

# Розробка сервіс-орієнтованої системи онлайн-замовлень на основі технології Windows Communication Foundation

магістерська кваліфікаційна робота  
спеціальність 8.05010301 «Програмне забезпечення систем»

Виконав:

ст. гр. 1ПЗ-14мі Стрельбіцький М.Ю.

Науковий керівник:

к.т.н., доц. Майданюк В.П.

## Актуальність теми

- Стрімкий розвиток мережі Інтернет сприяє розвитку електронної комерції. Це дає поштовх до розробки спеціалізованих веб-ресурсів для реалізації продукції потенційним споживачам.
- Для того, щоб один постачальник міг одночасно працювати із декількома сайтами, а на одному сайті могли розміщувати інформацію декілька постачальників, необхідно розробити автоматизовану систему, яка надає кожній зі сторін службу, або сервіс.

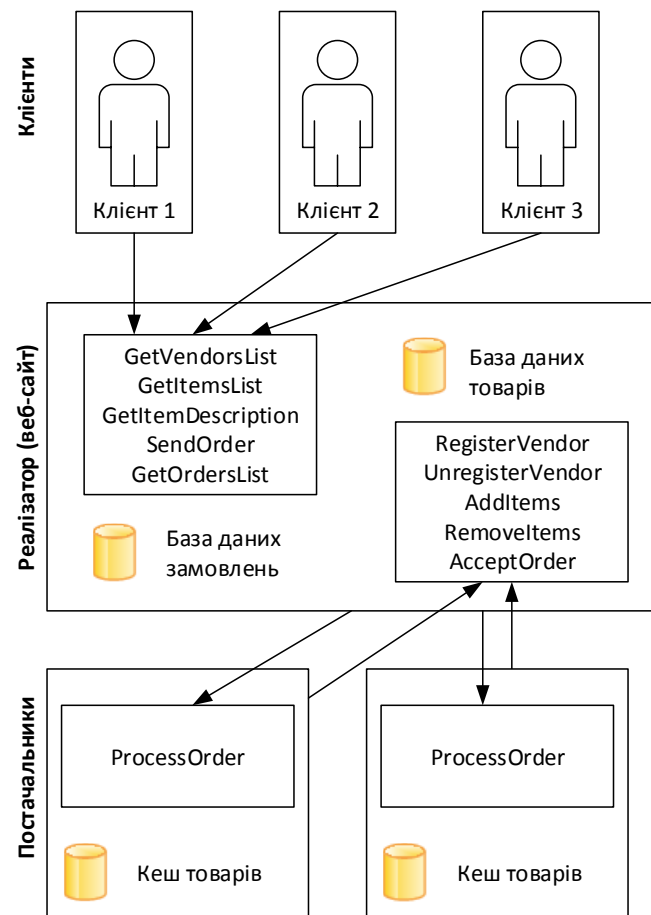
# Мета роботи, об'єкт та предмет дослідження

- *Метою роботи* є підвищення рівня автоматизації процесів електронної торгівлі шляхом розробки програмної системи для автоматизованого оформлення та обробки замовлень у режимі онлайн.
- *Об'єктом досліджень* є сервіс-орієнтовані системи та інформаційні процеси, пов'язані із обміном даними між компонентами таких систем.
- *Предметом досліджень* є сервіс-орієнтована архітектура програмних продуктів та її реалізація у вигляді системи онлайн-замовлень.

# Наукова новизна та практична цінність отриманих результатів

- *Наукова новизна* роботи полягає в наступному:
  - запропоновано використання сервіс-орієнтованого підходу для реалізації системи онлайн-замовлень, який зменшує часові витрати на розробку і впровадження, підвищує продуктивність роботи та прискорює інтеграцію програмного комплексу;
  - отримав подальший розвиток метод подання сайту у вигляді єдиної сторінки, що спрощує навігацію та підвищує інтерактивність.
- *Практична цінність* полягає в наступному:
  - розроблено програмний комплекс для автоматизації процесів оформлення та обробки онлайн-замовлень;
  - принципи, покладені в основу розробки, можна використовувати при розробці аналогічних систем для товарів будь-якого типу.

