МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ

ХАРКІВСЬКИЙ НАЦІОНАЛЬНИЙ УНІВЕРСИТЕТ

РАДІОЕЛЕКТРОНІКИ

ЗВІТ

З ПЕРЕДАТЕСТАЦІЙНОЇ ПРАКТИКИ БАКАЛАВРІВ

Місце проходження практики «НВПП Велєс-М»

у період з "18" квітня по "14" травня 2017 р.

Тема індивідуального завдання:

Система організації покупок у розумному магазині

|  |  |
| --- | --- |
| Студ. ПІ-13-6  Кокорін Павло Олександрович  \_\_\_\_\_\_\_\_\_  13.05.2017 | Керівник практики  Ворочек Ольга Григорієвна  Робота захищена з оцінкою \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_  Дата «\_\_»\_\_\_\_\_\_\_\_\_\_\_\_\_\_2017 |

Керівник випускної кваліфікаційної роботи \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

Ворочек Ольга Григорієвна (рекомендована оцінка)

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

(підпис)

Харків 2017

РЕФЕРАТ

Звіт з переддипломної практики бакалаврів: 20 с., 2 рис., 10 джерел.

Об'єкт проектування – система організації покупок у розумному магазині за допомогою мобільного додатку .

Метою роботи є створення веб-системи та мобільного додатку організації покупок у розумному магазині.

Методи розробки базуються на технології .NET MVC 5, сервер бази даних MSSQL і Web-сервер IIS.

У результаті роботи здійснена повне проектування та мобільного додатку організації покупок у розумному магазині, яка дозволяє користувачам створювати персональний акаунт на сайті, поповняти кошти цього акаунту та оплачувати товари у розумних магазинах, де інтегрована ця система.

ВЕБ-СИСТЕМА, MSSQL, РОЗУМНИЙ МАГАЗИН, ТОВАРИ, МОБІЛЬНИЙ ДОДАТОК, MSSQL, MVC 5, IIS.

ЗВІТ

[ВСТУП 4](#_Toc482443389)

[1 ОПИС ПІДПРИЄМСТВА 6](#_Toc482443390)

[2 АНАЛІЗ ПРЕДМЕТНОЇ ОБЛАСТІ 9](#_Toc482443391)

[2.1 Основні відомості 9](#_Toc482443392)

[2.2 Аналіз існуючих рішень 9](#_Toc482443393)

[3 ПОСТАНОВКА ЗАДАЧІ 13](#_Toc482443394)

[4 ФОРМУВАННЯ ВИМОГ ДО ПРОГРАМНОЇ СИСТЕМИ 14](#_Toc482443395)

[ВИСНОВКИ 19](#_Toc482443396)

[ПЕРЕЛІК ПОСИЛАНЬ 20](#_Toc482443397)

# ВСТУП

У сучасному світі усе навколо стрімко автоматизується. Автоматизування ручної праці це дуже важливий процес, який підвищує ефективність роботи, зменшує затрати, відкриває нові можливості и т.д. Тому не так давно з’явилися інтернет-магазини, в яких люди мають можливість робити покупки різноманітних товарів. Але ця інновація не витіснила звичайні магазини, в яких процес купівлі товарів все ж таки потребує ручної праці касира. Основні функції касирів у супермаркетах це підрахування суми купованих товарів, прийом грошових коштів, та видача решти. Я вважаю цей процес може и має бути автоматизований, за допомогою такого напрямку в IT, як IOT.

Я вважаю актуальність цієї ідеї дуже висока. По перше це вигідно звичайним людям, бо автоматизовані процеси виконуються швидше ніж ручні, це сприяє вирішено такої проблеми як великі черги у магазинах. По друге це може бути вигідно власникам магазинів, за умовою якщо вони будуть отримувати більший прибуток, або витрачати менше коштів на утримання магазинів (зарплатня касира). Звісно система для автоматизування закупок має бути легко спроектована, щоб по перше клієнти дуже швидко розуміли як її використовувати, по друге вона легко інтегрувалась до різноманітних супермаркетів. Також вона має бути не дуже дорогою (не більше ніж вартість утримання касирів). При цих умовах ми маємо всі підстави, що розробити цю систему та пропонувати її людям.

Під автоматизацією процесу покупки товарів ми розуміємо систему, яка дозволяє клієнту створити свій персональний акаунт, покласти кошти на цей акаунт, прийти до любого супермаркету, у якому інтегрована ця система, придбати потрібні товари, оплативши їх за допомогою власного смартфона. Як ми бачимо при цьому сценарії касир не потрібен. Звісно система передбачає також апаратну реалізацію, але у рамках цього диплому ми зосередимось на програмній реалізації. До апаратної реалізації відноситься :

– маркування товарів, для їх ідентифікації;

– створення терміналу оплати товарів. Функції термінала :

1. визначає групу товарів, які бажає оплатити користувач;
2. визначає суму вартості цих товарів;
3. має графічний інтерфейс взаємодії;
4. оплата за карткою;
5. оплата за телефоном;
6. оплата частини групи товарів.

