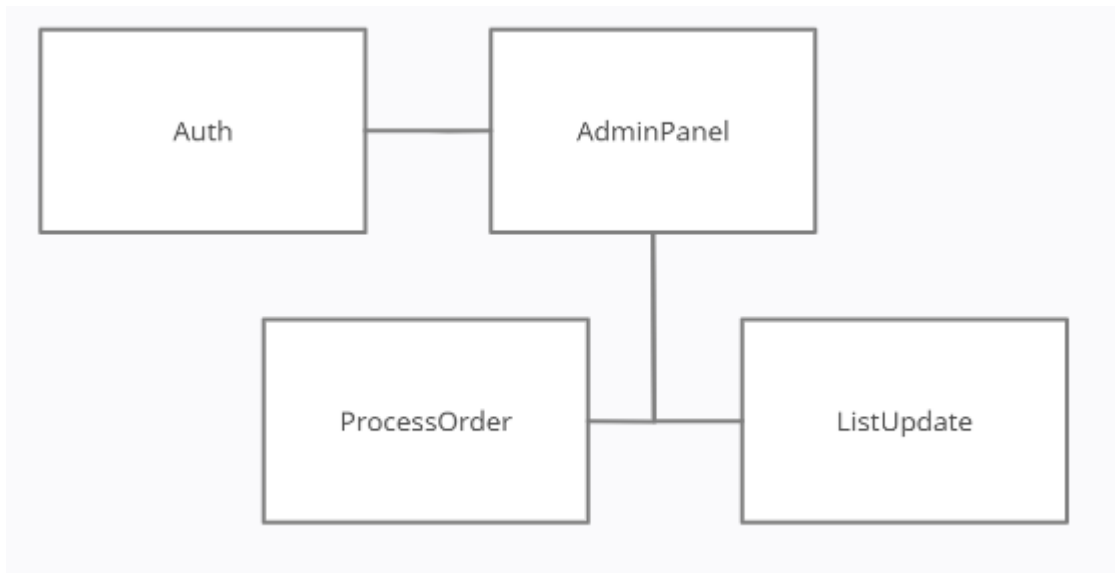


Рисунок 4.2 – Карта сайту для адміністратору.

Для адміністраторів є можливість авторизації після чого можна обробляти замовлення, та редагувати список страв.

4.2 Опис роботи програми

Входячи на сайт, зверху знаходиться навігаційна панель, яка складається з категорії та кошику, у центрі знаходяться всі товари

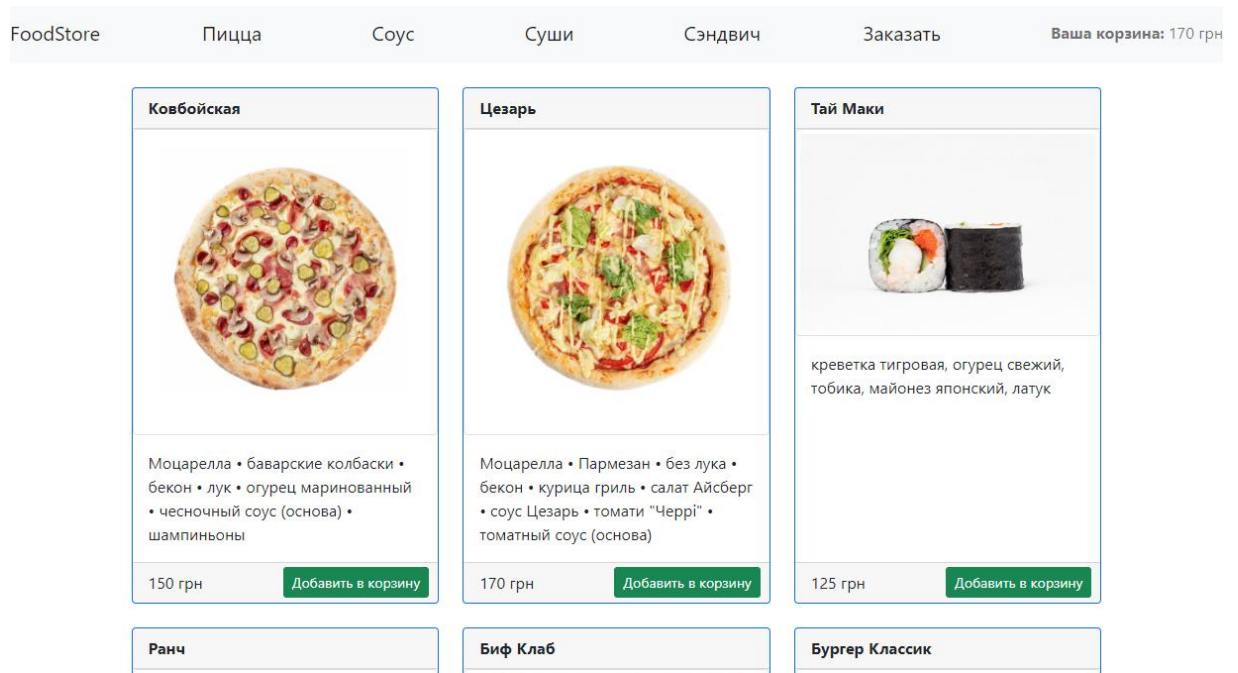


Рисунок 4.3 – Головна сторінка

Натиснувши «Пицца» можна встановити фільтр пошуку на даний тип товару, що допомагає користувачеві.

