

РЕАЛІЗАЦІЯ ЛОКАЛІЗАЦІЇ ВЕБ ТА МОБІЛЬНИХ ДОДАТКІВ НА BACK-END

Вінницький національний технічний університет

Анотація

У статті описано актуальність локалізації, а також існуючі способи та шляхи її реалізації на back-end-i. Розглянуто дослідження та статистичні дані світових науково-популярних джерел. Запропоновано універсальну платформу для локалізації веб та мобільних додатків на основі багатопшарової клієнт-серверної архітектури. Використано новітні технології та інструменти, що в свою чергу дозволило зробити розроблювану систему легкою для масштабування, надійною та кросплатформеною.

Ключові слова: локалізація, переклад, глобалізація, сервіс, лінгвістика, програмування, платформа, мережа, бекенд.

Abstract

The article analyzes the problem of localization describes its relevance at present, as well as known methods and ways of its implementation on the back-end. The latest researches and statistical data of world popular scientific sources are considered. A universal platform for localization of the web and mobile applications based on the multilayer client-server architecture is proposed. Used the latest technologies and tools, which in turn allowed the developed system to be easy to scale, reliable and cross-platform.

Keywords: localization, translation, globalization, service, linguistics, programming, platform, network, back-end.

Вступ

У ході розвитку всесвітньої мережі Інтернет створюється все більше і більше сайтів та додатків у кожному куточку світу. Але, на жаль, лише мешканці цих куточків можуть знати про них, адже переважно лише вони знають мову, на якій розроблено інтерфейс даного сайту чи додатку.

Локалізація сайту – це переклад контенту та інтерфейсу на мову цільової аудиторії у будь-якій країні, що забезпечує глобальний зв'язок з клієнтами по всьому світу, тому що набагато простіше впливати на вибір людей за допомогою їх рідної мови, полегшуючи розуміння і сприяючи зближенню людини, що надає послуги, з клієнтом.

Актуальність локалізації

На даний момент, кількість нативних носіїв мови у світі не співпадає із кількістю продуктів цією мовою, що видно з діаграм, представлених на рисунках 1 і 2. На це є декілька причин:

- розвиненість країни, де спілкуються цією мовою
- складність мови
- освіченість населення і т.д.

На жаль, це призводить до труднощів та незручностей. На дану тему було проведено багато досліджень, з метою визначити на скільки комфортніше сприймати інформацію своєю мовою. Ось деякі висновки з них:

- 80% користувачів, які користуються мобільними додатками не англomовні [1];
- локалізовані версії сайтів та додатків мають на 42% більший коефіцієнт кліків та на 22% більше переходів ніж одномовні [2];
- 72% покупців стверджують, що вони здійснюють покупки лише у тих магазинах, де є опис товару їхньою мовою (Common Sense Advisory) [3]
- 56% покупців вважають наявність перекладу на їхню мову важливішою за ціну продукту (Common Sense Advisory) [3]

