



Приватний заклад вищої освіти
Одеський технологічний університет «ШАГ»
Кафедра інформаційних технологій та фундаментальної підготовки

АВТОРЕФЕРАТ
випускної кваліфікаційної роботи бакалавра

**« Розробка веб-платформи для онлайн-комерції.
Клон Amazon»**

на здобуття бакалаврського рівня вищої освіти
зі спеціальності «122 Комп'ютерні науки»

Виконавці проекту

№	П.І.Б.	Група
1	Бойчев Олександр Юрійович	КН-П-181
2	Нікулов Олексій Леонідович	КН-П-181

Автор звіту
Бойчев Олександр Юрійович

РЕЗУЛЬТАТИ ЕКСПЕРТИЗИ

Експерт	П.І.Б.	Оцінка	Підпис
Науковий керівник	Черкезян Крістіне Арутюнівна		
Рецензент	доц. Суліма М. М.		
ДЕК (захист проекту)	доц. Голова Е. К.		

Одеса – 2022
АНОТАЦІЯ

УДК 004.9

Д-30

Бойчев О. Ю. Система узгодження взаємодії користувачів з урахуванням потреб товарообміну: Автореферат випускної кваліфікаційної роботи на здобуття бакалаврського рівня вищої освіти зі спеціальності «122 Комп'ютерні науки». - Одеса: ОТУШ, 2022.

Робота присвячена спрощенню взаємодії між споживачем та покупцем, для збуту товару, та задоволенню потреб споживача, незважаючи на відстань, та умов двох сторін.

Своєчасність цієї роботи регламентується тим, що зараз Інтернет-магазини працюють без перерв, вихідних та свят. На сайті, на відміну від звичайних магазинів, немає черг. Купівлю будь-якого обраного товару можна відкласти, при цьому вся інформація про нього зберігатиметься в обліковому записі. При виборі товару можна читати відгуки інших покупців і на їх основі робити свій вибір.

Переваги інтернет-магазину:

1. Низькі витрати
2. Знижені вимоги до персоналу
3. Немає географічних обмежень
4. Потенційний дохід у режимі 24/7
5. Продаж цифрових товарів
6. Ремаркетинг: зробити із відвідувача клієнта
7. Просування та залучення клієнтів
8. Прискорений процес покупки

ЗМІСТ

ВСТУП _____	3
Технічне завдання до проекту _____	4
РОЗДІЛ 1. Загальна характеристика системи _____	8
РОЗДІЛ 2. Реалізація серверної частини _____	9
РОЗДІЛ 3. Результати випробувань _____	12
ВИСНОВКИ _____	13
ПЕРЕЛІК ВИКОРИСТАНИХ ДЖЕРЕЛ _____	14

СИСТЕМА УПРАВЛІННЯ ЕЛЕКТРОННИМИ «P2P» ПОСЛУГАМИ (за прикладом сервісу Amazon)

Загальний опис.

Програмне забезпечення (ПЗ) призначене для створення розподіленої системи узгодженого управління контентом, який передбачає «P2P» (person-to-person) взаємодію. Функції системи мають забезпечити повний життєвий цикл управління контентом: створення, актуалізація, модифікація та знищення, а також встановлення «P2P» зв'язків між надавачем та користувачем послуги. Головна особливість системи полягає у тому, що наявні у неї інформаційні одиниці орієнтовані на визначену групу споживання (є таргетованими).

Призначення:

ПЗ орієнтується на створення контейнеру управління інформаційними одиницями на основі аналізу їх цільового призначення (таргетування). Визначальною функцією системи є утворення «P2P» комунікації, отримання зворотного зв'язку від кожної зі сторін взаємодії. ПЗ створюється за прикладом сервісу Amazon, у той же час має складатись за принципом універсальності, мати змогу змінити цільове призначення.

