**Vision and Scope Document**

**for**

**To Do App software system for task management**

**Version 1.0**

**approved**

**Prepared by**

**Korobeinyk Volodymyr**

**NURE**

## 03.06.2025

## 1. Постановка мети.

Ціллю роботи є розробка тестових веб-застосунків “To Do App”, що використовуються для дослідження продуктивності та ефективності різних state-менеджерів у середовищі Angular. Застосунки імітують типовий список задач із базовим CRUD-функціоналом, дозволяючи провести об’єктивне порівняння state-менеджерів NgRx, Akita та Ngx-base-state у рівних умовах.

Розроблені застосунки дозволяють моделювати типові сценарії взаємодії з глобальним та локальним станом, здійснювати моніторинг продуктивності, аналіз використання пам’яті, а також складності інтеграції в архітектуру Angular-додатків. Кінцевою метою є отримання експериментальних даних для формулювання рекомендацій щодо вибору оптимального state-менеджера залежно від специфіки проєкту.

Для реалізації застосунків використано сучасні технології:

* клієнтська частина. Для створення інтерфейсу веб-застосунку застосовано JavaScript-фреймворк Angular (v17+) у поєднанні з мовою TypeScript. Для стилізації інтерфейсу використано Tailwind CSS. У кожному з трьох варіантів реалізації стан керується за допомогою одного з обраних state-менеджерів – NgRx, Akita або Ngx-base-state. Для реактивного програмування використовується бібліотека RxJS;
* серверна частина: Серверна логіка реалізована за допомогою Node.js з використанням фреймворку NestJS. Сервер виконує роль API-посередника між клієнтом і базою даних, забезпечує обробку запитів, реалізацію CRUD-операцій, автентифікацію користувачів і валідацію даних;
* база даних. Для зберігання інформації про задачі та групи задач використано реляційну базу даних PostgreSQL. Для ORM-зв’язку між моделями та БД застосовано бібліотеку Sequelize.

Для реалізації програмної системи використано трирівневу архітектуру, яка передбачає чітке розділення на:

* клієнтський рівень (інтерфейс користувача, логіка відображення, state-менеджмент);
* серверний рівень (логіка обробки запитів, доступ до даних, реалізація REST API);
* рівень бази даних (зберігання структурованих сутностей “Task” та “Group”).

Архітектурно всі три клієнтські реалізації об’єднані в один monorepo-проєкт з використанням інструмента Nx, що дозволяє повторно використовувати компоненти, сервіси, утиліти та створити спільну бібліотеку shared для спрощення тестування state-менеджерів в однакових умовах.

## 2. Загальний опис системи.

Програмна система являє собою клієнт-серверний веб-застосунок, створений для дослідження продуктивності та ефективності різних state-менеджерів у середовищі Angular. Застосунок реалізує базову функціональність типового списку задач (To Do App) із підтримкою операцій створення, редагування, видалення та перегляду задач, а також зберігання їхнього статусу у локальному або глобальному стані.

Розроблена система використовується як тестовий полігон для порівняння трьох популярних state-менеджерів: NgRx, Akita та Ngx-base-state. Всі реалізації побудовані за єдиною структурною моделлю, що забезпечує рівні умови для порівняння показників продуктивності, споживання ресурсів та зручності інтеграції.

Програмна система вирішує завдання об’єктивного вибору state-менеджера в реальних умовах розробки Angular-додатків. Вона може бути використана як приклад у навчальних цілях, у внутрішніх дослідженнях, а також для побудови рекомендаційних систем вибору інструментів під конкретні задачі проєкту.

Клієнтська частина дозволяє користувачу взаємодіяти з інтерфейсом додатку для керування списком задач. У кожній версії реалізовано незалежне управління станом, що відповідає обраному state-менеджеру. Додаток має просту адаптивну структуру інтерфейсу, створену з використанням Angular та Tailwind CSS.

Серверна частина обробляє запити до бази даних, реалізує логіку CRUD-операцій для задач і груп задач, а також забезпечує базові функції авторизації та валідації. Вона побудована з використанням NestJS – фреймворку для побудови масштабованих Node.js-систем.

