

Just do it! App

Software Requirements 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

The following Software Requirements Specification has been accepted and approved by the following:

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

Table of Contents

Revision History.....	ii
Document Approval.....	ii
1. Introduction.....	1
1.1 Purpose.....	1
1.2 Scope.....	1
1.3 Definitions, Acronyms, and Abbreviations.....	1
1.4 References.....	1
1.5 Overview.....	1
2. General Description.....	2
2.1 Product Perspective.....	2
2.2 Product Functions.....	2
2.3 User Characteristics.....	2
2.4 General Constraints.....	2
2.5 Assumptions and Dependencies.....	2
3. Specific Requirements.....	2
3.1 External Interface Requirements.....	3
3.1.1 User Interfaces.....	3
3.1.2 Hardware Interfaces.....	3
3.1.3 Software Interfaces.....	3
3.1.4 Communications Interfaces.....	3
3.2 Functional Requirements.....	3
3.2.1 <Functional Requirement or Feature #1>.....	3
3.2.2 <Functional Requirement or Feature #2>.....	3
3.3 Use Cases.....	3
3.3.1 Use Case #1.....	3

3.3.2 Use Case #2.....	3
3.4 Classes / Objects.....	3
3.4.1 <Class / Object #1>.....	3
3.4.2 <Class / Object #2>.....	3
3.5 Non-Functional Requirements.....	4
3.5.1 Performance.....	4
3.5.2 Reliability.....	4
3.5.3 Availability.....	4
3.5.4 Security.....	4
3.5.5 Maintainability.....	4
3.5.6 Portability.....	4
3.6 Inverse Requirements.....	4
3.7 Design Constraints.....	4
3.8 Logical Database Requirements.....	4
3.9 Other Requirements.....	4
4. Analysis Models.....	4
4.1 Sequence Diagrams.....	5
4.3 Data Flow Diagrams (DFD).....	5
4.2 State-Transition Diagrams (STD).....	5
5. Change Management Process.....	5

1. Introduction

1.1 Purpose

Метою даного документа є створення онлайн додатку Списку, який дає можливість створювати свої списки справ, списки покупок товарів в магазині.

1.2 Scope

Just do it! App це додаток для створення списків справ, списків товарів для покупки в магазині. Мета цього додатку в тому, щоб можливо було зручно слідкувати за виконанням своїх справ та планування покупок, їх відмітка та оновлення нових завдань. Є можливість додавання декількох списків, редагування, видалення, зміна статусів справ, фільтрація, збереження на пристрої. Після авторизації в застосунку з'являється можливість мати декілька списків та синхронізувати їх на декількох пристроях.

1.3 Definitions, Acronyms, and Abbreviations

Term	Definition
Незареєстрований користувач	Користувач, який може користуватись додатком без переваг зареєстрованого користувача(тільки один список на одному пристрої).
Зареєстрований користувач	Користувач який зареєструвався в додатку, має повний доступ до всіх функцій додатку.
Веб-сервер	сервер, що приймає HTTP-запити від клієнтів, зазвичай веб-браузерів, і видає їм HTTP-відповіді, як правило, разом із HTML-сторінками
SRS	Специфікація Вимог до Програмного Забезпечення (Software Requirement Specification)
PostgreSQL	вільна система керування реляційними базами даних

1.4 References

Document title	Last update	Source
Product Backlog		
Sprint Backlog 1		
Sprint Backlog 2		
Sprint Backlog 3		
Sprint Backlog 4		

1.5 Overview

Решта цього документа складається з трьох розділів. Другий містить огляд функціональності системи. У цьому розділі також описано різні типи користувачів та їхні привілеї. Цей розділ також містить системні обмеження та припущення щодо продукту. Третій розділ надає детальну специфікацію вимог. У четвертому розділі йдеться про пріоритетність вимог. У п'ятому розділі описано, як змінюються та затверджуються вимоги до проекту. Він містить загальну інформацію про процес управління.