# Розробка архітектури системи

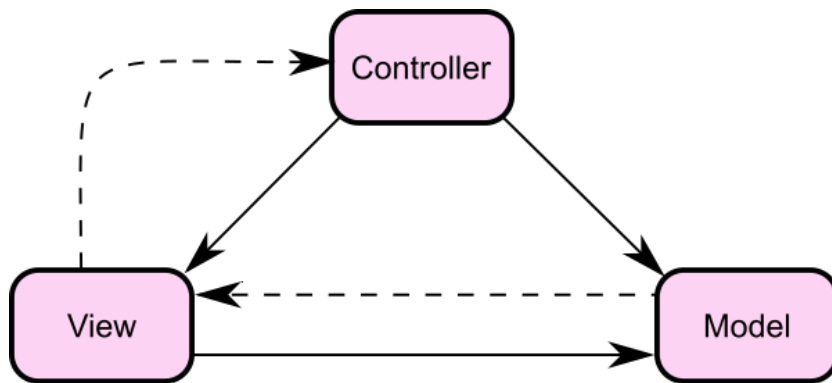


# Вибір сервіс-орієнтованої платформи для розробки

<b>.NET Framework</b>	<b>Java Enterprise Edition</b>
<ul style="list-style-type: none"><li>• єдиний стек технологій;</li><li>• Windows-орієнтованість;</li><li>• орієнтація на використання у невеликих проектах;</li><li>• уніфікованість бібліотек платформи;</li><li>• якісна підтримка та висока ступінь інтеграції при розробці веб-сервісів.</li></ul>	<ul style="list-style-type: none"><li>• гнучкість та розширюваність API Java EE, але низький рівень уніфікації;</li><li>• мультиплатформеність;</li><li>• орієнтація на використання у великих проектах;</li><li>• запуск додатків Java EE потребує використання дорогих серверів;</li><li>• хороша підтримка ORM-систем.</li></ul>

# Розробка веб-сайту: шаблон Model-View-Controller

Шаблон MVC забезпечує поділ компонентів веб-сайту на три складові:



- *Модель (Model)* зберігає дані та надає інтерфейс доступу до них;
- *Представлення (View)* відображає дані кінцевому користувачеві;
- *Контролер (Controller)* реагує на дії користувача та повідомляє Модель про зміни.

# Розробка веб-сайту: моделі

- У якості моделей на сайті виступають об'єкти доступу до даних, які отримуються із баз даних: інформація про наявні моделі автомобілів та виконані запити. Вони мають назву Data Access Layers (DAL), або шар доступу до даних.
- Шар доступу до даних надає функції для додавання, зміни або редагування даних.
- Дані зберігаються у файлах формату XML, якими зручно керувати за допомогою технології LINQ to XML.



# Розробка веб-сайту: представлення

- У випадку ASP.NET MVC представлення – це шаблон HTML-сторінки (якщо необхідно, зі вставками коду мовою C#), який компілюється у готову html-сторінку та скрипти мовою JavaScript.
- Веб-сайт виконано як односторінковий (single-page).
- Для реалізації динамічного контенту та оперативної синхронізації контенту із постачальником були використані такі мови, технології та інструменти: HTML, CSS, JavaScript, Twitter Bootstrap, jQuery, AJAX, ASP.NET SignalR, Knockout.

# Розробка веб-сайту: контролер

- У ASP.NET є два типи контролерів: контролер MVC та API-контролер.
- Кожному представленню відповідає свій контролер MVC. Оскільки сайт реалізовано як односторінковий, його представленню відповідає контролер початкової сторінки.
- API-контролери використовуються для реалізації сервісів Web API, які будуть розглянуті далі.