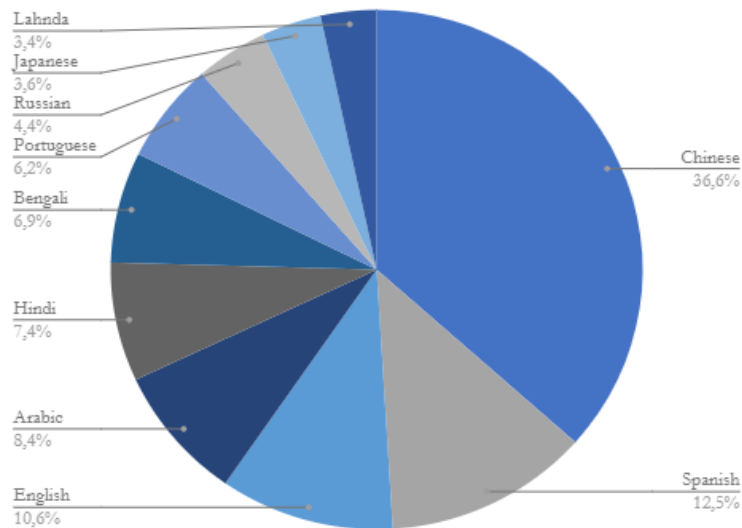


Рис. 1. Популярність мови у світі

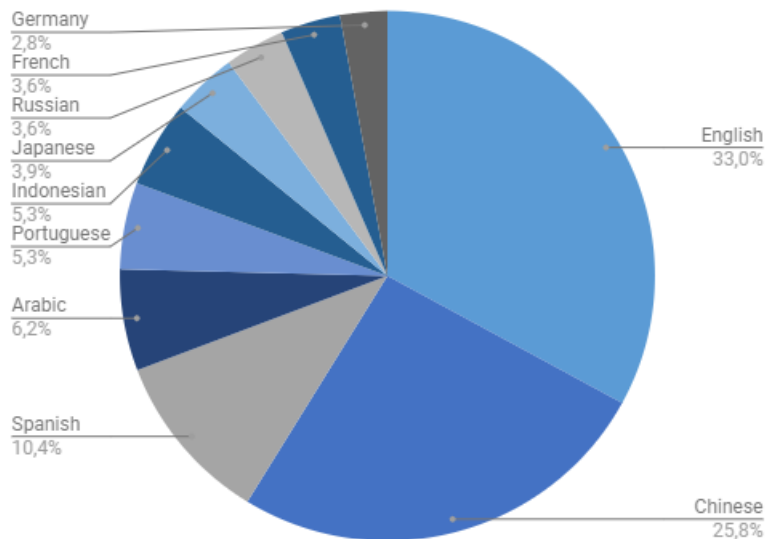


Рис. 2. Популярність мови в інтернеті

Наочно продемонструвати актуальність локалізації певного сайту можна за допомогою формули 1:

$$T = \frac{a_u - a_n}{k_w} \quad (1)$$

де a_u – це загальна кількість користувачів цього сайту; a_n – це кількість носіїв мови сайту серед користувачів; k_w – це коефіцієнт популярності мови в світі, який в свою чергу визначається за формулою 2:

$$k_w = \frac{L}{100} \quad (2)$$

де L – популярність мови у світі, яку можна побачити на рисунку 1.

Беручи до уваги формулу 1, можна зрозуміти, що якщо актуальність локалізації напряму залежить від кількості іноземних користувачів цього сайту та обернено від популярності мови у світі.

Зважаючи на це, стає зрозуміло, що локалізація – це невід’ємна частина успіху програмного продукту.

Архітектура системи

В основу розроблюваної системи покладена клієнт-серверна архітектура, яка має ряд переваг відносно відомих програмних засобів локалізації (зниження об’єму даних, що передаються, ізоляція доступу користувача до даних та логіки обробки запитів, можливість додавання клієнтів іншого типу) [4]. У свою чергу клієнт та сервер також не монолітні. Для підсилення абстракції, зменшення кількості коду та збільшення продуктивності вони побудовані на основі патернів проектування [5]. Крім того, вони поділені на шари, що підсилює інкапсуляцію, робить систему інтуїтивно зрозумілою та спрощує процес її тестування. Далі наведено запропоновану авторами структуру шарів клієнта і сервера.

Шари клієнта:

- шар представлення – відповідає за зовнішній вигляд додатка;
- проміжний шар – формує запити для сервера, та обробляє відповіді.

Шари сервера:

- проміжний шар – являє собою “ворота”, для доступу до сервера та перевіряє права користувача;
- шар логіки – містить у собі всю обробку даних та взаємодіє із віддаленим сервером для машинного перекладу;
- шар роботи з даними – ізолює стандартні операції роботи з даними (створення, редагування, читання та видалення).

