

# “Just Do It” Web Application Software Requirement Specification

---

Version 1

18.09.2023

Olexandr Bykreev

Lead Software Engineer

## Revision History

Date	Description	Author	Comments
18.09.2023	Version 1	Olexandr Bykreev	First Revision
.			
.			
.			

## Document Approval

Наступна специфікація вимог до програмного забезпечення була прийнята та затверджена наступними особами:

Signature	Printed Name	Title	Date
	O. Bykreev	Lead Software	01.10.2023
	A. Chupryna	Product Owner	
	M. Hryenko	Developer	
	V. Mykytenko	QA Engineer	
	S. Lutai	Scrum master	

## 1. Introduction

Наступний документ визначає докладну специфікацію для веб-застосунку "Just Do It". Основна мета цього документу - надати технічний зразок, який можна використовувати програмними інженерами для створення та проектування програмного забезпечення.

### 1.1 Purpose

Мета цього SRS полягає у деталізації специфікацій та вимог до веб-застосунку "Just Do It", який призначений для осіб, які шукають ефективний, зручний у використанні інструмент управління

завданнями. Цей документ призначений для розробників, керівників проектів та тестувальників, які будуть працювати над розробкою, впровадженням та верифікацією веб-застосунку "Just Do It".

## 1.2 Scope

- **Software Product:** Програмний продукт, який потрібно розробити - це веб-застосунок під назвою "Just Do It".
- **Functionality:** Додаток надає користувачам платформу для управління їх завданнями, з різноманітними функціями, доступними на основі їх статусу реєстрації. Зареєстровані користувачі мають додаткові можливості, такі як створення кількох списків завдань та зберігання даних у хмарній базі даних.
- **Application:** Застосунок надасть ефективне рішення для управління завданнями з простим та інтуїтивно зрозумілим користувацьким інтерфейсом, на основі добре встановленої моделі, як TodoMVC.

## 1.3 Definitions, Acronyms, and Abbreviations

- **CRUD:** Створення, Читання, Оновлення, Видалення (Create, Read, Update, Delete)
- **SRS:** Специфікація Вимог до Програмного Забезпечення (Software Requirement Specification)
- **DB:** База Даних (Database)

## 1.4 References

- TodoMVC - <https://todomvc.com/>

## 1.5 Overview

Цей документ поділений на різні розділи, які деталізують загальні та специфічні вимоги веб-застосунку "Just Do It". Документ продовжується загальним описом продукту, специфічними вимогами, включаючи зовнішні вимоги до інтерфейсу, функціональні вимоги та нефункціональні вимоги, аналіз моделей та процес управління змінами.

# 2. General Description

## 2.1 Product Perspective

Веб-застосунок "Just Do It" є незалежною платформою, натхненною існуючими моделями управління завданнями, такими як TodoMVC.

## 2.2 Product Functions

Основна функція застосунку - управління завданнями, де користувачі можуть додавати, видаляти, редагувати та перемикати завдання. Зареєстровані користувачі мають додаткові функціональні можливості, такі як управління кількома списками завдань та доступ до панелі керування CRUD.

## 2.3 User Characteristics

Від користувачів системи очікується наявність базових навичок роботи з комп'ютером та Інтернетом. Вони представляють різні демографічні групи і шукають цифровий інструмент для ефективного управління своїми завданнями.

## 2.4 General Constraints

Веб-застосунок повинен бути розроблений так, щоб бути реактивним на різних пристроях та платформах. Він повинен бути агностичним до браузера та дотримуватися веб-стандартів.

## 2.5 Assumptions and Dependencies

- Припускається, що база даних PostgreSQL буде використовуватися для зберігання даних зареєстрованих користувачів.
- Застосунок буде розроблено з використанням фреймворку Next.js.

## 3. Specific Requirements

### 3.1 External Interface Requirements

#### 3.1.1 User Interfaces

Користувацький інтерфейс повинен бути інтуїтивним та мінімалістичним, забезпечуючи легке навігацію та доступ до всіх функцій.

#### 3.1.2 Hardware Interfaces

Не застосовується.

#### 3.1.3 Software Interfaces

- Фронтенд: Next.js
- Бекенд: База даних PostgreSQL

#### 3.1.4 Communications Interfaces

Не застосовується.