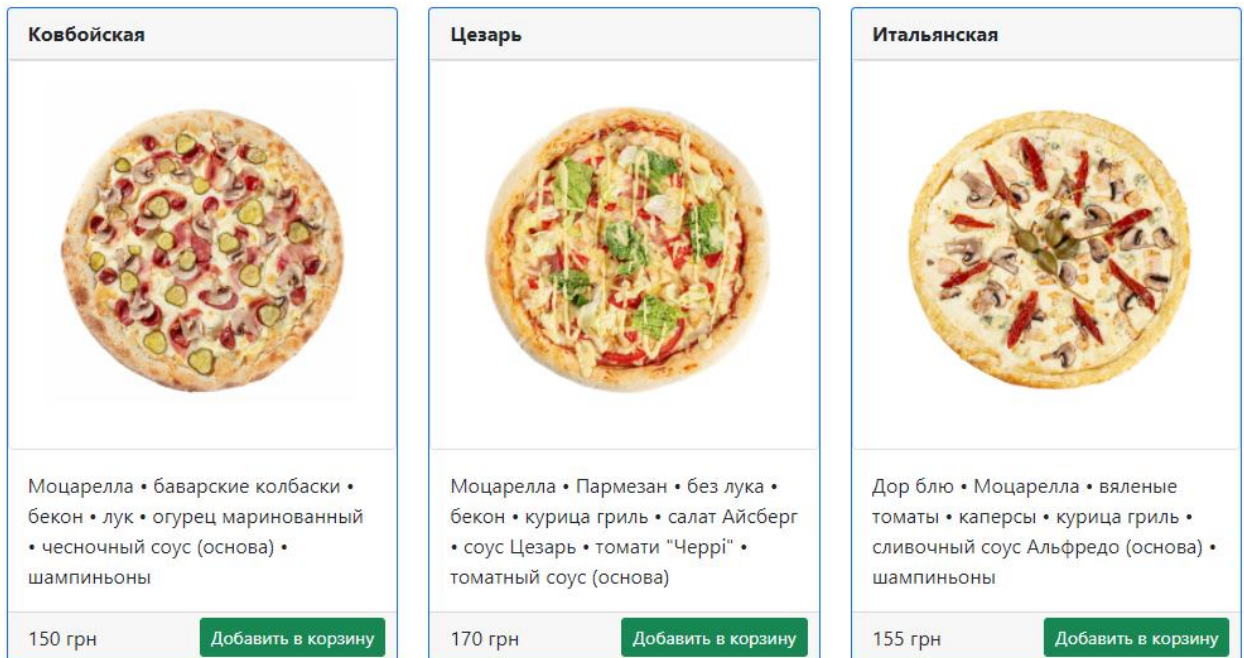


Рисунок 4.4 – Пошук товару за фільтром

Натиснувши «Заказать» користувач перейде до списку товарів, який він обрав, в кошику автоматично обчислюється загальна сума замовлення. Якщо користувач бажає продовжити обирати товар то він натисне «Продолжить покупки», але якщо придбав усе що треба, натиснувши «Оформить заказ» перейде до оформленню заказу

Оформить заказ сейчас

Пожалуйста введи ваши контактные данные, и мы сразу отправим товар!

Данные

Ваше имя:

Александр

Телефон:

88912921

Адрес доставки

Адрес

улица

Город

Одесса

Обработать заказ

Рисунок 4.5 – Оформлення заказу

Якщо користувач не заповнить всі поля то додаток підкаже, де користувач допустив помилку

Оформить заказ сейчас

Пожалуйста введи ваши контактные данные, и мы сразу отправим товар!

- Укажите город

Данные

Ваше имя:

Александр

Телефон:

88912921

Адрес доставки

Адрес

улица

Город

Обработать заказ

Рисунок 4.6 – Помилка при оформленні заказу

Після того як користувач заповнить всі поля, він отримає повідомлення, що заказ був успішно оброблений.

Спасибо!

Спасибо за выбор нашего магазина. Заказ в кратчайшие сроки будет отправлен в службу доставки.