2. General Description

2.1 Product Perspective

Система складатиметься з трьох частин: веб-додатку користувача, веб-сервера та сервера бази даних. Веб-програма користувача як зовнішня програма на стороні клієнта, серверна веб-програма як серверна, база даних для зберігання даних. Сервер баз даних містить базу даних PostgreSQL, яка використовується для зберігання даних. Клієнт не може безпосередньо спілкуватися з сервером бази даних. Тільки веб-сервер повинен мати доступ до бази даних. Коли клієнту потрібні дані, він формує та надсилає запит до веб-сервера. Далі веб-сервер інтерпретує запит і повертає помилку, якщо запит не може бути виконаний, інакше він формує та надсилає запит на сервер бази даних. База даних повертає помилку або дані, а потім веб-сервер реорганізує цю інформацію необхідним чином і надсилає назад клієнту.

2.2 Product Functions

- Реєстрація (Зареєструватися)
- Авторизація (Авторизуватися)

- Створити список
- Додати/Видалити пункти
- Редагувати та змінити статус
- Фільтрувати за статусом
- Зберегти список на пристрої користувача
- Зберегти список на сервері
- Видалити групою

2.3 User Characteristics

Існує два типи користувачів, які взаємодіють з системою: не зареєстровані користувачі, які можуть створити лише один список, який не можна переглянути на інших пристроях та зареєстровані користувачі, які можуть створювати до 10 списків одночасно та керувати ними з різних пристроїв після авторизації.

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

Не зареєстровані користувачі можуть створити лише один список, який не можна переглянути на інших пристроях. Зареєстровані користувачі можуть створювати до 10 списків одночасно та керувати ними з різних пристроїв після авторизації

3.1.2 Hardware Interfaces

Не застосовується.

3.1.3 Software Interfaces

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

3.1.4 Communications Interfaces

Щоб користувач мав доступ до веб-системи, необхідно встановити підключення до Інтернету.

3.2 Functional Requirements

3.2.1 Зареєструватися

3.2.1.1 Introduction

Не зареєстровані користувачі не мають доступ до всіх функцій додатка, це обмежено технічними можливостями. Щоб розпочати процес реєстрації, користувач повинен ввести електронну пошту, пароль та ім'я користувача.

3.2.1.2 Inputs

Текст: email (6-32 символи), ім'я користувача(4-32 символи), пароль (6-32 символи), повторити пароль (6-32 символи).

3.2.1.3 Processing

При натисканні кнопки реєстрації клієнт надсилає запит із усіма вхідними даними на веб-сервер. Цей користувач записується в базі даних.

3.2.1.4 Outputs

Якщо форма була заповнена успішно, користувача переводить на сторінку створення списку. Якщо форма була заповнена невірно, з'являється повідомлення з підказкою, як треба ввести вірні дані.

3.2.1.5 Error Handling

Повідомлення з помилкою з'явиться в таких випадках:

- Електронна пошта не містить символ "@" або "."
- Ім'я користувача менше 2-х символів
- Пароль коротший за 6 символів або довший за 32 символи
- Пароль містить не латинські символи або спеціальні символи

3.2.1.6 Constraints

Не зареєстровані користувачі.

3.2.2 Авторизуватися

3.2.2.1 Introduction

Неавторизовані користувачі можуть увійти, якщо вони мають зареєстрований профіль у системі. Щоб увійти, користувач повинен ввести електронну адресу.

3.2.2.2 Inputs

Текст: валідний email, пароль (6-32 символи).

3.2.2.3 Processing

Веб-сервер запитує базу даних для пошуку користувача з такою електронною поштою, якщо знаходить, то перевіряє пароль. Якщо дані вірні, то підтверджує вхід та відкриває сторінку списків.

3.2.2.4 Outputs

Якщо вхід виконано успішно, система відображає списки користувача або якщо їх немає, то форму створення списку.