– створення безопарних виходів з магазину, щоб через них неможливо було пронести

маркованих неоплачених товар.

До програмної реалізації відноситься :

– створення сайту за наступними умовами:

1. регистрація користувачів, з наступним розділенням на ролі (адміністратор, користувач, власник магазину);
2. аутентифікація користувачів за соціальними мережами (VK, Facebook, linkedIn);
3. використання 3-слойної архітектури (UI, BLL, DAL);
4. локалізація сайту за декількома язиками, але обов’язково англійським;
5. використовування автоматичного маппінгу об’єктів;
6. використовування логеру для запису виключних ситуацій;
7. управління персональними даними;
8. управління персональних акаунтом користувача.

– створення мобільного додатку з можливістю оплати групи товарів (емуляція);

Метою роботи буде реалізація програмної частини системи, розглядання існуючих аналогів, аналіз предметної області. У перспективі ця система може застосовуватись у будь-яких магазинах з касовою системою оплати.

# 1 ОПИС ПІДПРИЄМСТВА

Практика проходила на підприємстві "НВПП Велєс-М". Ця компанія надає високоякісні ІТ послуги для клієнтів у таких сферах як електронний бізнес, документообіг, високонавантажені веб-системи, мобільні додатки.

Наразі Харківська філія налічує більше 30 працівників. Компанія «НВПП Велєс-М» знаходиться на ринку з 2009 року, почавши свою роботу з Net, PHP та веб-дизайну. Зараз компанія є аутсорсинговим центром, що спеціалізується на розробці .Net, PHP з філією в Харкові (Україна). Компанія НВПП Велєс-М складається з команди досвідчених консультантів і розробників, креативних дизайнерів і фахівців в області маркетингу, які знають, як досягти найкращих результатів.

Працівники повною мірою володіють популярними технологіями, але, в той же час, відкриті для нових технік і постійно прагнуть до вдосконалення. Високий професіоналізм дозволяє їм створювати додатки, які перевершують всі очікування.

Більшість проектів розробляються із використанням гнучких методологій. Гнучка розробка програмного забезпечення — клас методологій розробки програмного забезпечення, що базується на ітеративній розробці, в якій вимоги та розв'язки еволюціонують через співпрацю між само організовуваними багатофункціональними командами. Більшість гнучких методологій націлені на мінімізацію ризиків, шляхом зведення розробки до серії коротких циклів, що мають назву ітерацій, які зазвичай тривають один-два тижні.

Гнучка розробка - найкращий засіб для підвищення продуктивності розробників програмного забезпечення. Різноплановий сервіс-портфоліо і, відповідно, різноманітність проектів, дозволяє не тільки стабільно зростати, алей надавати великі можливості для розвитку співробітникам.

Під час навчання студенти підвищують свій професійний рівень, беруть участь у роботі семінарів і гуртків, організованих на базі наукових лабораторій, лабораторії Бізнес-Інкубатор, учбово-наукового загону «Програміст».

Студенти мають можливість за бажанням вивчати додаткові навчальні дисципліни, що викладаються науковцями університетів-партнерів дистанційно англійською мовою. Зокрема це 5 дисциплін з комп’ютерної графіки та 6 дисциплін з програмування для бакалаврської підготовки. Це дозволяє студентам суттєво підвищити свій рівень фахової підготовки та отримати відповідні сертифікати та кредити за окремими дисциплінами.

З 2008 року кафедра щорічно проводить Відкриту міжнародну Зимову школу з програмування. Зимова школа – це унікальний захід, єдиний на території Україні. В програмі школи органічно поєднуються теоретична і практична підготовка у вигляді лекційних занять і щоденних змагань з розв’язання складних і цікавих алгоритмічних задач. Перебування в школі дає можливість учасникам і тренерам з різних міст інтенсивно спілкуватися і полегшує необхідний для постійного підвищення професійного рівня обмін досвідом.

Фундаментальною проблемою, над якою працюють на кафедрі, є створення нового покоління обчислювальних систем та розробка на цій базі методів моделювання та синтезу інтелектуальних систем і процесів.

Науковими напрямками кафедри є:

• розробка формального апарату методів логіки, алгебри, лінгвістичної алгебри та систем логічної підтримки проектування нових перспективних інформаційних технологій;

• розробка математичних моделей процесів інтелектуальної діяльності (зору, слуху, пізнання) як теоретичної бази створення інженерних систем;

• розробка систем комп’ютерного зору;

• проектування та розробка систем дистанційного комп’ютерного та мобільного навчання;

• розпізнавання образів;

• інтелектуальний аналіз даних;

• біометричні, медичні та діагностичні системи.

На кафедрі приділяють увагу розвитку наукової студентської творчості. Тут сформовані умови для власних студентських розробок як ініціативних проектів для подальшого залучення зовнішнього інвестування. Студентам пропонується займатися власними розробками за перспективними напрямками кафедри. Створюється інформаційний портал для пошуку та сумісної роботи над власними проектами викладачів і студентів. Планується більш широке залучення студентів до різноманітних конкурсів інноваційних проектів. Студентам створюються умови для трансформації успішних проектів у власний бізнес.