Рисунок 4.7 – Повідомлення

Щоб увійти в додаток як адміністратор потрібно пройти через авторизацію

Вход в систему

Пожалуйста, войдите в систему, чтобы получить доступ к админ. панели:

Имя пользователя:

Пароль:

Рисунок 4.8 – Форма авторизації

Після авторизації адміністратор має можливість редагувати, видаляти та додавати нові страви

21	4 сыра	Пицца	189 грн	Удалить
22	Тоскана	Пицца	190 грн	Удалить
23	Дьябло	Пицца	177 грн	Удалить
24	4 мяса	Пицца	177 грн	Удалить
25	Баварская	Пицца	177 грн	Удалить
26	Футо ролл	Суши	155 грн	Удалить
27	Сяке маки	Суши	110 грн	Удалить
28	Филадельфия classic	Суши	156 грн	Удалить
29	Канада	Суши	175 грн	Удалить
30	Онегай	Суши	125 грн	Удалить
31	Полосатый дракон	Суши	165 грн	Удалить

[Добавить блюдо](#)

Рисунок 4.9 – Форма для редагування бази даних

Натиснувши на назву товару адміністратор перейде до вкладки зміни цього товару, якщо натисне на «Удалить» то видалить обраний запис, і якщо «Добавить блюдо» то перейде до форми створення нового товару, як показано на рис 4.10

Редактирование блюда «>>»

Название

Акулий плавник

Описание

Рис, плавник акулы, унаги соус, нори

Категория

Суши

Цена (грн)

150

Картинка [Выберите файл...](#)

Нет картинки

[Сохранить](#)

[Отменить изменения и вернуться к списку](#)

Рисунок 4.10 – Створення нового товару

4.3 Тестування web-додатку

Мета тестування: виявлення функціональних помилок, невідповідностей технічному завданню і очікуванням Замовника шляхом реалізації стандартних, а також нетривіальних тестових сценаріїв.

Класифікація функцій, які будуть протестовані:

- 1) додавання інформації;
- 2) перегляд інформації, перегляд відфільтрованої інформації;
- 3) видалення та редагування інформації з бази даних.

Календарний план тестування наведено у таблиці 3.1.

Таблиця 3.1 – Календарний план робіт

Номер тесту	Дата Початку	Дата Кінця	Дні						
			1	2	3	4	5	6	7
1	22.05.2021	29.05.2021							
2									
3									

У таблиці 3.2 показані результати тестування.

Таблиця 3.2 – Успішність виконання тестів

Відмітка на виконання	Види тестів	Примітка
✓	Додавання інформації	Тест пройдено успішно
✓	Перегляд інформації, перегляд відфільтрованої інформації.	Тест пройдено успішно
✓	Видалення та редагування інформації з бази даних.	Тест пройдено успішно

ВИСНОВКИ

В результаті виконання роботи розроблено інформаційно-програмне забезпечення автоматизації обслуговування клієнтів, яке дозволить автоматизувати діяльність менеджера, скоротити часові витрати на продаж товару, вивантажувати або завантажувати дані з системи автоматизації фірми на віртуальний сервер, де розміщений інтернет-магазин.

Сформульовано та проаналізовано проблеми, що виникають при проектуванні інформаційно-програмного забезпечення автоматизації обслуговування клієнтів, а також запропоновано шляхи їх вирішення.

Проведено когнітивний аналіз необхідності розробки інформаційно-програмного забезпечення і його переваги в порівнянні з ручною роботою. Зроблено детальний огляд і аналіз можливостей інтеграції системи автоматизації та інтернет-магазину, розглянуті і обрані кращі середовища розробки для інтернет-магазину.

Проведено детальний звіт про етапи розробки Web-сайту.

Дана система розроблялася для рішення наявної проблеми.

Переваги розробленої системи:

- відсутність дій з боку менеджера для здійснення інтеграції;
- можливість збереження структури каталогів;
- швидке і безпомилкове виконання операцій, на які вручну

Витрати б велику кількість часу.

СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ

1. Гутманс Э., Баккен С., Ретанс Д. PHP 5. Профессиональное программирование. пер. с англ. СПб.: Символ-Плюс, 2006. 704 с.
2. Дейт К. Дж. Введение в системы баз данных: Пер. с англ.- 6-е изд.– К.; М.; СПб.: Издательский дом «Вильямс», 2000. – 848 с.
3. Томасон Лаура, Люк Веллинг. Разработка WEB-приложений на PHP и MySQL: пер. с англ. 2-е изд. СПб.: ООО ДиаСофтЮП, 2003. 672 с.
4. Мейер Д. Теория реляционных баз данных: Пер. с англ. – М.: Мир, 1987. – 608 с.
5. Welling L., Thomson L. PHP and MySQL Web Development (4th Edition). Addison-Wesley Professional, 2008. – 1008 с.
6. Карпова Т.С. Базы данных: модели, разработка, реализация. СПб.: Питер, 2001. 304 с.
7. Люк Веллинг, Лора Томсон. Разработка WEB-приложений с помощью PHP и MySQL. М.: Вильямс, 2010. 848 с.
8. Jon Galloway, K. Scott, Professional ASP.NET MVC 5, 2014. – 624 с.