Вимоги до функціоналу. Обов'язкова частина

Головна сторінка

Стартовий інтерфейс ПЗ (головна сторінка або активність) має за мету популяризацію системи. На ньому має бути забезпечено відображення популярних показників як-то загальна кількість користувачів та інформаційних одиниць, вибірка з найбільш популярного контенту та нових дописів, зовнішня реклама. Значний акцент має бути зроблений на детальному пошуку контенту за принципом таргетування. Головна сторінка обмежує функції системи для незареєстрованих користувачів лише переглядом та пошуком, інші функції надаються у разі авторизації користувача.

Реєстрація користувачів:

Користувач ПЗ має змогу зареєструватись як за допомогою авторизаційних даних (логіну/паролю), так і за допомогою облікових записів популярних соціальних мереж (Google, Facebook, тощо).

При реєстрації обов'язково зазначається засіб оперативного контактування (ЗОК) - телефон та електронна пошта. На тестовому етапі, за умови ускладнення замовлення послуг телефонії чи СМС, можливе використання лише електронної пошти. Засіб оперативного контактування при реєстрації підлягає верифікації через відповідь на тестовий виклик чи надісланий лист, або введення коду, пересланого через повідомлення довільного типу. Рекомендується використовувати ЗОК у якості авторизаційного логіну користувача.

За умови що користувач забув пароль, ЗОК має бути використаний для його відновлення. Схема відновлення паролю жорстко не регламентується.

Також передбачається наявність у системі користувачів з додатковими правами: модераторів та адміністраторів. Реєстрація кореневого адміністратора системи має бути реалізоване при інсталяції системи, наступним користувачам змінюють права наявні адміністратори.

Користувачі системи поділяються на надавачів та споживачів послуг. При реєстрації надавача послуги вимагається додаткова перевірка його прав та ліцензій на

можливість надання послуг чи ведення електронної комерції. Для користувачів усіх категорій вимагається формальна валідація усіх реєстраційних даних (відповідність загальноновживаним або власним шаблонам).

Особисті кабінети користувачів:

Особисті кабінети різних груп користувачів - надавачів та споживачів послуг - мають забезпечувати різну функціональність. Дозволяється на початковому етапі (до впровадження механізмів перевірки ліцензій надавачів послуг) функції надавача послуг делегувати користувачам з підвищеним статусом керування (адміністраторам) через що кабінет надавача послуги може виконуватись як панель адміністрування контенту.

Особистий кабінет надавача послуги має забезпечити необхідні засоби для

- Управління даними та метаданими (пропозицій та їх описів, цільових груп, тощо) - створення, редагування, видалення (блокування), перегляд.
- Перегляд поточної активності щодо власних пропозицій
- Контакткування з користувачами-споживачами, що розміщують замовлення на контент, за допомогою ЗОК або через внутрішні засоби системи (чат або месенджер)

Особистий кабінет споживача послуги має забезпечити необхідні засоби для

- Встановлення належності до цільових груп
- Пошуку контенту інших користувачів за метаданими, групування та фільтрування результатів за різними ознаками, наявними у знайденому контенті, а також за цільовою групою. Для об'ємних результатів має бути забезпечена пагінація.
- Контакткування з іншими користувачами-надавачами послуг, що розміщують пропозиції, за допомогою ЗОК або через внутрішні засоби системи (чат або месенджер)

Також засобами особистих кабінетів має бути передбачено можливість

- Зміни персональних даних, зокрема, платіжних засобів. У разі зміни важливих даних, що раніше проходили верифікацію, вимагається їх повторна верифікація.
- Зворотного зв'язку - встановлення власної оцінки (лайк або бал) для спожитої або наданої послуги, написання відгуку (коментаря) або рекомендації за фактом взаємодії.

Вимоги до функціоналу. Опціональна частина

Модерація

Для забезпечення дотримання загальноприйнятих норм спілкування, а також з метою недопущення публікування контенту, який містить заборонені компоненти (ненормативні висловлювання, заклики до протиправних дій, приниження чи дискримінація, тощо), усі інформаційні одиниці, а також створені користувачами метадані (маркери) мають в обов'язковому порядку пройти погодження (модерацію). Право надати погодження належить кореневому адміністратору системи, а також модераторам. При погодженні інформаційна одиниця має зберігати ідентифікатор користувача, який надав погодження.