Окремим модулем архітектури представлено сховище, що містить реляційну базу даних для збереження всіх даних у відношеннях, нереляційну документо-орієнтовану базу даних для збереження текстів та перекладів, а також сховище бінарних файлів та пошуковий сервер.

Окрім того особливістю запропонованої системи є використання сторонніх хмарних сервісів, а саме Firebase для аутентифікації користувача в системі та Google Translate API для здійснення машинного перекладу.

Програмна реалізація

Для реалізації серверної частини був обраний .NET Core Framework та мова С#. Особливістю даного фреймворку є кросплатформеність та оптимізація порівняно із .NET Framework. У якості ORM використовується Entity Framework Core [6], який спрощує процес взаємодії із реляційною базою даних (створення таблиць із класів моделей, підтримка міграцій) та MongoDB Driver для NoSQL бази даних. Обмін даними реалізований за допомогою надсилання HTTP запитами серіалізованих даних у форматі JSON, що набагато оптимізованіший ніж його класичний аналог XML.

Для збереження даних використовуються MS SQL (всі реляційні дані), MongoDB (документи з текстом та перекладами), Azure Blob Storage (картинки проектів та вкладені до тексту файли). Також для швидкого пошуку по документах використовується пошуковий сервер Elasticsearch, для розгортання якого був використаний Docker.

Завдяки бібліотеці SignalR досягнуто обмін даних між сервером і клієнтом у реальному часі (без оновлення сторінки).

Висновки

Усі описані засоби дозволили зробити систему сучасною, надійною, кросплатформеною, багатофункціональною, а головне – корисним інструментом для власників веб та мобільних додатків та перекладачів.

СПИСОК ВИКОРИСТАНОЇ ЛІТЕРАТУРИ

1. Google Play improves all around, still trails iOS in revenue [Електронний ресурс]. – Режим доступу: <https://www.androidauthority.com/q2-2014-app-stats-show-google-play-ios-improve-407106/>.
2. Language Localization: Pushing Mobile Ad Conversion Across the Globe [Електронний ресурс]. – Режим доступу: <http://www.adotas.com/2014/04/language-localization-pushing-mobile-ad-conversion-across-the-globe/>.

3. Survey of 3,000 Online Shoppers Across 10 Countries Finds that 60% Rarely or Never Buy from English-only Websites [Електронний ресурс]. – Режим доступу: <http://www.common senseadvisory.com/Default.aspx?Aid=21500&Contenttype=ArticleDet&tabID=64>.

4. Bass, Len. Software architecture in practice / Len Bass, Paul Clements, Rick Kazman. 3rd ed. p. cm. – (SEI series in software engineering). Includes bibliographical references and index. ISBN 978-0-321-8 1 573-6.

5. Fowler, Martin, 1963- Patterns of enterprise application architecture / Martin Fowler. p. cm. Includes bibliographical references and index. ISBN 0-321-12742-0.

6. Entity Framework Core [Електронний ресурс]. – Режим доступу: <https://docs.microsoft.com/en-us/ef/core/>.

Смольц Дмитро Олександрович – студент групи ІКІ-156, факультет інформаційних технологій та комп'ютерної інженерії, Вінницький національний технічний університет, м. Вінниця, e-mail: dima.smolts@gmail.com.

Науковий керівник: **Черняк Олександр Іванович** – канд. техн. наук, доцент, доцент кафедри обчислювальної техніки, Вінницький національний технічний університет, м. Вінниця, e-mail: chernyak@vntu.edu.ua.

Dmytro Smolts – Faculty of Information Technologies and Computer Engineering, Vinnytsia National Technical University, Vinnytsia, e-mail: dima.smolts@gmail.com.

Supervisor: **Oleksandr I. Chernyak** – Cand. Sc. (Eng.), Assistant Professor of the Computer Techniques Chair, Vinnytsia National Technical University, Vinnytsia, e-mail: chernyak@vntu.edu.ua.