Наукові дослідження проводяться у рамках наукових лабораторій та центрів: НДЛ «Моделювання інтелектуальних систем», НДНЛ «Програмне забезпечення автоматизованих систем», НДНЛ «Інформаційні технології у системах навчання і машинного зору», лабораторія мозкоподібних структур НАН України, яка працює над розробкою теорії та принципів побудови мозкоподібних ЕОМ з реляційними мережами.

Навесні 2007 року кафедра виступила ініціатором підписання угоди про освітнє та наукове співробітництво в ІТ-галузі між Харківським національним університетом радіоелектроніки та Інститутом математики і системної інженерії (School of Mathematics and Systems Engineering) університету м. Вакхо (Linnaeus University), Швеція.

# 2 АНАЛІЗ ПРЕДМЕТНОЇ ОБЛАСТІ

## 2.1 Основні відомості

Предметною областю даного проекту є процес роздрібної купівлі товарів у магазинах з касовою системою оплати. Специфікою цієї області є широкий спектр користувачів. Звичайно люди найрізноманітніших професій, віків, освіти, можливостей та достатків користуються супермаркетами кожен день. Цей фактор вимагає розробки системи до якої буде легко адаптуватися різним людям. Також ця особливість потребує «плавного» інтегрування системі до існуючих супермаркетів, бо знайдуться люди, які не захочуть нічого змінювати і, щоб вони не залишили цей маркет, треба залишити місця зі старою системою оплати. Дуже важливу роль грає ще те, що система буде використовувати банковські операції, тому обов’язково знайдуться зловмисники, які будуть намагатися вкрасти персональні дані користувачів, щоб отримати доступ до акаунтів. Тому високу увагу треба уділити безпеці даних. Також слід зауважити, що мі маємо справу з найрізноманітнішими товарами, різних форм, розмірів, деякі зберігаються у холодильниках, деякі запаковані у контейнери різних матеріалів и т.д. Тому дуже важливим є процес маркування продуктів, щоб він залишався діючим довгий час, щоб він розпізнавався на достатній відстані.

## 2.2 Аналіз існуючих рішень

На даний момент такі системи тільки зароджуються. Широкого застосування вони поки що не отримали, але дуже швидко набирають популярність. Наприклад, компанія Amazon представила концепцію супермаркету майбутнього, повністю позбавленого від касових терміналів і черг, пов'язаних з оплатою покупок.

Нова платформа названа Amazon Go. Ідея полягає в тому, щоб автоматично відстежувати дії відвідувачів у торговельному залі, контролюючи, які товари вони беруть з полиць або ставлять назад.

Для здійснення покупок відвідувачеві на вході в супермаркет буде досить відсканувати спеціальний код в додатку Amazon Go на смартфоні. Після це в справу вступить новітня система комп'ютерного зору з алгоритмами глибокого машинного навчання.

Аналізуючи дії покупців, а також інформацію від численних сенсорів, встановлених в торгових залах та на стелажах, комп'ютерна система зможе формувати точний список товарів, взятих кожним відвідувачем. Якщо користувач виставить який-небудь продукт назад на полицю, він буде автоматично «викреслять» з віртуальної корзини.

Після завершення шопінгу відвідувач може спокійно вийти з супермаркету, не піклуючись про оплату: гроші автоматично спишуться з рахунку, асоційованого з додатком Amazon Go.

В даний час нова система проходить тестування за участю співробітників Amazon. Супермаркет майбутнього пропонує готові до вживання продукти харчування, зокрема, легкі закуски.

Перший магазин Amazon Go, як очікується, розкриє двері на початку 2017 року, він розташується в Сіетлі.

У Москві компанія X5 Retail Group, що володіє торговими мережами «Перекресток», «Карусель», «Пятерочка», відкриє влітку 2013 року перший в місті інноваційний супермаркет, який буде працювати без продавців. Називатися він буде «ПеПяКа», повідомляє газета «Известия».

Всі продукти в «супермаркеті майбутнього» забезпечать радіочастотними чіпами RFID, ця технологія дозволить не викладати товари на транспортну стрічку. Вартість вмісту візка порахують на виході, а розплатитися можна буде через спеціальний термінал за допомогою банківської карти або навіть мобільного телефону.

Розумними в магазині стануть навіть полки, які будуть сповіщати менеджерів, які товари закінчуються в залі або на складі. Творці супермаркету також планують ввести систему розпізнавання осіб для ведення вікової статистики по покупцям. Все це дозволить скоротити витрати як мінімум на 20%.

Першим пілотним магазином, де випробують інноваційний підхід до продажу, стане супермаркет «ПеПяКа» на території офісу X5 Retail Group. При цьому в партнерах компанії значиться корпорація «Роснано», а загальна сума інвестицій становить 350 мільйонів рублів.