3.2.2.5 Error Handling

Якщо вхід не вдається, перед формою входу з'являється повідомлення з помилкою.

3.2.2.6 Constraints

Неавторизовані/не зареєстровані користувачі.

3.2.3 Створити список

3.2.3.1 Introduction

Для створення списку користувачеві потрібно ввести назву.

3.2.3.2 Inputs

Текст: Назва (1-32 символи).

3.2.3.3 Processing

Створюється список після введення назви.

3.2.3.4 Outputs

Після створення списку користувач одразу бачить пункти створеного списку.

3.2.3.5 Error Handling

Якщо сталася помилка надати відповідне повідомлення з текстом.

3.2.3.6 Usertype

Всі типи користувачів.

3.2.4 Додати/видалити пункти

3.2.4.1 Introduction

Користувач, який створив список, може його редагувати: додавати, видаляти пункти списку.

3.2.4.2 Inputs

Текст: Назва (1-32 символи), Пункти (1-32 символи).

3.2.4.3 Processing

Користувач вибирає список для редагування через назву списку. Відкривається потрібний список в який додає нові чи видаляє існуючі пункти списку.

3.2.4.4 Outputs

Під час редагування списку користувачеві відображається відредагований список.

3.2.4.5 Error Handling

Якщо сталася помилка надати відповідне повідомлення з текстом.

3.2.4.6 Usertype

Всі типи користувачів.

3.2.5 Редагувати та змінити статус

3.2.5.1 Introduction

Користувач може змінювати статус вибраного пункта на “Completed”, а також відмінити статус “Completed”.

3.2.5.2 Inputs

Флажок пункту списку (два стани).

3.2.5.3 Processing

Користувач ставить галочку навпроти вибраного пункту, таким чином редагує статус пункта на “Completed”. Якщо йому потрібно змінити статус з “Completed” на “Active”, то він знімає флажок біля вибраного пункту списку.

3.2.5.4 Outputs

Під час редагування статусів пунктів користувачеві відображаються нові статуси пунктів.

3.2.5.5 Error Handling

Якщо сталася помилка надати відповідне повідомлення з текстом.

3.2.5.6 Usertype

Всі типи користувачів.

3.2.6 Фільтрувати за статусом

3.2.6.1 Introduction

Користувач може фільтрувати пункти списку вибравши відповідний фільтр.

3.2.6.2 Inputs

Кнопки фільтрів (“All”, “Active”, “Completed”)

3.2.6.3 Processing

За замовчуванням відображається статус “All”, при цьому відображаються усі пункти списку, на нього можливо переключити з інших фільтрів. Якщо користувач вибирає статус “Active”, то відображаються тільки пункти зі статусом “Active”. Якщо вибирає статус “Completed”, то відображаються тільки пункти зі статусом “Completed”.

3.2.6.4 Outputs

Під час фільтрації пунктів списку користувачеві відображаються відфільтровані пункти списку.

3.2.6.5 Error Handling

Якщо сталася помилка надати відповідне повідомлення з текстом.

3.2.6.6 Usertype

Всі типи користувачів.

3.2.7 Зберегти список на пристрої користувача

3.2.7.1 Introduction

Список зберігається на пристрої користувача.

3.2.7.2 Inputs

Список

3.2.7.3 Processing

Під час роботи користувача зі списками, вони зберігаються автоматично в браузері на пристрої користувача.

3.2.7.4 Outputs

Списки збережені автоматично

3.2.7.5 Error Handling

Якщо сталася помилка надати відповідне повідомлення з текстом.

3.2.7.6 Usertype

Неавторизовані/не зареєстровані користувачі.

3.2.8 Зберегти список на сервері

3.2.8.1 Introduction

Всі списки зареєстрованих користувачів зберігаються автоматично на сервері.

3.2.8.2 Inputs

Список

3.2.8.3 Processing

Під час роботи користувача зі списками, вони зберігаються автоматично на сервері, тому до них є доступ з різних пристроїв після авторизації в додатку.