### 3.2 Functional Requirements

Основні функціональні можливості застосунку описані з урахуванням різних ролей користувачів:

#### 3.2.1 Anonymous User Functions

##### 3.2.1.1 Introduction

Користувачі без реєстрації можуть використовувати базовий список завдань.

##### 3.2.1.2 Inputs

- Опис завдання
- Статус завдання (виконано/не виконано)

##### 3.2.1.3 Processing

- Додавання, видалення, перейменування, перемикання завдань.
- Фільтрація завдань за статусом.
- Перемикання всіх завдань.
- Очищення виконаних завдань.

#### 3.2.1.4 Outputs

- Оновлений список завдань.

#### 3.2.1.5 Error Handling

- Неприпустимі символи у вводі.
- Надмірна довжина вводу.

### 3.2.2 User Authentication and Authorization

#### 3.2.2.1 Introduction

Цей вимога включає реєстрацію, вхід та вихід користувачів.

#### 3.2.2.2 Inputs

- Реєстрація: `email`, `name`, `password`.
- Вхід: `email`, `password`.
- Вихід: Немає.

#### 3.2.2.3 Processing

- Реєстрація: Перевірка формату електронної пошти, міцності пароля та унікальності електронної пошти перед створенням нового об'єкта користувача.
- Вхід: Перевірка відповідності електронної пошти та пароля існуючому об'єкту користувача, та авторизація користувача.
- Вихід: Деавторизація користувача.

#### 3.2.2.4 Outputs

- Реєстрація: Підтвердження успішної реєстрації або повідомлення про помилку.
- Вхід: Токен авторизації або повідомлення про помилку.
- Вихід: Підтвердження виходу.

#### 3.2.2.5 Error Handling

- Реєстрація: Надання повідомлення про помилку для неприпустимого формату електронної пошти, слабкого пароля або неунікальної електронної пошти.
- Вхід: Надання повідомлення про помилку для некоректної електронної пошти або пароля.
- Вихід: Обробка будь-якої помилки, яка може виникнути під час процесу виходу.

### 3.2.3 To-do List Management

### 3.2.3.1 Introduction

Цей вимога включає створення, видалення та перейменування списків завдань.

### 3.2.3.2 Inputs

- Створення: `title`.
- Видалення: `listId`.
- Перейменування: `listId, newTitle`.

### 3.2.3.3 Processing

- Створення: Створення нового об'єкта списку з вказаним заголовком.
- Видалення: Видалення вказаного об'єкта списку.
- Перейменування: Оновлення заголовку вказаного об'єкта списку.

### 3.2.3.4 Outputs

- Створення: Новий створений об'єкт списку або повідомлення про помилку.
- Видалення: Підтвердження видалення або повідомлення про помилку.
- Перейменування: Оновлений об'єкт списку або повідомлення про помилку.

### 3.2.3.5 Error Handling

- Створення: Обробка помилок, таких як неприпустимий заголовок.
- Видалення: Обробка помилок, таких як неприпустимий `listId`.
- Перейменування: Обробка помилок, таких як неприпустимий `listId` або неприпустимий новий заголовок.

## 3.2.4 Task Management

### 3.2.4.1 Introduction

Цей вимога включає створення, видалення, перейменування та перемикання статусу завдань.

### 3.2.4.2 Inputs

- Створення: `title, priority`.
- Видалення: `taskId`.
- Перейменування: `taskId, newTitle`.
- Перемикання статусу: `taskId`.

### 3.2.4.3 Processing

- Створення: Створення нового об'єкта завдання з вказаним заголовком та пріоритетом.
- Видалення: Видалення вказаного об'єкта завдання.
- Перейменування: Оновлення заголовку вказаного об'єкта завдання.
- Перемикання статусу: Перемикання статусу завершення вказаного об'єкта завдання.

#### 3.2.4.4 Outputs

- Створення: Новий створений об'єкт завдання або повідомлення про помилку.
- Видалення: Підтвердження видалення або повідомлення про помилку.
- Перейменування: Оновлений об'єкт завдання або повідомлення про помилку.
- Перемикання статусу: Оновлений об'єкт завдання або повідомлення про помилку.