Привілейовані функції

З метою покращення економічної ефективності проєкту передбачається наявність додаткових функцій системи, що доступні лише за умови платної підписки (або

внутрішніх досягнень) користувачів. До таких функцій можна віднести відображення ЗОК інших користувачів (для прямого контактування), розширені засоби фільтрації та групування. Також можливе встановлення обмеження щодо кількості опублікованих одиниць контенту або замовлень.

Архітектура

ПЗ має бути спроектоване за клієнт-сервеною розподіленою архітектурою з чітко відокремленими структурними вузлами системи:

Mockup - візуальна модель інтерфейсу користувача із зазначенням зв'язків і переходів між різними частинами інтерфейсу.

Backend - серверне програмне забезпечення, що забезпечує збереження та оброблення поточних даних, взаємодію з базою даних. Надає програмний інтерфейс взаємодії (API) з іншими вузлами.

Web App - додаток, доступний з веб-простору (сайт), що забезпечує візуальний інтерфейс користувачів усіх категорій

Mobile App (опціонально) - додаток, що встановлюється на мобільних пристроях (смартфон, планшет). Дана частина виконується за наявності у команді проекту трьох і більше програмістів.

.

Усі модулі ПЗ (окрім візуальної моделі Mockup) повинні обов'язково включати до свого складу засоби

- тестування функціональності та модульні тести,
- розгортання на новому пристрої (інсталяції) із встановленням початкових таблиць у БД та кореневого користувача-адміністратора

Технології

Зберігання вихідних кодів ПЗ повинно здійснюватись за допомогою системи контролю версій (вибір конкретної системи узгоджується групою проекту). Структура файлів та каталогів повинна відбивати архітектуру ПЗ, на зразок:

- root
 - Backend
 - Web
 - Mobile
 - Mockup (вихідні файли або посилання на зовнішній проєкт)

Для кожного з модулів рекомендується (з можливістю часткової зміни) наступні технології створення:

Backend - .NET технології із хмарним розміщенням (Azure),

Web - NET технології (ASP, Razor Pages) або Node JS, React чи Angular

Mobile - Java, або Kotlin, або Flutter

Mockup - визначається ментором з дизайну

Команда проєкту

Для виконання поставлених завдань передбачається робота команди розробників окремо за кожним з модулів проєкту.

Мінімальна команда проєкту - два учасники, у такому складі один учасник виконує модуль Backend та програмну частину Web, другий - модель Moscur та розміткову частину Web.

Рекомендована команда проєкту - три учасники, у такій команді кожен з учасників відповідає за одну з частин: Moscur, Backend, Web.

Дозволяється включення до команди більшої кількості учасників, у такому разі модуль Mobile App набуває обов'язкової реалізації. Також модуль Moscur може бути поділений на моделі Web та Mobile частин.

Звітність

За результатами виконання завдань із створення ПЗ кожен з учасників команди проєкту складає власний (індивідуальний) звіт, що складається з практичної частини та автореферату. Практична частина є прямим результатом роботи і подається у вигляді посилання на репозиторій (чи інший URL) або у вигляді архіву на довільному носіїві даних. Вихідний код повинен оформлятися у відповідності до загальноприйнятих стандартів, на зразок стандартів GNU <http://www.gnu.org/prep/standards/>, у тому числі з додержанням вимог до коментування та документування коду.

Автореферат містить описову частину, у якій розкриваються основні концепти проєкту, деталізуються завдання до виконавця, встановлюється актуальність роботи, її об'єкт та предмет, обґрунтовується вибір технологій та відзначається шлях виконання та ступінь досягнення поставлених задач, посилання на джерела інформації.

У відповідності до технічного завдання (ТЗ) система була спроектована за архітектурою “MVC”(Model-View-Controller). Система була розподілена на 3 частини:

- UX/UI
- Backend
- Frontend

Частина UX/UI відповідає за дизайн проекту. Дизайн формулює із себе набір моделей які створюють інтерфейс для взаємодії з користувачем щоб користувачеві було інтуїтивно зрозуміло та візуально приємно контактувати з ним. При створенні та проектуванні інтерфейсів були використані всі основні принципи UX/UI дизайну.