Експерти, втім, налаштовані скептично. По-перше, технологія RFID має свої недоліки: мітки погано читаються з металевих поверхонь і ємностей з рідинами. А якщо товар загорнути у фольгу, сигнал і зовсім не пройде. Але головна проблема - за чий рахунок встановлювати мітки RFID, кожна з яких коштує три рубля. Наприклад, американська сісти Wal-Mart вже багато років впроваджує ці технології, але до сих пір не зуміла домовитися з постачальниками. Також слід відмітити, що в інтернеті більше немає інформації щодо цього проекту, тому можна припустити, що він був провальний.

Магазини без продавців з'являться в Японії до 2025 року. Уряд Японії домовився про співпрацю з п'ятьма найбільшими операторами мереж супермаркетів (Seven & I Holdings, Lawson, Ministop, FamilyMart Uny Holdings і East Japan Railway), щоб спільно розробити систему автоматичних магазинів без продавців до 2025 року. Про це повідомляє видання Nikkei Asian Review.

Нова система дозволить не пробивати штрих-код кожного товару окремо, а зчитувати інформацію продуктів в кошику покупця одночасно з допомогою спеціальних чіпів. Впровадження такої системи в 50 тис. Магазинів країни буде коштувати близько 46 млн доларів США.

У чіпах буде міститися інформація про дату і місце виготовлення продукту, а також про дату закінчення терміну придатності. Інформація може бути переглянута віддалено, що дозволить виробникам перевіряти стан продукції для коригування планів продажів і виробництва. Також інформація буде доступна і покупцям. Вартість одного чіпа обійдеться в 10 ієн (0,09 доларів).

Міністерство економіки, торгівлі і промисловості Японії прийняв рішення надати підтримку виробникам для масового випуску таких чіпів. Саме впровадження автоматичних магазинів пов'язано з нестачею продавців в Японії.

У Шанхаї відкрився перший магазин без продавців, шведська інноваційна компанія Wheelys відкрила в Шанхаї перший в світі магазин без продавців. Магазин працює 24 години на добу, сім днів на тиждень, і це тільки пілотний проект, компанія розраховує продавати свою концепцію по франшизі. Варто відзначити, що раніше про плани відкрити подібний магазин повідомляв Amazon, але, як ми бачимо, гіганта ринку переграла зовсім невелика компанія.

Після успішного регіонального тестування в маленькому шведському містечку в минулому році, Wheelys відкрила магазин Wheely247 в Шанхаї як приклад працездатності концепції в великих містах.

Купувати в такому магазині нескладно. Клієнт встановлює на смартфон додаток, що дозволяє отримати доступ до магазину. Для покупки товару потрібно просто відсканувати штрих-код товару, який він хоче придбати, і при виході з магазину додаток списує з кредитної картки вартість покупок.

З метою попередження крадіжок весь простір магазину проглядається відеокамерами.

Надалі Wheelys планує продавати ліцензії на використання технології, так що будь-який рітейлер може інтегрувати її в уже існуючі магазини. За словами компанії, «те, що Uber зробив для таксі, ми робимо для роздробу». Зараз магазин працює в форматі бета-тестування, тобто для його відвідування ще треба залишити заявку на сайті, це зроблено, щоб уникнути напливу цікавих роззяв.

# 3 ПОСТАНОВКА ЗАДАЧІ

Інформаційна система організації покупок у розумному магазині повинна надавати користувачеві можливості оплати товару за допомогою свого телефону, за допомогою картки, готівкою. Також має бути доступна функція оплати частки товару, якщо користувач у останній момент змінив свою думку та не хоче купляти частину товару. Система носить характер банківських операцій, тому має бути максимально захищеною від зловмисників, які можуть намагатися вкласти персональні данні користувачів. В рамках інформування користувача, вона повинна своєчасно надавати актуальну і достовірну інформацію про можливості таких дій, сплата, недостатність коштів, неможливість сплати и т.д. В цілому система повинна надавати користувачеві весь обсяг інформації, необхідний для того, спокійно користуватися системою не користуючись додатковою інформацією.

Метою роботи є проектування та реалізація програмної системи покупок у розумному магазині, зрозуміло, що частина функцій буде носити характер ємуляції. Програмне забезпечення складається з 2 додатків: веб-додаток, написаній за технологією .NET MVC 5 та мобільний додаток реалізований на операційній системі Android.

До функціональних вимог можна віднести:

* можливість створення користувача, з власним акаунтом коштів, та змінювання персональних даних ;
* завантаження коштів на акаунт;
* синхронізація веб-додатка з мобільним додатком;
* оплата покупок з мобільного додатка;

Нефункціональні вимоги:

* серверна частина повинна бути реалізована за допомогою технології .NET [8] та мови C#;
* в якості першої СУБД потрібно обрати MS SQL Server;
* клієнт являє собою WEB-додаток[10], створений з використанням ASP.NET MVC. Повинен бути реалізований за допомогою мови програмування C# [3], JavaScript [6].