#### 3.2.4.5 Error Handling

- Створення: Обробка помилок, таких як неприпустимий заголовок або пріоритет.
- Видалення: Обробка помилок, таких як неприпустимий `taskId`.
- Перейменування: Обробка помилок, таких як неприпустимий `taskId` або неприпустимий новий заголовок.
- Перемикання статусу: Обробка помилок, таких як неприпустимий `taskId`.

### 3.3 Use Cases

#### 3.3.1 Use Case: Adding a New Task

- **Actor:** Анонімний/Зареєстрований користувач
- **Preconditions:** Користувач перебуває на головній сторінці додатку.
- **Basic Flow:**
  1. Користувач вводить опис завдання.
  2. (Для зареєстрованих користувачів) Користувач вибирає пріоритет для завдання.
  3. Користувач натискає кнопку "Додати".
  4. Завдання додається до списку.
- **Exceptions:**
  1. Недійсні символи вводу або надмірна довжина вводу викликають повідомлення про помилку.

#### 3.3.2 Use Case: Deleting a Task

- **Actor:** Анонімний/Зареєстрований користувач
- **Preconditions:** У користувача є принаймні одне завдання у своєму списку.
- **Basic Flow:**
  1. Користувач натискає кнопку "Видалити" поруч із завданням.
  2. Завдання видалюється зі списку.
- **Exceptions:**
  1. Якщо `taskId` є недійсним, відображається повідомлення про помилку.

#### 3.3.3 Use Case: Renaming a Task

- **Actor:** Анонімний/Зареєстрований користувач
- **Preconditions:** У користувача є принаймні одне завдання у своєму списку.
- **Basic Flow:**
  1. Користувач натискає на опис завдання.
  2. Користувач змінює опис завдання.
  3. Користувач натискає за межі текстового поля або натискає "Enter", щоб зберегти зміну.

- **Exceptions:**

1. Недійсні символи вводу, надмірна довжина вводу або недійсний `taskId` викликають повідомлення про помилку.

### 3.3.4 Use Case: Toggling a Task's Completion Status

- **Actor:** Анонімний/Зареєстрований користувач
- **Preconditions:** У користувача є принаймні одне завдання у своєму списку.
- **Basic Flow:**
  1. Користувач натискає на прапорець поруч із завданням.
  2. Статус виконання завдання перемикається.
- **Exceptions:**
  1. Якщо `taskId` є недійсним, відображається повідомлення про помилку.

### 3.3.5 Use Case: Filtering Tasks by Status

- **Actor:** Анонімний/Зареєстрований користувач
- **Preconditions:** У користувача є принаймні одне завдання у своєму списку.
- **Basic Flow:**
  1. Користувач вибирає фільтр статусу (наприклад, Всі, Активні, Виконані).
  2. Список завдань оновлюється, щоб показати лише завдання, які відповідають вибраному статусу.

### 3.3.6 Use Case: Toggling All Tasks

- **Actor:** Анонімний користувач
- **Preconditions:** У користувача є принаймні одне завдання у своєму списку.
- **Basic Flow:**
  1. Користувач натискає кнопку "Перемкнути все".
  2. Усі завдання відзначаються як виконані або активні залежно від їх поточного статусу.

### 3.3.7 Use Case: Clearing Completed Tasks

- **Actor:** Анонімний користувач
- **Preconditions:** У користувача є принаймні одне виконане завдання у своєму списку.
- **Basic Flow:**
  1. Користувач натискає кнопку "Очистити виконані".
  2. Усі виконані завдання видаляються зі списку.

### 3.3.8 Use Case: Registering a New User

- **Actor:** Анонімний користувач
- **Preconditions:** Користувач перебуває на сторінці реєстрації додатку.
- **Basic Flow:**
  1. Користувач вводить свій email, ім'я та пароль у відповідні поля.
  2. Користувач натискає кнопку "Реєстрація".
  3. Система перевіряє формат email, міцність пароля та унікальність email.

4. Створюється новий об'єкт Користувача, і користувач отримує підтвердження про успішну реєстрацію.

- **Exceptions:**

1. Недійсний формат email, слабкий пароль або неунікальний email викликають повідомлення про помилку.

