**СИСТЕМА МАРКЕТИНГОВОГО ТЕСТУВАННЯ**

**Контекст:**

Уявімо ситуацію, що інтернет-магазин хоче ефективно продавати свій асортимент, використовуючи різні способи, включаючи психологічні та маркетингові трюки. Для цього у них може виникнути ідея проведення опитування серед користувачів, яке міститиме три питання:

1. Який колір кнопки «купити» Вам більше до вподоби?
2. Яку ціну на цей товар ви вважаєте найбільш привабливою?
3. Який шрифт найбільше відгукується у вашому серці?

Щоб здійснити цю задумку, інтернет-магазин повинен реалізувати додаток, який допоможе провести це опитування та візуалізувати зібрані метрики. Наш сервіс спеціально створений для таких потреб і надає всі необхідні інструменти для організації та аналізу опитувань, що допоможе інтернет-магазину оптимізувати свій продаж за допомогою точних даних та ефективних стратегій.

**Команда:**

Тімлід - Бондаренко Олександр, ІМ-22

Бекенд розробник - Павлик Гліб, ІМ-22

Фронтенд розробник - Назарій Горчинський, ІМ-22

**Цілі:**

**Розглянемо ті цілі, які ми ставимо собі перед розробкою проєкту(goals):**

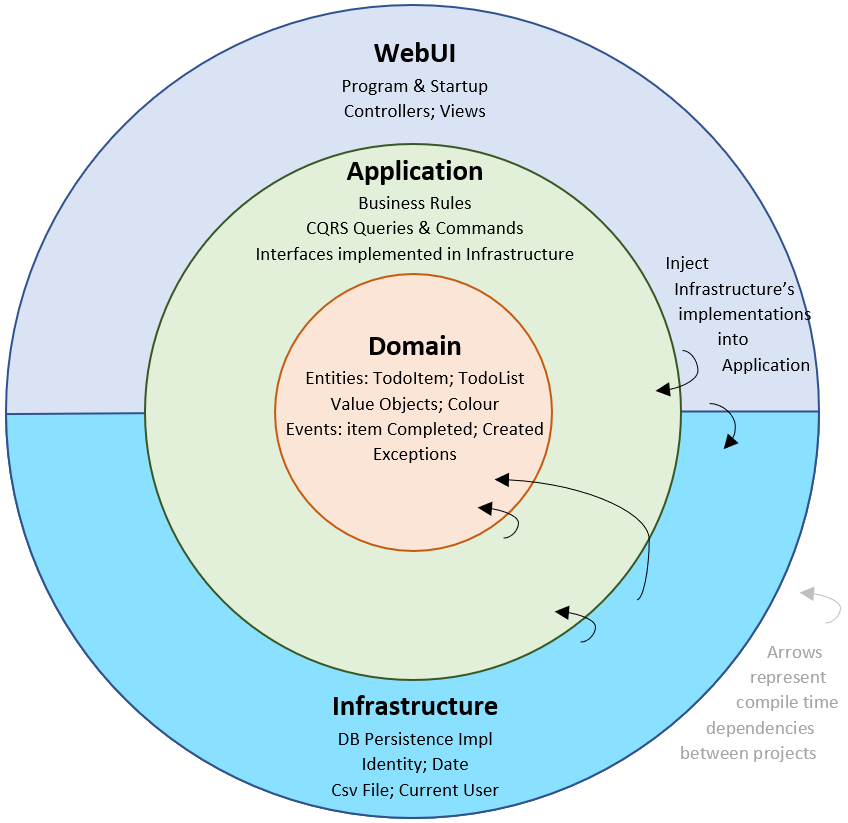
1. Провести опитування серед користувачів інтернет-магазину.
2. Зібрати дані з опитування для подальшого аналізу.
3. Візуалізувати зібрані дані для зручності аналізу та демонстрацій.
4. Аналізувати популярність певних функцій або продуктів у магазині.
5. Використовувати візуалізацію для демонстрації зацікавленості користувачів у певних деталях або функціях.

**А також те, що ми не маємо на меті реалізувувати(non-goals):**

1. Реалізація можливості постійної підтримки проєкту після завершення періоду тестування.
2. Розвиток додаткових функцій або можливостей, які виходять за рамки тестування та аналізу даних.
3. Впровадження механізмів для масштабування та розширення сервісу в майбутньому поза межами проекту.

**Архітектура**

Наша архітектура є монолітним проєктом, що ґрунтується на шаблоні DDD (Domain Driven Design) та відповідає принципам чистої архітектури (Clean Architecture). Вона складається з кореневого шару Domain, де розташовані основні сутності системи. Навколо нього розташовані шари Infrastructure, які відповідають за взаємодію з базою даних, Application, який містять бізнес-логіку, UI, відповідальний за графічний інтерфейс. Детальну структуру цієї архітектури можна побачити на діаграмі.



**Шар Domain**

Цей шар складається виключно з наших сутностей, або класів, які є кореневими для цього проєкту. Ми не прописуємо жодних зовнішніх залежностей тут. В основному, це місце, де зберігаються наші дані, які ми хочемо протестувати, і нічого більше.

**Шар Application**

У цьому пункті ми зосередимося на написанні бізнес-логіки для нашого застосунку. Для цього ми використовуватимемо шар інфраструктури, а не будемо прямо звертатися до бази даних. У цьому ж шарі ми можемо взаємодіяти з нашими сутностями напряму, однак, ми застосовуємо підхід CQRS (Command Query Responsibility Segregation) та використовуємо для цього DTO (Data Transfer Object), щоб працювати лише з тими даними, які нам потрібні у даному випадку. Вся логіка реалізується в класах, які ми називаємо сервісами.

**Шар Infrastructure**

У цьому шарі ми налаштуємо підключення до бази даних, її початкове заповнення та реалізували репозиторії. Для зручності буде використано підхід з використанням так званою InMemoryDatabase, але потрібно мати на увазі, що сам сервіс є “SQL-ready”. Тобто, якщо змінити зовсім малу кількість логіки коду, то ми вже отримаємо сервіс, який легко записуватиме все до вказаної бази даних.

**Шар UI**

У цьому шарі буде використано все, що ми розглядали вище, але у вигляді кінцевих даних. Тут не керується виконанням бізнес-логіки, а лише є кінцевим отримувачем певних даних, які відображаються на сторінках (Views). Саме з цього місця запускається застосунок.

**Система автентифікації**

Фреймворку ASP.NET має в своєму складі сервіс під назвою Identity scaffolding. Він автоматично створює всю необхідну інфраструктуру для забезпечення авторизації та аутентифікації. У цьому проєкті вона буде використовуватись з метою перевірки, що лише справжні користувачі сервісу можуть брати участь в опитуваннях.

Без авторизації клієнт сайту не матиме доступу до опитувань. Після авторизації та проходження опитування користувач не зможе повторно брати участь у ньому, але зможе переглянути загальні результати цього опитування.

**Тести**

На всі методи, що розглядаються у шарі Application, будуть написані юніт-тести, які охоплюватимуть різні аспекти цих методів. Крім того, буде налаштовано CI, що дозволить перевіряти правильність розроблення певних фрагментів застосунку і попереджати інших розробників про можливі проблеми.