Частина “Backend” була створена як система для роботи з даними, а саме їх обробці та зберіганню протягом усього життєвого циклу ПЗ.

Ця частина включає такі розділи як: база даних і програмний інтерфейс програми(API) яка надає можливість користувачеві взаємодіяти з усіма функціями програми. Частина «Backend» була розроблена на основі такого програмного забезпечення як ASP.NET і СУБД MS SQL Server. Під час створення програми були використані такі додаткові пакети як “Bootstrap”, “SPA Proxy” та “ER SQL Server”.

Частина «Frontend» дає користувачеві отримати зручний доступ до всіх функцій програми надаючи йому добре опрацьований візуальний інтерфейс.

РОЗДІЛ 2

РЕАЛІЗАЦІЯ СЕРВЕРНОЇ ЧАСТИНИ

Серверна частина інформаційної системи реалізована з використанням наступних технологій:

- Засоби розроблення: мова програмування C#10 (ASP.Core .net 6.0) та середовище Visual Studio
- Додаткове ПЗ та засоби: Docker, Visual Studio Code, Azure Data Studio, Postman

У складі серверної частини передбачено та реалізовано наступні функції:

- Авторизація клієнта,
- Реєстрація нових клієнтів

При відкритті головної сторінки користувачем випадає список найпопулярніших товарів, список товарів які були нещодавно переглянуті користувачем, а також деякі категорії. Користувач може переходити за категоріями, переглядати товари та їх опис, так само користувачеві видає список схожих товарів. Для того, щоб користувач міг придбати товар, йому необхідно зареєструватися. При реєстрації користувач повинен обов'язково вказати свої дані, зокрема номер телефону та його пошту для оповіщення користувача про якісь події.

Перш ніж користувач пройде реєстрацію, його дані проходять початкову валідацію. Авторизація користувача відбувається шляхом введення його логіну та пароля який він вказав при реєстрації. При реєстрації або авторизації користувача відбувається взаємодія з "Backend" частиною для перевірки існування користувача.

Система була спроектована таким чином, щоб клієнтська частина могла зручно передавати дані на серверну частину. Нижче наведено метод CreateOrUpdate який відповідає за створення або видалення користувача. Передавання даних здійснюється шляхом медота POST.

[HttpPost]

```
public async Task CreateOrUpdate(Clients data)
{
    var res = await GetByld(data.Id);
    if (res == null)
    {
        await _applicationContext.AddAsync(data);
    }
    else
    {
        var local = _applicationContext.Set<Clients>().Local.FirstOrDefault(entry
=> entry.Id.Equals(data.Id));
        if (local != null)
            _applicationContext.Entry(local).State = EntityState.Detached;
        _applicationContext.Entry(data).State = EntityState.Modified;
    }
}
```

```

    }
    await _applicationContext.SaveChangesAsync();
}

```

В основному користувач взаємодіє зі сторінкою товарів, користувач може залишати відгуки, ставити запитання продавцю, переглядати повну інформацію про товар, додавати товар у кошик. Для цього було створено кілька класів, які реалізують всю вище перераховану функціональність.

```

using Microsoft.EntityFrameworkCore;
using Taon.Data;
using Taon.Data.Models;

namespace Taon.Repositories
{
    public class CAQRepository
    {
        private readonly ApplicationDbContext _applicationContext;

        public CAQRepository(ApplicationContext context)
        {
            _applicationContext = context;
        }

        public async Task<List<CAQ>> GetAll()
        {
            return await Task.Run(() =>
            {
                IQueryable<CAQ> data = _applicationContext.CAQ;
                var res = data.Select(el => el).ToList();
                return res;
            });
        }

        public async Task<CAQ> GetById(int id)
        {
            return await Task.Run(() =>
            {
                IQueryable<CAQ> data = _applicationContext.CAQ;
                var res = data.FirstOrDefault(el => el.Id == id);
                return res;
            });
        }
    }
}