### 3.3.9 Use Case: Logging In

- **Actor:** Анонімний користувач

- **Preconditions:** Користувач перебуває на сторінці входу в додаток.

- **Basic Flow:**

1. Користувач вводить свій email та пароль у відповідні поля.
2. Користувач натискає кнопку "Вхід".
3. Система перевіряє відповідність email та пароля існуючому об'єкту Користувача.
4. Користувач отримує токен авторизації та переадресовується на головну сторінку.

- **Exceptions:**

1. Неправильний email або пароль викликають повідомлення про помилку.

### 3.3.10 Use Case: Logging Out

- **Actor:** Зареєстрований користувач

- **Preconditions:** Користувач увійшов у додаток.

- **Basic Flow:**

1. Користувач натискає кнопку "Вихід".
2. Користувач виходить з системи та переадресовується на сторінку входу.

- **Exceptions:**

1. Будь-яка помилка, яка виникає під час процесу виходу, викликає повідомлення про помилку.

### 3.3.11 Use Case: Creating a New List

- **Actor:** Зареєстрований користувач

- **Preconditions:** Користувач увійшов у додаток.

- **Basic Flow:**

1. Користувач переходить до розділу для створення нового списку.
2. Користувач вводить назву для нового списку.
3. Користувач натискає кнопку "Створити".
4. Створюється новий об'єкт Список, і користувач направляється на новостворений список.

- **Exceptions:**

1. Недійсна назва викликає повідомлення про помилку.

### 3.3.12 Use Case: Deleting a List

- **Actor:** Зареєстрований користувач

- **Preconditions:** У користувача є принаймні один створений список.

- **Basic Flow:**

1. Користувач переходить до списку, який він хоче видалити.
2. Користувач натискає кнопку "Видалити".



3. Вказаний об'єкт Список видаляється, і користувач переадресовується на головну сторінку.

- **Exceptions:**

1. Недійсний `listId` викликає повідомлення про помилку.

### 3.3.13 Use Case: Renaming a List

- **Actor:** Зареєстрований користувач

- **Preconditions:** У користувача є принаймні один створений список.

- **Basic Flow:**

1. Користувач переходить до списку, який він хоче перейменувати.

2. Користувач натискає на назву списку.

3. Користувач вводить нову назву і натискає "Enter" або натискає за межі текстового поля, щоб зберегти зміну.

- **Exceptions:**

1. Недійсна назва або недійсний `listId` викликають повідомлення про помилку.

## 3.4 Classes / Objects

### 3.4.1 User (Користувач)

#### 3.4.1.1 Attributes

- `id`: Number (Число), унікальний ідентифікатор для кожного користувача.
- `email`: String (Рядок), електронна адреса користувача для входу та комунікації.
- `name`: String (Рядок), ім'я користувача для персоналізації.
- `password`: Encrypted string (Зашифрований рядок), використовується для аутентифікації.
- `lists`: Array/List of List objects (Масив/Список об'єктів List), які представляють власні списки завдань користувача.

#### 3.4.1.2 Functions

- `register(email, name, password)`: Реєструє нового користувача.
- `login(email, password)`: Аутентифікує користувача.
- `logout()`: Виходить з системи.
- `createList(title)`: Створює новий список завдань.
- `deleteList(listId)`: Видаляє вказаний список завдань.

### 3.4.2 List (Список)

#### 3.4.2.1 Attributes

- `id`: Number (Число), унікальний ідентифікатор для кожного списку завдань.
- `title`: String (Рядок), назва списку завдань.
- `ownerId`: Number (Число), ідентифікатор користувача, який володіє списком.
- `tasks`: Array/List of Task objects (Масив/Список об'єктів Task), які представляють завдання в межах списку завдань.

#### 3.4.2.2 Functions

- `createTask(title, priority)`: Створює нове завдання у списку.
- `deleteTask(taskId)`: Видаляє вказане завдання зі списку.
- `renameList(newTitle)`: Перейменовує список завдань.