# 4 ФОРМУВАННЯ ВИМОГ ДО ПРОГРАМНОЇ СИСТЕМИ

В сучасних інформаційних технологіях фаза життєвого циклу, на якій фіксуються вимоги на розробку програмного забезпечення, є визначальною для його якості, термінів та вартості робіт. Саме на цій фазі мають бути зафіксовані реальні потреби користувачів, що стосуються функціональних, операційних та сервісних можливостей.

Вимоги до програмного забезпечення − набір вимог щодо властивостей, якості та функцій [програмного забезпечення](https://uk.wikipedia.org/wiki/%D0%9F%D1%80%D0%BE%D0%B3%D1%80%D0%B0%D0%BC%D0%BD%D0%B5_%D0%B7%D0%B0%D0%B1%D0%B5%D0%B7%D0%BF%D0%B5%D1%87%D0%B5%D0%BD%D0%BD%D1%8F), що буде розроблено, або знаходиться у розробці. Вимоги визначаються в процесі [аналізу вимог](https://uk.wikipedia.org/wiki/%D0%90%D0%BD%D0%B0%D0%BB%D1%96%D0%B7_%D0%B2%D0%B8%D0%BC%D0%BE%D0%B3) та фіксуються в специфікації вимог, [діаграмах прецедентів](https://uk.wikipedia.org/wiki/%D0%94%D1%96%D0%B0%D0%B3%D1%80%D0%B0%D0%BC%D0%B0_%D0%BF%D1%80%D0%B5%D1%86%D0%B5%D0%B4%D0%B5%D0%BD%D1%82%D1%96%D0%B2) та інших [артефактах](https://uk.wikipedia.org/wiki/%D0%90%D1%80%D1%82%D0%B5%D1%84%D0%B0%D0%BA%D1%82) процесу аналізу та розробки вимог.

Види вимог за рівнями:

а) бізнес-вимоги − визначають призначення ПЗ, можуть описуватися в документі про бачення (vision) та документі про межі проекту (scope);

б) вимоги користувача − визначають набір завдань користувача, які повинна вирішувати програма, а також сценарії їхнього вирішення в системі. Ці вимоги можуть мати вигляд тверджень, варіантів використання, історій користувача, сценаріїв взаємодії;

в) функціональні вимоги − визначають «що» повинен робити програмний продукт. Ці вимоги описуються в документі Специфікації програмного забезпечення (SRS).

Для кращого розуміння програмної системи, що розроблюється розглянемо діаграму прецедентів (use case).

Діаграма прецедентів – в [UML](https://uk.wikipedia.org/wiki/UML) [діаграма](https://uk.wikipedia.org/wiki/%D0%94%D1%96%D0%B0%D0%B3%D1%80%D0%B0%D0%BC%D0%B0), на якій зображено відношення між акторами та прецедентами в системі. Також перекладається як діаграма варіантів використання.

Діаграма прецедентів є [графом](https://uk.wikipedia.org/wiki/%D0%93%D1%80%D0%B0%D1%84_(%D0%BC%D0%B0%D1%82%D0%B5%D0%BC%D0%B0%D1%82%D0%B8%D0%BA%D0%B0)), що складається з множини акторів, прецедентів (варіантів використання), асоціацій між акторами та прецедентами, відношень серед прецедентів, та відношень узагальнення між акторами. Діаграми прецедентів відображають елементи моделі варіантів використання.

Суть даної діаграми полягає в наступному: проектовану систему представлено у вигляді безлічі сутностей чи акторів, що взаємодіють із системою за допомогою варіантів використання. Варіант використання (use case) використовується для опису послуг, які система надає актору. Іншими словами, кожен варіант використання визначає деякий набір дій, який виконує система при діалозі з актором.

Розпочинається створення діаграми використання із ідентифікації варіантів використання та дійових осіб. Це необхідно для того, щоб:

* чітко розмежувати систему і її оточення;
* визначити, які дійові особи і як саме взаємодіють з системою, який функціонал (варіанти використання) очікується від системи;
* визначити і описати загальні поняття, які необхідні для детального опису функціонала системи (прецедентів).

Алгоритм створення діаграми прецедентів:

* визначення дійових осіб;
* визначення варіантів використання;
* складання опису кожного варіанту використання;
* опис моделі прецедентів в цілому.

Таким чином діаграма прецедентів чітко відображує акторів, прецеденти, а також відносини між ними і наочно ілюструє узагальнене представлення функціонального призначення системи.