Сервер також реалізує базові механізми безпеки, включаючи, валідацію запитів, обмеження доступу до ресурсів. У перспективі можлива інтеграція з зовнішніми сервісами або розширення функціоналу для симуляції складніших сценаріїв.

## 3. Основний функціонал системи

У системі існує лише один тип користувача — User. Усі функції доступні для цього типу користувача без обмежень, оскільки тестовий додаток не передбачає поділу за ролями та аккаунтами. Основна увага приділяється реалізації базового функціоналу для створення, редагування, перегляду, видалення та пошуку задач і груп задач, а також зміні статусу задач (виконано / не виконано) (див. табл. 3.1).

Таблиця 3.1 – Функціонал програмного застосунку “To Do App”

|  |  |  |
| --- | --- | --- |
| № | Функція | Опис |
| 1 | Створення групи задач | Дозволяє створити нову групу для логічного об’єднання задач. Користувач вводить назву групи. |
| 2 | Редагування групи задач | Дозволяє змінити назву існуючої групи задач. |
| 3 | Видалення групи задач | Дозволяє видалити групу задач. При цьому можуть бути також видалені задачі, що до неї входять. |
| 4 | Пошук груп задач | Дозволяє шукати групи задач за ключовим словом у назві. |

Продовження таблиці 3.1

|  |  |  |
| --- | --- | --- |
| 5 | Створення задачі | Дозволяє створити нову задачу в межах певної групи. Задається назва задачі, опис (необов’язково). |
| 6 | Редагування задачі | Дозволяє редагувати інформацію про задачу: змінити назву, опис або групу належності. |
| 7 | Видалення задачі | Дозволяє повністю видалити задачу зі списку. |
| 8 | Зміна статусу задачі | Дозволяє встановити або змінити статус задачі як «виконано» або «не виконано» |
| 9 | Фільтрація задач за статусом | Дозволяє відфільтрувати задачі за статусом виконання (усі, виконані, невиконані). |
| 10 | |  |  | | --- | --- | |  | Пошук задач у межах групи | | Дозволяє знайти задачі в рамках конкретної групи за ключовим словом |

## 4. Загальні обмеження

Тестовий застосунок To Do App, призначений для дослідження state-менеджерів у середовищі Angular, має наступні обмеження:

* застосунок не може бути використаний без доступу до інтернету, оскільки взаємодіє з серверною частиною через API;
* система не передбачає багатокористувацький режим або розмежування прав доступу — є лише роль користувача;
* користувач не може створити задачу без вибраної або створеної групи задач;
* відсутня функція сортування задач за пріоритетністю або дедлайном — реалізовано лише базову фільтрацію за статусом;
* користувач не може створити дві групи задач із однаковою назвою (в межах однієї сесії без оновлення);
* застосунок не підтримує офлайн-режим або локальне збереження задач у браузері;
* додаток не оптимізовано для мобільних пристроїв — інтерфейс орієнтовано на роботу в браузері десктопного типу.

## 5. Припущення та залежності

Тестовий застосунок To Do App, що призначений для дослідження ефективності state-менеджерів у Angular, має наступні припущення та залежності:

* для створення задачі користувач повинен попередньо створити або вибрати групу задач;
* користувач має стабільне інтернет-з’єднання, оскільки застосунок працює у клієнт-серверній архітектурі;
* користувач буде використовувати застосунок у середовищі сучасного браузера (Chrome, Firefox, Edge), що підтримує стандарт ES6+;
* користувач використовує десктопну версію браузера, оскільки мобільна адаптація не передбачена;
* для тестування кожного state-менеджера застосунок запускається у окремому середовищі або як окремий підпроєкт в monorepo-структурі;
* серверна частина працює на Node.js з NestJS, а база даних PostgreSQL розгорнута локально або на тестовому хості;
* порівняльний аналіз state-менеджерів виконується за однакових умов і на однаковому наборі задач і груп.