#### 3.4.3 Task (Завдання)

##### 3.4.3.1 Attributes

- `id`: Number (Число), унікальний ідентифікатор для кожного завдання.
- `title`: String (Рядок), назва завдання.
- `completed`: Boolean (Булевий), вказує, чи завершено завдання чи ні.
- `listId`: Number (Число), ідентифікатор списку, до якого належить завдання.
- `priority`: Enum (Low, Medium, High) (Перелік (Низький, Середній, Високий)), рівень пріоритету завдання.

##### 3.4.3.2 Functions

- `toggleStatus()`: Перемикає статус завдання між завершеним та не завершеним.
- `renameTask(newTitle)`: Перейменовує завдання.
- `changePriority(newPriority)`: Змінює пріоритет завдання.

### 3.5 Non-Functional Requirements

#### 3.5.1 Performance

- Додаток має завантажуватися швидко, за 3 секунди або менше, на звичайному широкосмуговому з'єднанні.
- Більшість дій у додатку (95%) повинні відбуватися миттєво, за менше ніж 1 секунду.

#### 3.5.2 Reliability

- Система повинна бути дуже надійною, з максимальним часом простою лише 1 хвилина на день.

#### 3.5.3 Scalability

- Система повинна працювати добре, навіть коли 1,000 користувачів використовують її водночас.
- База даних повинна швидко зростати, якщо приєднуються нові користувачі, з темпом зростання 10% кожного місяця.

#### 3.5.4 Security

- Усі дані користувачів повинні бути захищені за допомогою шифрування, незалежно від того, передаються вони чи зберігаються.
- Паролі повинні зберігатися безпечно за допомогою сильної криптографії.
- Перевірки безпеки повинні проводитися двічі на рік для виявлення та усунення вразливостей.

### 3.5.5 Usability

- Додаток повинен виглядати добре та працювати добре як на телефонах, так і на комп'ютерах.
- Дизайн повинен бути зручним для користувача, що робить додаток легким для навігації та використання.

### 3.5.6 Backup and Recovery

- Повне резервне копіювання даних користувачів повинно відбуватися щодня.
- Резервні копії повинні зберігатися в іншому місці для захисту від регіональних катастроф.

## 3.6 Inverse Requirements

Не вказано.

## 3.7 Design Constraints

- Додаток повинен бути реалізований з використанням Next.js для фронтенду.
- Для зберігання даних на бекенді повинна використовуватися PostgreSQL.

## 3.8 Logical Database Requirements

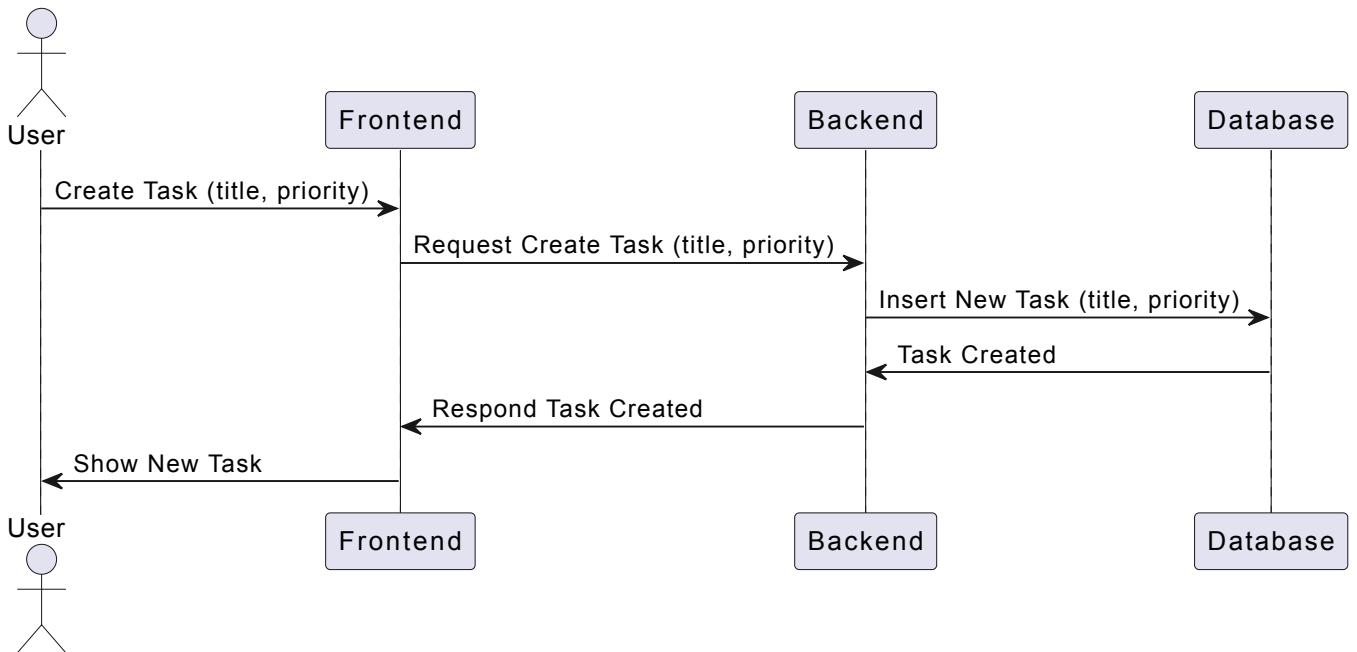
- PostgreSQL буде використовуватися для зберігання даних зареєстрованих користувачів та їхніх списків справ.
- Цілісність даних, збереження та можливості зберігання повинні відповідати найкращим практикам управління базами даних.