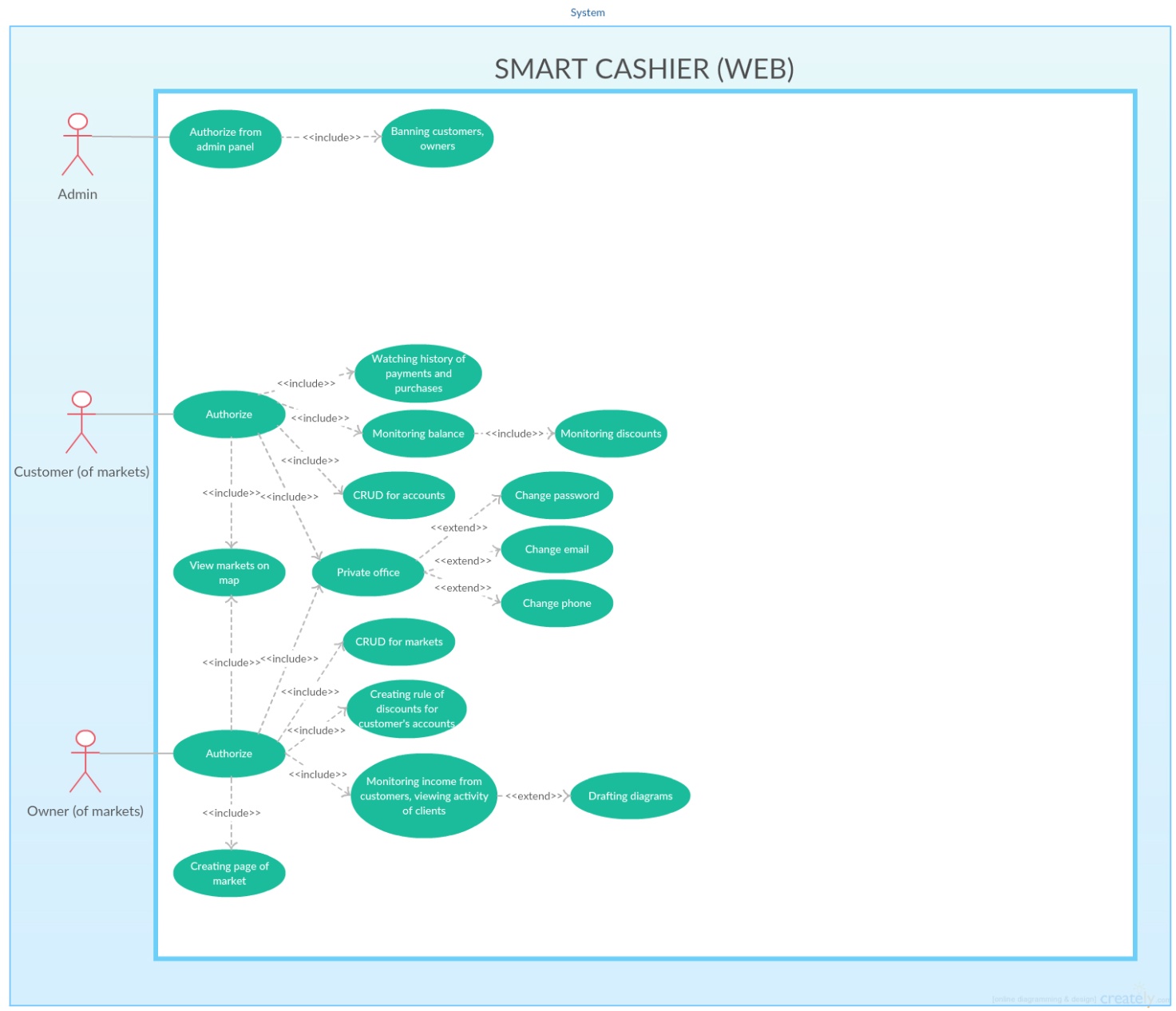


Рисунок 3.1 – Діаграма прецедентів

Використовуючи її, розробнику легше розуміти систему, підсистему або класи, а також поглянути ззовні на переваги використання елементів для того чи іншого контексту.

На етапі аналізу вимог і попереднього проектування системи на основі побудованих моделей системних прецедентів будуються діаграми класів системи.

Діаграма класів, будучи логічним поданням моделі, представляє детальну інформацію про структуру моделі системи з використанням термінології класів об’єктно-орієнтованого програмування, а саме: про внутрішній устрій системи (про архітектуру системи). На діаграмі класів можуть бути вказані внутрішня структура і типи відносин між окремими об’єктами і підсистемами, що призводить до розвитку концептуальної моделі системи.

Клас в мові UML позначає деяку множину об’єктів, що володіють однаковою структурою і взаємозв’язками з об’єктами інших класів. У діаграмах класів системи вказуються об’єкти з моделі системних прецедентів з їх описом і вказівкою взаємозв’язків між класами.

Синтаксис діаграм класів є ефективним засобом структурування вимог до елементів проектованої системи, до їх даних, інтерфейсів, функціональності.

У програмі реалізовані класи, вміст яких зображено на рис. 3.2.