3.2.8.4 Outputs

Списки збережені автоматично

3.2.8.5 Error Handling

Якщо сталася помилка надати відповідне повідомлення з текстом.

3.2.8.6 Usertype

Тільки зареєстровані користувачі.

3.2.9 Видалити групою

3.2.9.1 Introduction

Виділенні пункти списку можна видаляти групою.

3.2.9.2 Inputs

Виділені пункти списку.

3.2.9.3 Processing

Користувач може видалити групу виділених пунктів кнопкою “Clear Completed”.

3.2.9.4 Outputs

Видалені пункти зникнуть.

3.2.9.5 Error Handling

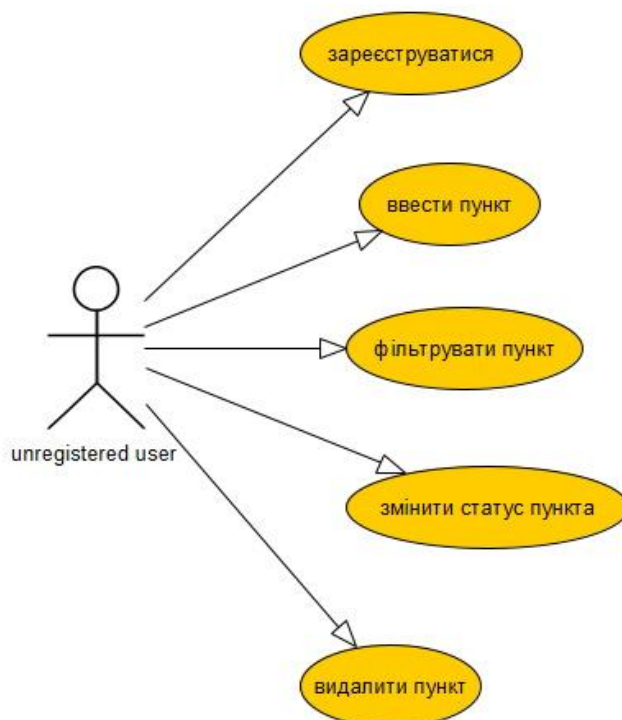
Якщо сталася помилка надати відповідне повідомлення з текстом.

3.2.9.6 Usertype

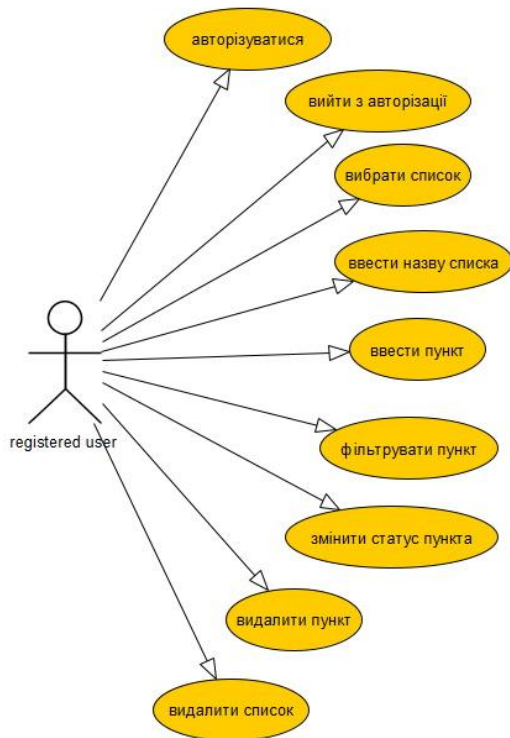
Всі типи користувачів.

3.3 Use Cases

3.3.1 Unregistered user Use Case



3.3.2 Registered user Use Case



3.4 Classes / Objects

3.4.1 User

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

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

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.