## 3.9 Other Requirements

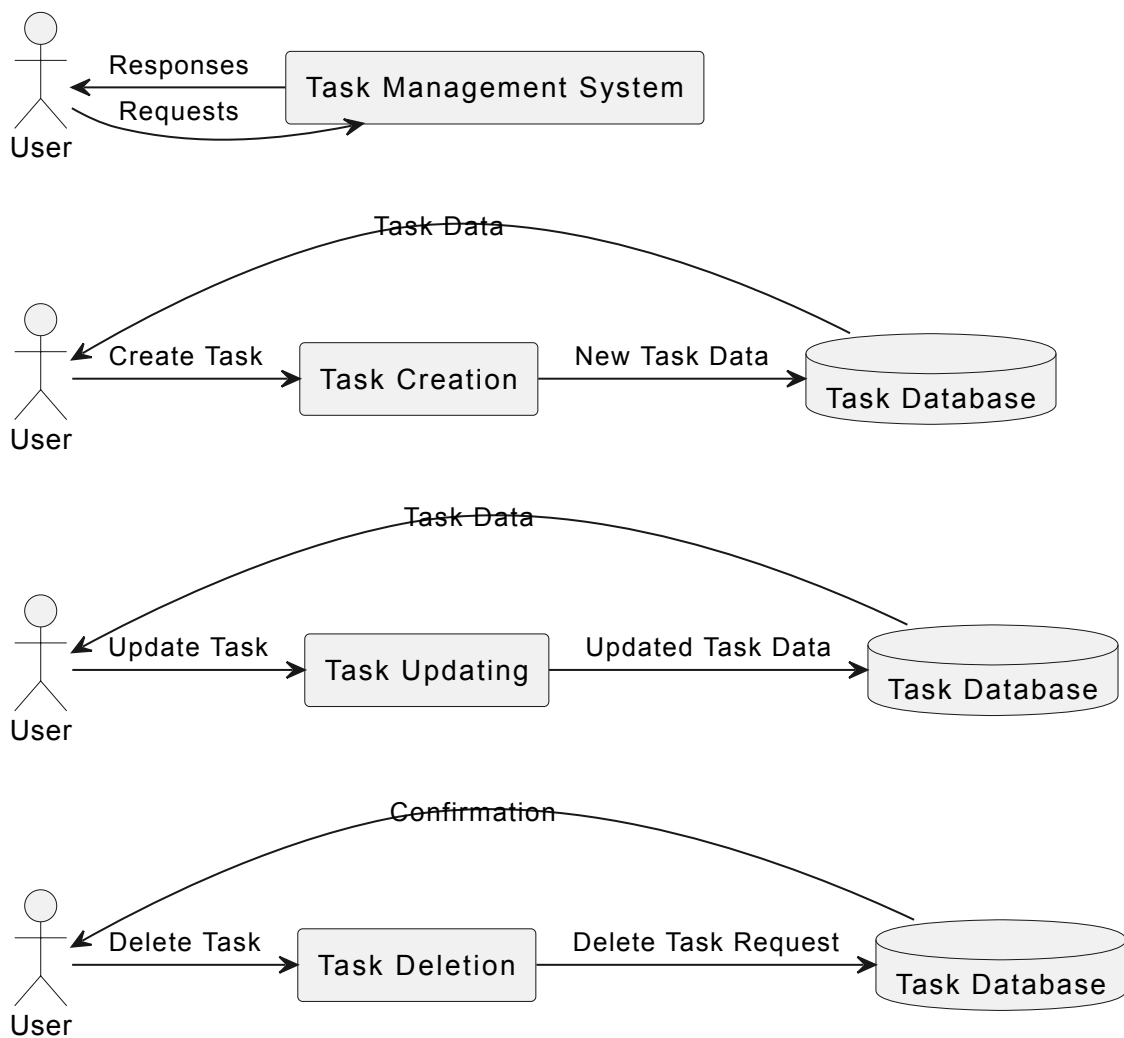
Не вказано.