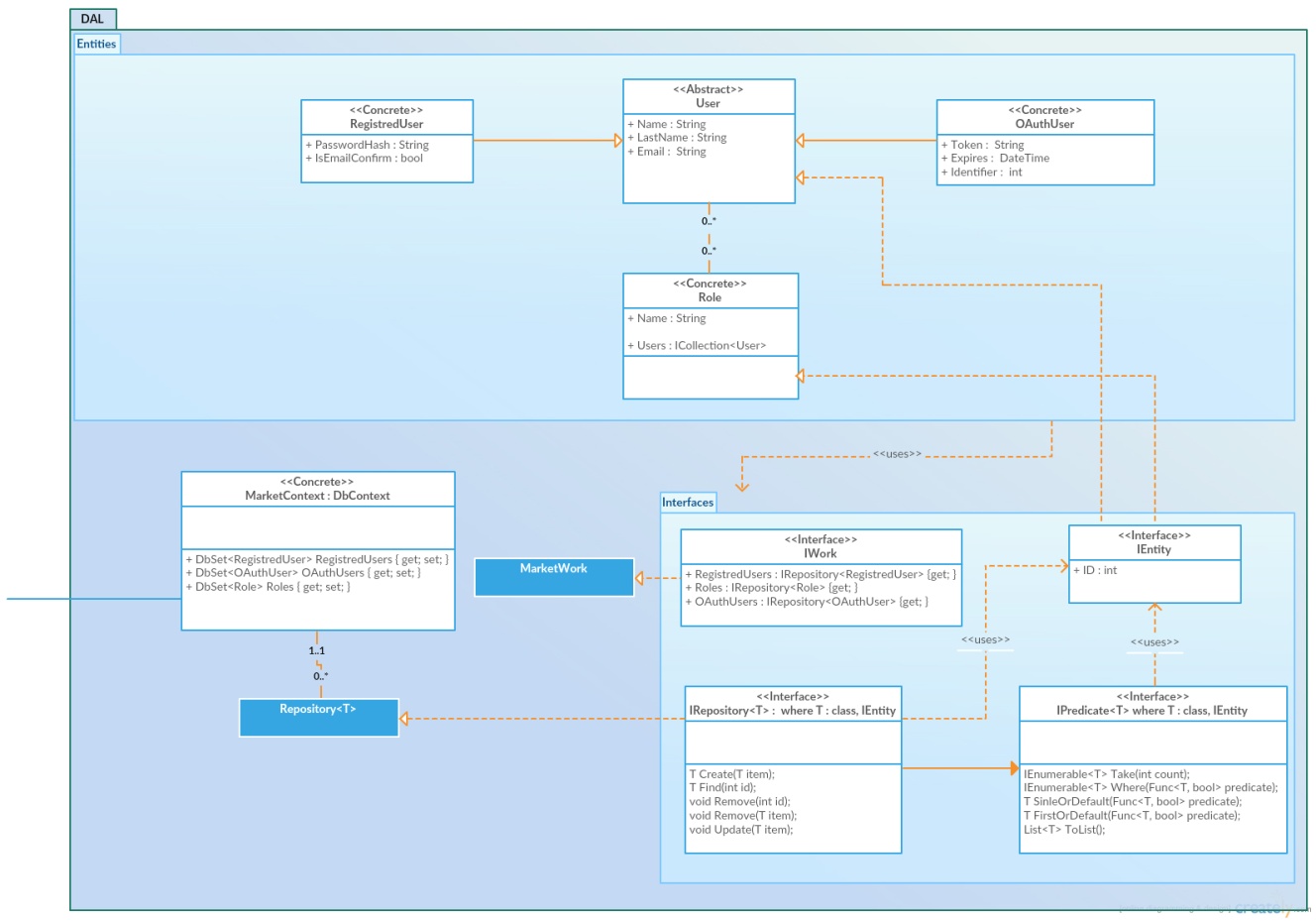


Рисунок 3.2 – Діаграма класів для слою DAL

Основна задача слою DAL (Data Access Layer) це дотримання даних з істочника даних, усунення зайвої інформації та передавання у зручному форматі до верхнього слою.

Цей слой буде використовуватися тільки слоєм вищим за рівнем, тобто BLL. Основна задача BLL (Bussiness Logic Layer) це обробка існуючих моделей, виконання бізнес-логіки, робота з ресурсами, формування ViewModels, та передавання на UI слой зручных для відображання моделей.

Діаграма класів – це тип діаграм, які частіше за все використовуються при моделюванні об’єктно-орієнтованих систем.

Зв’язки у діаграмі класів.

Асоціація показує, що об'єкти однієї сутності (класу) пов'язані з об'єктами іншої сутності.

Якщо між двома класами визначена асоціація, то можна переміщатися від об'єктів одного класу до об'єктів іншого. Цілком припустимі випадки, коли обидва кінці асоціації відносяться до одного і того ж класу. Це означає, що з об'єктом деякого класу дозволено зв'язати інші об'єкти з того ж класу. Асоціація, що зв'язує два класи, називається бінарною. Можна, хоча це рідко буває необхідним, створювати асоціації, що зв'язують відразу кілька класів; вони називаються n-арнимі. Графічно асоціація зображується у вигляді лінії, що з'єднує клас сам з собою або з іншими класами.

Асоціації може бути присвоєно ім'я, яке описує природу відносини. Зазвичай ім'я асоціації не вказується, якщо тільки ви не хочете явно задати для неї рольові імена або у вашій моделі настільки багато асоціацій, що виникає необхідність посилатися на них і відрізняти один від одного. Ім'я буде особливо корисним, якщо між одними і тими ж класами існує кілька різних асоціацій.