# Приклад веб-сайту реалізатора

Car Shop Orders history Current order


Vendors

Ford

**Honda**


Cars

Accord Civic **CR-V** Pilot



## Honda CR-V

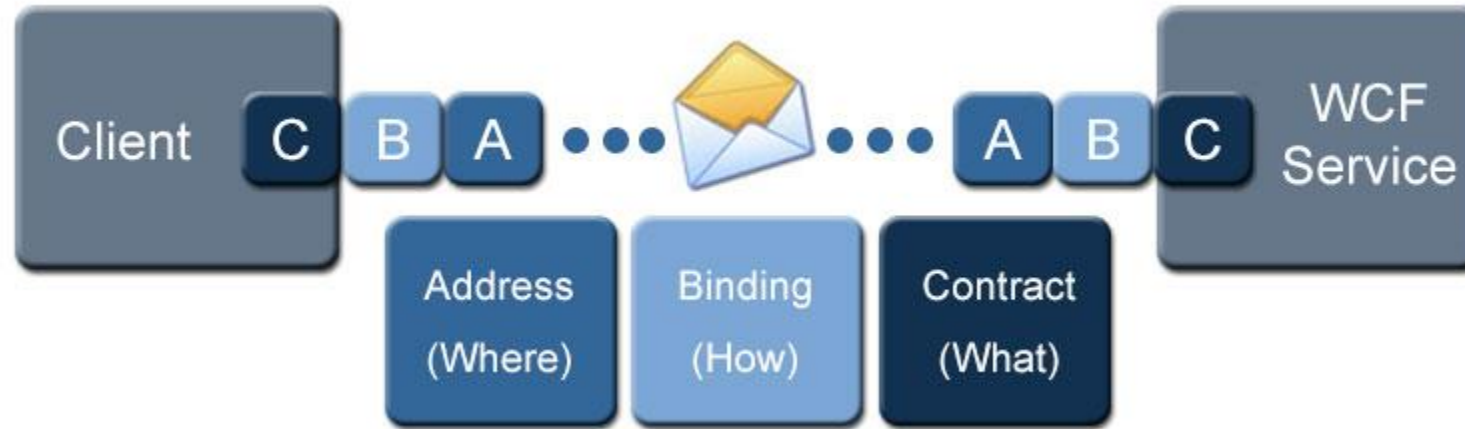
The Honda CR-V is a compact crossover, manufactured since 1995 by Honda. It was loosely derived from the Honda Civic. There are discrepancies as to what "CR-V" stands for. Most state that it stands for "Comfortable Runabout Vehicle", although sales literature in UK has reportedly made references to "Compact Recreational Vehicle". Honda began producing the CR-V in Sayama, Japan, and Swindon, UK, for worldwide markets, adding North American manufacturing sites in East Liberty, Ohio, in 2007; El Salto, Jalisco, Mexico, in late 2007; and Alliston, Ontario, Canada, in 2012. The CR-V also is produced in Wuhan (Hubei province) for the Chinese market by the Dongfeng Honda Automobile Company, a joint venture with Dongfeng Motor Corporation. The CR-V is Honda's smallest SUV other than the related Element sold in Canada and the United States, and the HR-V sold in Europe. In size, the CR-V slots between the Element and Pilot, but due to the Element's discontinuation during the 2011 model year, the CR-V has regained its status as Honda's entry-level SUV.

Available colors	Price
	\$26480

# Розробка сервісів сайту

- Веб-сервіс – програмна система зі стандартизованим інтерфейсом, яка ідентифікується уніфікованим ідентифікатором ресурсів (URI).
- Для розробки сервісів сайту використано технології Windows Communication Foundation (WCF) та Web API.