# 4. Analysis Models

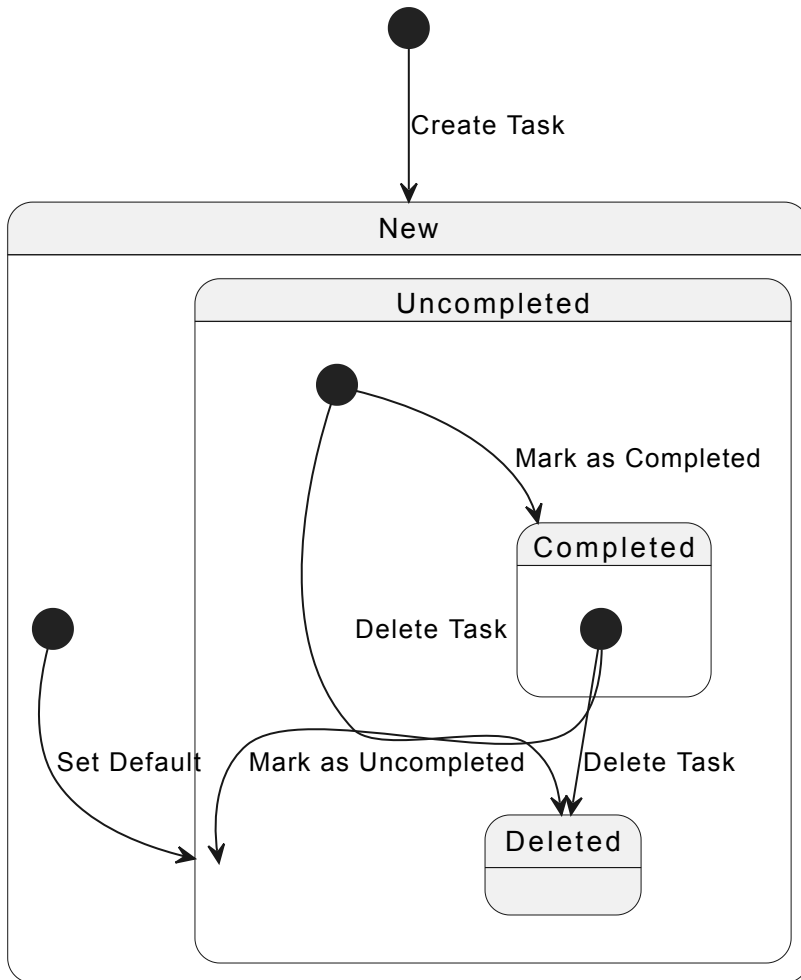
## 4.1 Sequence Diagrams



## 4.2 Data Flow Diagrams (DFD)



## 4.3 State-Transition Diagrams (STD)



## 5. Change Management Process

- Всі запити на зміни мають бути подані через певний інструмент управління проектом (Trello).
- Запити на зміни мають бути розглянуті та схвалені керівником проекту перед їх впровадженням.
- Документ SRS буде відповідно оновлено, щоб відображати схвалені зміни.