Клас, що бере участь в асоціації, грає в ній деяку роль. По суті, це "обличчя", яким клас, що знаходиться на одній стороні асоціації, звернений до класу з іншого її боку. Ви можете явно позначити роль, яку клас грає в асоціації.

Агрегація. Проста асоціація між двома класами, яка відображає структурне відношення між рівноправними сутностями, коли обидва класи знаходяться на одному концептуальному рівні, і ні один не є більш важливим, ніж інший. Але іноді доводиться моделювати відношення типу «частина/ціле», в якому один з класів має більш високий ранг (ціле) і складається з декількох менших за рангом (частин). Ставлення такого типу називають агрегацією; воно зараховане до відносин типу «має» (з урахуванням того, що об'єкт-ціле має кілька об'єктів-частин). Агрегація є окремим випадком асоціації і зображується у вигляді простої асоціації з незафарбованим ромбом з боку «цілого».

Графічно агрегація представляється порожнім ромбом на блоці класу, і лінією, яка від цього ромба до міститься класу.

Композиція - більш суворий варіант агрегації. Відома також як агрегація за значенням.

Композиція має жорстку залежність часу існування екземплярів класу контейнера та примірників містяться класів. Якщо контейнер буде знищений, то весь його вміст буде також знищено.

Графічно представляється як і агрегація, але з зафарбовані ромбиком.

# ВИСНОВКИ

Метою даної роботи була розробка система організації покупок у розумному магазині.

В ході даної роботи проведено докладне дослідження предметної області, обґрунтовано доцільність розробки системи, проведен аналіз існуючих аналогів, описані принципи її роботи та виявлено основні функції.

Реалізація проекту відбувалася за допомогою мов програмування C# та JavaScript, використовувався фреймворк MVC 5 та бібліотек EntityFramework 6, Automapper, NLogger, JQuery. Інтерфейс комунікації між сторонами бекенду та фронтеду побудован за допомогою протоколу передачі HTML та XML-даних. Серверний додаток має REST-ful архітектуру. В якості СУБД виступає MS SQL.

Виконані проектування і реалізація системи і розраховані витрати на її створення. Доведено раціональну обґрунтованість впровадження системи, а також її популярність та необхідність у сучасному світі.

Особливості технології розробки дозволяють при необхідності здійснити перенесення системи на платформи, відмінні від тієї, для якої велася розробка, зі збереженням функціонала.

Розроблена система може використовуватися для професійної роботи фахівців, а саме: інтегрування функціоналу систему та її архітектури під реальні стартами цієї тематики, використовування ідеї з оплати, реалізації взаємодії користувачів системи з магазином тощо.

Потрібно відзначити, що в ході розробки не було реалізовано апаратної частини системи, тому робити повні висновки щодо економічної доцільності неможливо.

У першому виданні додатку впроваджено функціональність для фахівців IT-галузі. З урахуванням гнучкою архітектури, здатністю збільшувати ширину баз даних, система буде здатна підлаштуватися під будь-яку предметну область і надати їй величезну користь. Тобто розглянуте програмне забезпечення може використовуватися людьми за всіми типами професій.

Розроблена система може використовуватися для приклада кодової бази та показу архітектури програмного забезпечення для покращення інших проектів.

# ПЕРЕЛІК ПОСИЛАНЬ

1. Кнут Д. Э. Искусство программирования. [Текст] / Д. Э. Кнут. - М.: Вильямс, 2002. – 301 с.
2. Макконнелл С. Совершенный код. Мастер-Класс [Текст] / С. Макконнел. - М.: Русская Редакция, 2010. – 896 с.
3. Мартин Р. Чистый код. Создание, анализ и рефакторинг [Текст] / Р. Мартин. – СПб.: Питер, 2010. – 464 с.
4. Мацумото Ю. Язык программирования C# [Текст] / Флэнаган Д., Мацумото Ю. - СПб.: БХВ-Петербург, 2008. — 440 с.
5. Приемы объектно-ориентированного проектирования. Паттерны проектирования [Текст] / Э. Гамма, Р. Хелм, Р. Джонсон, Дж. Влиссидес. – СПб.: Питер, 2001. – 368 с.
6. Стефанов С. Javascript Быстрый старт [Текст] / С. Стефанов. – СПб.: Питер, 2017. – 304 с.
7. Фаулер М. Рефакторинг: улучшение существующего кода [Текст] / М. Фаулер. – СПб.: Символ-Плюс, 2003. – 432 с.
8. Фернандес О. Путь .NET. Подробное руководство по созданию приложений в среде Visual Studio [Текст] / О. Фернандес. - СПб.: Символ-Плюс, 2009. – 768 с.
9. Фитцджеральд М. Изучаем C# [Текст] / М. Фитцджеральд. — СПб.: БХВ-Петербург, 2008. — 336 с.
10. Хэнссон Д.Х. Гибкая разработка веб-приложений в среде C# [Текст] / Томас Д., Хэнссон Д.Х. - СПб.: Питер, 2008. – 716 с.