3.4.3.2 Functions

- toggleStatus(): Перемикає статус завдання між завершеним і незавершеним.
- renameTask(newTitle): Перейменовує 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 Availability

- Сервіс повинен бути доступний 24 години та 7 днів на тиждень.
- 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.

- Перевірки безпеки повинні проводитися двічі на рік для виявлення та усунення вразливостей.

3.5.5 Usability

- 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

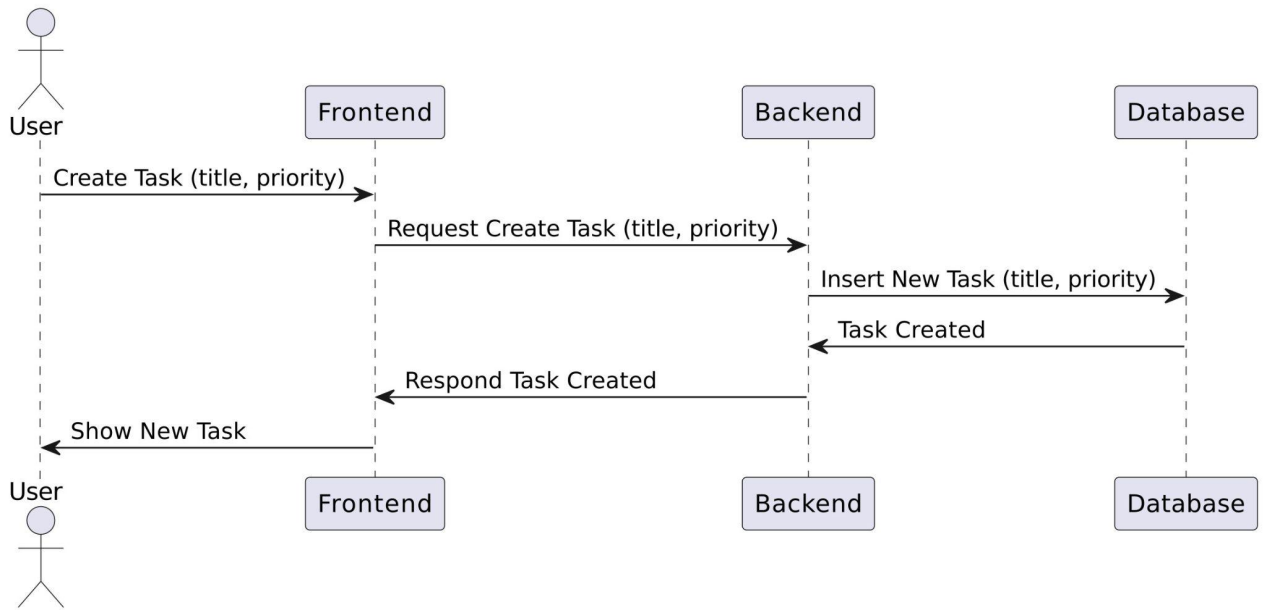
- PostgreSQL буде використовуватися для зберігання даних зареєстрованих користувачів та їхніх списків справ.
- Цілісність даних, збереження та можливості зберігання повинні відповідати найкращим практикам управління базами даних.

