Мета: підвищення ефективності сервісу управління завданнями, який задоволить усі потреби користувачів

Бізнес функції:

1. Реєстрація користувачів: функція для створення облікових записів користувачів.
2. Створиння списків завдань: можливість користувачів створювати списки завдань.
3. Створення завдань: можливість користувачів створення завдань.
4. Безпека та захист даних: функції для шифрування даних та ідентифікації і автентифікації користувачів.
5. Моніторинг і аналітика: функції для моніторингу стану системи, аналізу активності користувачів і виявлення аномалій.

Атрибути якості:

1. Безпека (Security)
2. Доступність (Availability)
3. Масштабованість (Scalability)

**Utility tree**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Атрибути якості | Уточнення атрибутів | Сценарій | Бізнес-цінність | Вплив на архітектуру |
| Security | Шифрування даних | Забезпечення шифрування збережених даних за допомогою шифру Blowfish | H | L |
| Availability | Час роботи | Досяжність 99,99% | H | M |
| Scalability | Час підключення | Користувачі підключені протягом 6-12 годин/день | H | M |
| Час відгуку системи на зростання навантаження | Швидкість відповіді системи на зростання кількості користувачів або операцій. | H | H |

**Six-part scenario for Security**

Стимул: Користувач намагається отримати доступ до облікового запису, використовуючи неправильний пароль.

Джерело стимулу: Користувач.

Артефакт: Система безпеки.

Середовище: Runtime.

Відгук: Система блокує доступ користувача.

Міра відгуку: Кількість блокувань доступу, час, що минув на блокування доступу.

**Six-part scenario for Availability**

Стимул: Користувач намагається отримати доступ до програми Todo list.

Джерело стимулу: Користувач.

Артефакт: Програма Todo list.

Середовище: Runtime.

Відгук: Програма Todo list доступна для користувача.

Міра відгуку: Час, який минув від моменту запуску програми до моменту отримання доступу користувачем.

**Six-part scenario for Scalability**

Стимул: Кількість користувачів, які одночасно використовують Todo list, збільшується.

Джерело стимулу: Зростання кількості користувачів.

Артефакт: Архітектура Todo list.

Середовище: Runtime.

Відгук: Система успішно обробляє всі запити від нових користувачів.

Міра відгуку: Час відгуку на запити користувачів, кількість помилок, що виникають під час роботи.

1. **Security**

Атрибут якості Безпека (Security) для проекту Todo list необхідний для забезпечення захисту даних користувачів, які зберігаються в додатку. До цих даних можуть належати:

* Імена користувачів
* Адреси електронної пошти
* Паролі

Атрибут безпеки має забезпечувати захист цих даних від несанкціонованого доступу, зміни або знищення.

Ідентифікуємо загрози за допомогою моделі STRIDE:

|  |  |
| --- | --- |
| **Тип загрози** | **Опис** |
| Спуфінг | Зловмисник отримує незаконний доступ до автентифікаційних данних іншого користувача, таких як ім’я користувача та пароль, для подальшого їх використанняу незаконних цілях |
| Незаконна зміна | Зловмисник отримує доступ до бази даних та підробляє дані користувачів |
| Відмова | Користувач відмовляється визнавати, що він створив певне завдання |
| Розкриття інформації | Користувачі можуть мати можливість переглядати чи редагувати завдання інших користувачів, якщо немає належного контролю доступу до завдань |
| Відмова в обслуговуванні | Атакуючий відправляє велику кількість запитів до сервера застосунку, завантажуючи його ресурси і споживаючи всю доступну пропускну здатність. Як результат, легітимні користувачі не можуть отримати доступ до сервера |
| Несанкціоноване отримання привілеїв | Зловмисники можуть змінити налаштування контролю доступу або використовувати слабкі місця в політиці безпеки для здійснення атак |

Для забезпечення безпеки використовується ідентифікація та автентифікація користувачів. Ідентифікуємо користувачів за логіном та автентифікуємо за допомогою паролю.

Для забезпечення безпеки користувачів, паролі шифруються за допомогою алгоритму в якому за основу взятий шифр Blowfish.

Blowfish – криптографічний алгоритм, який реалізує блочне симетричне шифрування.

Розроблений Брюсом Шнайєром в 1993 році. Являє собою шифр на основі мережі Фейстеля. Виконано на простих і швидких операціях: XOR, підстановка, додавання. Не запатентований і вільно поширюваний.

В нашому проекті для шифрування паролів ми використовуємо BCrypt.

BCrypt в C# – це адаптивна криптографічна функція формування ключа, яка використовується для безпечного зберігання паролів.

BCrypt має кілька переваг перед іншими функціями для зберігання паролів, зокрема:

* + Вона є адаптивною, тобто час її роботи можна налаштувати, щоб ускладнити атаки перебором.
  + Вона використовує сіль (salt), яка додається до пароля перед його хешуванням. Це ускладнює атаки з використанням райдужних таблиць.
  + Вона є стійкою до атак з використанням квантових комп'ютерів.

1. **Availability**

Атрибут якості Доступність (Availability) для проекту Todo list потрібен для того, щоб забезпечити доступність користувачів до списку завдань. Це означає, що користувачі повинні мати можливість створювати, переглядати, редагувати та видаляти завдання в будь-який час і з будь-якого місця, де є доступ до інтернету.

Атрибут Доступність (Availability) для проекту Todo list визначає наступне:

* Час доступності: Користувачі повинні мати можливість отримувати доступ до списку завдань 24 години на добу, 7 днів на тиждень.
* Місце доступності: Користувачі повинні мати можливість отримувати доступ до списку завдань з будь-якого місця, де є доступ до інтернету.
* Надійність доступу: Користувачі повинні мати можливість отримувати доступ до списку завдань без перерв або збоїв.

Приклади того, як атрибут Доступність (Availability) може бути реалізований в проекті Todo list:

* Використання хмарної інфраструктури: Хмарні сервіси, такі як Amazon Web Services або Microsoft Azure, забезпечують високу доступність і надійність.
* Використання реплікації даних: Реплікація даних дозволяє забезпечити доступ до списку завдань навіть у разі збоїв в одному з серверів.
* Використання балансування навантаження: Балансування навантаження дозволяє розподілити навантаження на кілька серверів, що також підвищує доступність.

1. **Scalability**

Атрибут якості Масштабованість (Scalability) для проекту Todo list потрібен для того, щоб забезпечити можливість розширення програми відповідно до зростання кількості користувачів або обсягу даних. Це означає, що програма повинна бути здатною підтримувати зростання навантаження без погіршення продуктивності або надійності.

Атрибут Масштабованість (Scalability) для проекту Todo list визначає наступне:

* Горизонтальна масштабованість: Програма повинна бути здатна масштабуватися шляхом додавання нових серверів.
* Вертикальна масштабованість: Програма повинна бути здатна масштабуватися шляхом підвищення продуктивності окремих серверів.

Приклади того, як атрибут Масштабованість (Scalability) може бути реалізований в проекті Todo list:

* Використання архітектури розподілених систем: Архітектура розподілених систем дозволяє масштабувати програму шляхом додавання нових серверів.
* Використання хмарної інфраструктури: Хмарні сервіси, такі як Amazon Web Services або Microsoft Azure, забезпечують можливість масштабування як горизонтально, так і вертикально.

Реалізація атрибуту Масштабованість (Scalability) вимагає ретельного планування та проектування. Важливо враховувати такі фактори, як:

* Очікуване зростання кількості користувачів та обсягу даних.
* Витрати на масштабування.
* Потенційні проблеми з безпекою та надійністю.