# Розробка WCF-сервісу сайту



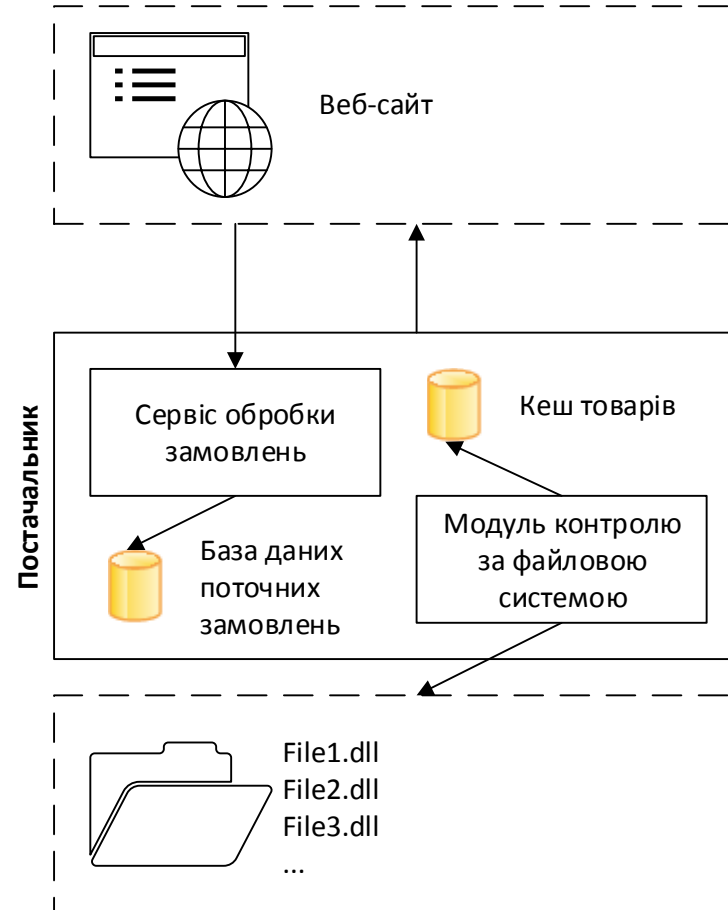
Для використання WCF-сервісу необхідно створити кінцеву точку (Endpoint), яка складається із таких компонентів:

- Адреса (Address) – URL-ідентифікатор сервісу, наприклад <http://car-shop.com/CarShopService.svc>;
- Прив'язка (Binding) – описує протокол передачі даних (в нашому випадку – HTTP) та механізми захисту, якщо вони є;
- Контракт (Contract) – інтерфейс доступу до даних. За допомогою сервісу можна додавати або видаляти автомобілі з сайту.

# Розробка сервісів Web API

- Web API – ще один тип веб-сервісів, де основний акцент зроблено на простішому типі комунікації – Representational State Transfer (REST). Виклик методів виконується через GET та POST-запити.
- Для передачі даних у вигляді об'єктів використовується формат JSON.
- У шаблоні ASP.NET MVC такі сервіси реалізуються у вигляді API-контролерів:
  - InformationController – отримання даних про постачальників і товари;
  - RegisterController – реєстрація постачальників та її скасування;
  - OrderController – надсилання та прийом виконаних замовлень.

