

## СТЕК ТЕХНОЛОГІЙ ДЛЯ РОЗРОБКИ ПРОГРАМНОЇ СИСТЕМИ ДЛЯ ФОРМУВАННЯ СПИСКІВ БАЖАНЬ ТА КЕРУВАННЯ НИМИ

*В. О. Воропаєв<sup>1</sup>, А. А. Вальтер<sup>2</sup>, М. С. Широкопетлева<sup>3</sup>*

<sup>1</sup> студент кафедри програмної інженерії, ХНУРЕ, Харків, [vladyslav.voropaiev1@nure.ua](mailto:vladyslav.voropaiev1@nure.ua)

<sup>2</sup> студент кафедри програмної інженерії, ХНУРЕ, Харків, [anna.valter@nure.ua](mailto:anna.valter@nure.ua)

<sup>3</sup> старший викладач кафедри програмної інженерії, ХНУРЕ, Харків

Процес координації вибору подарунків у колективному середовищі часто страждає від неефективності, наприклад, дублювання та поганої координації, через обмежену функціональність існуючих систем. У цій роботі представлено програмну архітектуру, призначену для вирішення цих проблем, з акцентом на гнучкому і масштабованому рішенні для управління як індивідуальними, так і груповими сценаріями дарування подарунків. Система, заснована на об'єктно-орієнтованій підході та монолітній архітектурі, пропонує розширені можливості для створення списків бажань, координації спільних подарунків і генерації автоматичних рекомендацій. Завдяки використанню сучасних технологій, таких як .NET 8.0, Angular 18 і PostgreSQL, запропонована система значно покращує користувацький досвід і сприяє безперешкодному плануванню та координації подарунків.

Існують онлайн-сервіси для створення списків бажань і керування ними, однак, багато з цих систем, не вирішують складнощів, пов'язаних з координацією подарунків у групах. Сьогодні існує низка платформ для управління списками подарунків, але більшість з них розроблені для індивідуальних користувачів і надають обмежену підтримку для колективного дарування. Rewish [1], наприклад, дозволяє користувачам створювати списки бажань, але не пропонує надійних функцій для групової координації. Аналогічно, Wishpicks [2] надає базові можливості для створення списків бажань, але не має механізмів для уникнення дублікатів подарунків та організації спільних зусиль між кількома користувачами. Існуючих систем недостатньо для управління спільними зусиллями, що призводить до неефективності, наприклад, дублювання подарунків і поганого користувацького досвіду. Ці системи також страждають від обмеженої масштабованості та персоналізації, які є важливими у сценаріях, де кілька користувачів повинні співпрацювати над вибором подарунків. Коли сім'ї, друзі чи колеги планують подарунки разом, цим інструментам бракує необхідних функцій для уникнення дублювання подарунків та забезпечення ефективної комунікації між користувачами.

Метою представленої роботи є проектування програмної системи, яка дозволить знайти рішення поставленої проблеми і надасть можливості індивідуального та групового управління подарунками. Однією із задач є розробка архітектурного рішення, яке забезпечить необхідну гнучкість, масштабованість і простоту використання для задоволення різноманітних потреб групових сценаріїв дарування подарунків, а також спростить процес для індивідуальних користувачів.

Для реалізації системи запропонований наступний стек технологій для проектування системи з використанням DDD-підходу [3].

Платформа .NET 8.0 була обрана через її високу продуктивність, ефективну роботу з асинхронними процесами та широку екосистему. Важливим фактором є інтеграція з іншими компонентами, такими як SignalR, що забезпечує обмін даними в реальному часі через вебсокети, що критично для колективної координації подарунків. SignalR

дозволяє забезпечити низьку латентність і надійність передачі даних між клієнтом і сервером.

MediatR використовується для побудови архітектури з мінімальним зв'язком між компонентами, що підвищує гнучкість системи та полегшує її модульність і підтримку. Такий підхід відповідає принципам CQRS, що забезпечує чітке розділення запитів і команд в архітектурі.

В якості ORM обрано Entity Framework Core завдяки можливості спростити взаємодію застосунку з базою даних, реалізованою на PostgreSQL. Це дозволяє абстрагуватися від низькорівневої взаємодії з SQL і зосередитися на логіці бізнес-правил, що підвищує продуктивність розробки.

СУБД PostgreSQL обрана за її надійність, потужні можливості для зберігання великих обсягів даних і підтримку транзакційності. Вона добре підходить для обробки складних запитів, які можуть виникати під час координації подарунків у великих групах. PostgreSQL також надає розширені можливості для масштабування бази даних у випадках збільшення навантаження та керування привілеями користувачів.

AutoMapper автоматизує процес мапінгу між бізнес-об'єктами, що дозволяє мінімізувати помилки і підвищує швидкість розробки. FluentValidation додає можливості для чіткої декларації валідаційних правил, що робить код легким для підтримки й тестування.

Angular 18 з TypeScript було обрано через його можливості для побудови масштабованих односторінкових додатків, де модульність коду та інтеграція з сучасними бібліотеками грають ключову роль. Використання таких інструментів, як Angular Material і SASS, забезпечує високий рівень кастомізації інтерфейсу користувача, що покращує користувацький досвід.

Для підвищення продуктивності застосунку застосовується lazy loading, що забезпечує завантаження тільки необхідних компонентів на вимогу. Це не лише оптимізує використання ресурсів, але й дозволяє швидше реагувати на дії користувача в системі, підвищуючи її ефективність.

Важливим аспектом системи є забезпечення безпеки даних користувачів. Для цього обрано Firebase Authentication, який надає рішення для аутентифікації та зберігання профілів користувачів. Ця технологія дозволяє забезпечити швидке впровадження безпеки, а також підтримку різних методів авторизації (електронна пошта, соціальні мережі, тощо).

Для розгортання серверної частини використовується Docker, що дозволяє легко створювати контейнери з ізольованим середовищем для сервісів. Це забезпечує портативність і спрощує процеси CI/CD. Віртуальне середовище Render дозволяє ефективно управляти контейнеризованими застосунками, а хостинг клієнтської частини на Vercel оптимізує продуктивність і автоматизує розгортання.

В результаті було створено програмну систему, яка ефективно вирішує проблему координації та організації вибору подарунків як у персональних, так і в колективних сценаріях. Система надає зручний інтерфейс для створення та управління вішалістами, дозволяє спільно обирати та дарувати подарунки, а також генерує рекомендації, що підвищує точність вибору подарунків. Це значно покращує користувацький досвід і спрощує процес дарування, особливо в ситуаціях, де участь у виборі подарунків беруть декілька людей.

#### **Список літератури:**

1. Plan Your Dreams to make them True | REWISH [Електрон. ресурс]. – Режим доступу: <https://rewish.io/home> Plan Your Dreams to make them True | REWISH
2. Wishpicks: Платформа для керування вішалістами | Wishpicks [Електрон. ресурс]. – Режим доступу: <https://wishpicks.com/uk/wishlists> Wishpicks: Платформа для керування вішалістами | Wishpicks
3. Evans E. *Domain-Driven Design: Tackling Complexity in the Heart of Software* / Pearson Education, Limited. – 2021. – №1 – С. 8.