3.9 Other Requirements

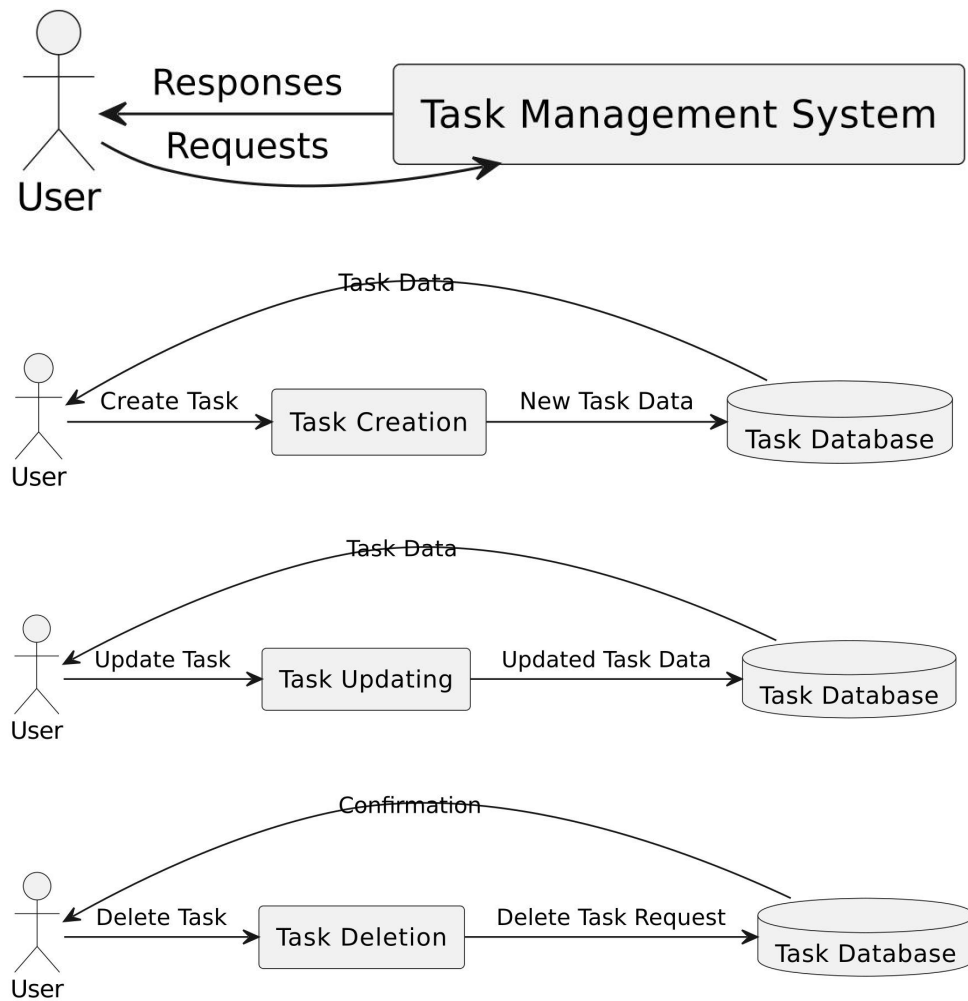
Не вказано.

4. Analysis Models

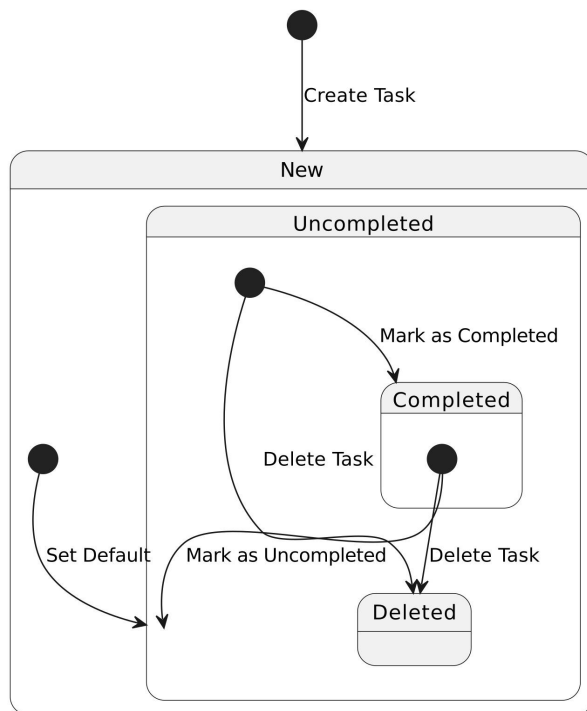
4.1 Sequence Diagrams



4.3 Data Flow Diagrams (DFD)



4.2 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.