# Розробка структури клієнтського додатку постачальника



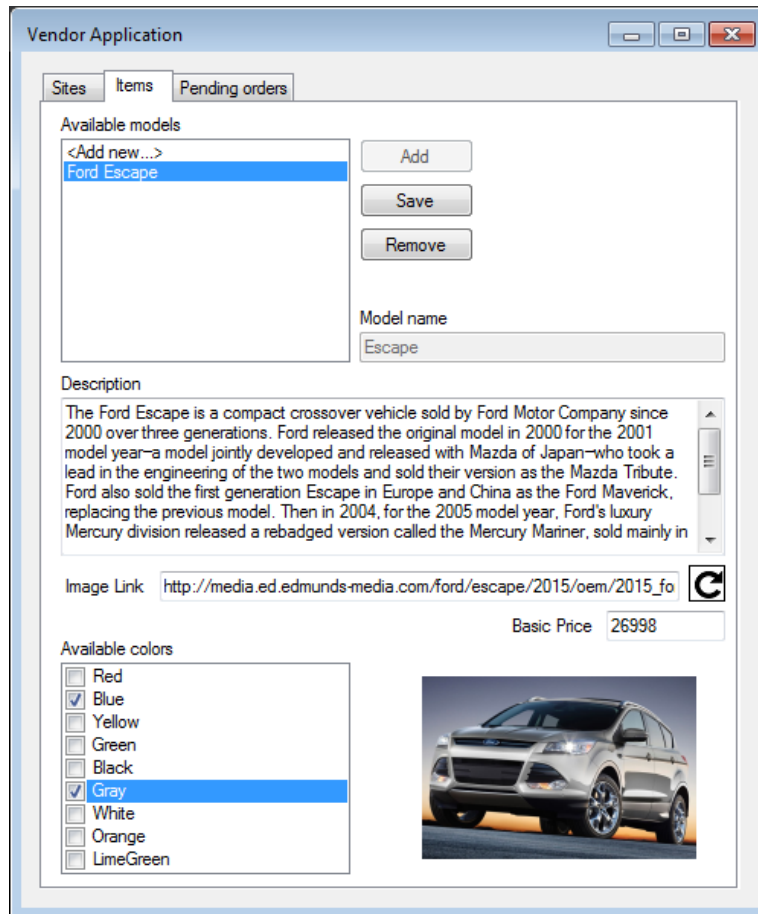
# Розробка WCF-сервісу постачальника

Для використання WCF-сервісу створюємо кінцеву точку (Endpoint):

- Адреса (Address) – URL-адреса сервісу, наприклад <http://vendor-site.com/VendorService>;
- Прив'язка (Binding) – як і на сервісі сайту, використовується протокол HTTP;
- Контракт (Contract) – інтерфейс доступу до даних. Сервіс постачальника надає можливість додавати замовлення до черги на виконання.



# Приклад клієнтського додатку



# Тестування роботи системи

- Виконано автономне тестування веб-сайту та додатку постачальника, а також комплексне тестування системи.
- Веб-сайт коректно обробляє запити користувача, додає обрані товари в кошик та коректно відображає контент. Адаптивний дизайн спрацьовує при зменшенні розмірів вікна браузера.
- Додаток коректно опрацьовує вхідні дані користувача, працює із файловою системою, додаючи та видаляючи бібліотеки.
- Додаток своєчасно реагує на надходження товарів від сайту, а після їх обробки відповідні зміни відображаються на сайті.

## Економічна частина

У ході виконання економічних розрахунків було отримано такі результати:

- абсолютна ефективність вкладених інвестицій складає 62644 грн.;
- відносна ефективність розробки – 71%;
- термін окупності – 1 рік 5 місяців.

# Висновки

- Виконано розробку архітектури програмного комплексу. За основу обрано сервіс-орієнтований підхід для реалізації розподіленої програмної архітектури.
- У якості базової обрано платформу розробки .NET Framework. Основною мовою розробки обрано C#, а сама розробка виконувалась у середовищі Microsoft Visual Studio.
- Виконано розробку веб-сайту з використанням шаблону ASP.NET MVC. Для дизайну сайту використано ряд інструментів та технологій: HTML, CSS, JavaScript, Twitter Bootstrap, jQuery, AJAX, ASP.NET SignalR, Knockout. Розроблено ряд WCF-сервісів та Web API сервісів для віддаленого використання функціоналу сайту.
- Розроблено клієнтський додаток постачальника у вигляді Windows Forms Application. Розроблено сервіс для віддаленого користування зареєстрованими сайтами.
- Виконано компонентне та комплексне тестування системи, яке показало її повну працездатність.
- Розрахунок економічних показників повністю підтвердив доцільність розробки. Термін окупності розробки склав 1,4 роки, що вказує на швидку окупність вкладених інвестицій.

Дякую за увагу!