```

```

public async Task CreateOrUpdate(CAQ data)
{
    var res = await GetById(data.Id);
    if (res == null)
    {
        await _applicationContext.AddAsync(data);
    }
    else
    {
        var local = _applicationContext.Set<CAQ>().Local.FirstOrDefault(entry =>
entry.Id.Equals(data.Id));
        if (local != null)
            _applicationContext.Entry(local).State = EntityState.Detached;
        _applicationContext.Entry(data).State = EntityState.Modified;
    }
    await _applicationContext.SaveChangesAsync();
}

public async Task Delete(int id)
{
    var item = await _applicationContext.Set<CAQ>().FindAsync(id);
    await Task.Run(() =>
    {
        _applicationContext.Set<CAQ>().Remove(item);
    });
    await _applicationContext.SaveChangesAsync();
}

}
}

```

Для того, щоб повністю ознайомитися з проектом, перейдіть на <https://bitbucket.org/AlexeyNikulov/treemix/src/master/>

РОЗДІЛ 3 РЕЗУЛЬТАТИ ВИПРОБУВАНЬ

Результатом компіляції коду програми є з'єднання серверної та клієнтської частини створюється файл консольної програми, який запускає сервер. Далі ми можемо розмістити наш проект на будь-якому зручному хостингу, в даному

випадку сервер був розміщений на Azure. Всі основні тести були проведені в додатку Postman для зв'язки з клієнтом, для перевірки коректності даних була задіяна Azure Data Studio. Робота користувача із сайтом починається з відкриття головної сторінки. Користувач може переходити на доступні для нього сторінки. Під час того як користувач проходить реєстрацію або авторизацію йому на пошту або телефон приходить повідомлення з кодом твердження для того, щоб користувач міг вільно завершити процес реєстрації або авторизації.

Після реєстрації або авторизації користувачеві видається доступ до купівлі товару, можливість залишити feedback після купівлі товару або поставити запитання продавцеві щодо обраного товару. Після того, як користувач завершив оформлення замовлення, система їх фіксує і відправляє всі дані в базу даних (за це відповідає частина Backend). Подальші дії користувач може здійснювати через особистий кабінет. В особистому кабінеті користувач може робити такі речі як: редагувати свої дані, переглядати статус своїх замовлень, отримує доступ до системи особистих повідомлень, можливість видалення свого профілю, вводити код gift карт, переглядати історію пошуку. Всі функції, які були описані в технічному завданні, були успішно реалізовані і протестовані. Для того щоб повністю ознайомитися з усіма можливостями клієнтської частини можна перейти в репозиторій

<https://bitbucket.org/AlexeyNikulov/treemix/src/master/>

ВИСНОВКИ

На основі ТЗ було розроблено проект, який відповідає всім вимогам. При створенні проекту було проведено детальний аналіз предметної галузі та розроблено план розробки системи. Розроблено максимально зручну та

доступну роботу потенційного клієнта, доступні та зрозумілі діалогові вікна. Розбиті за категоріями товари, що мають докладний опис з ілюстраціями, гнучка можливість пошуку за різними критеріями, перегляд «новинок» та товарів, що найбільше продаються.

Система була спроектована, реалізована та протестована з урахуванням основних потреб користувачів. Система була реалізована за допомогою архітектурного патерну MVC та складається з 3-х частин:

- UX/UI
- Backend
- Frontend

Система була успішно встановлена на хостинг та успішно протестована на доступну функціональність. В результаті тестування системи підтвердили загальну працездатність створеної системи.

ПЕРЕЛІК ВИКОРИСТАНИХ ДЖЕРЕЛ

- 1 <https://ru.weblium.com/blog/plyusy-i-minusy-internet-magazinov/> - переваги інтернет-магазину з сторони бізнесу
- 2 <http://www.gnu.org/prep/standards/> - зразок стандартів GNU
- 3 <https://bitbucket.org/AlexeyNikulov/treemix/src/master/> - посилання на проект