

Нефункціональні вимоги

- Вимоги до продуктивності(Performance Requirements)

StudyHub- це додаток, для підвищення продуктивності, розроблений, щоб студенти могли ефективно керувати своїм розкладом та завданнями. Саме тому система повинна:

1. ефективно реагувати в реальному часі: відображати розклад, надсилати сповіщення, оновлювати статуси завдань, завантажувати дошку.
2. забезпечувати синхронізацію даних, як розклад із розкладами на сайті ЛНУ, чи будь-які зміни в завданні.

(особливо під час періодів з особливим навантаженням, як початок навчального року або період сесії).

3. забезпечити безперебійну обробку одночасних користувачів.

- Програмні та системні вимоги

1. Мова інтерфейсу

- Основна мова інтерфейсу — українська.
- Додатково передбачена підтримка англійської (налаштовується в профілі користувача).
- Текст повідомлень і сповіщень повинен бути зрозумілим і лаконічним.

2. Вимоги до програмного забезпечення (ПЗ)

- Серверна частина:
 - ASP.NET Core 8.0 або новіше
 - Nginx як reverse-proxy/load balancer
 - Docker
- Клієнтська частина:
 - .NET MAUI з Blazor Hybrid для інтерфейсу

3. База даних

- Використовується PostgreSQL як основна СУБД.
- Redis для кешування часто змінюваних даних (розклад, статуси задач).
- Резервне копіювання БД відбувається щонайменше 1 раз на добу.

• Вимоги до безпеки(Safety Requirements)

Додаток є важливим джерелом інформації для студентів. Саме тому неправильна його робота може призвести до небажаних для успішності користувача наслідків.

1. Сповіщення повинні вчасно нагадувати про дедлайни
2. Розклад повинен докладно відображати поточні заняття
3. Статус завдань повинен постійно бути актуальним

• Вимоги до захисту(Security Requirement)

Аплікація має справу із особистою інформацією, тому потрібно:

1. Виключити неавторизований доступ до додатку
2. Запобігти втраті даних про завдання та графіки
3. Паролі повинні бути захешовані
4. Авторизовані користувачі повинні мати доступ до зміни лише власних даних

• Атрибути якості програмного забезпечення(Software Quality Attributes)

1. Availability: Додаток повинен бути доступним 24/7. Даунтайм може негативно відобразитися на успішності користувачів.
2. Correctness: Додаток повинен точно відображати актуальний графік, вчасно надсилати дедлайни.
3. Maintainability: Додаток повинен забезпечувати легке обслуговування, оновлення та виправлення помилок. Оскільки додаток призначений для довгострокового використання, його структура повинна бути простою в управлінні та модифікації за необхідності.
4. Usability: Інтерфейс додатку повинен бути простим та інтуїтивно-зрозумілим кожному користувачу.
5. Reliability: Додаток повинен бути стабільним та не виходити із ладу

6. Scalability: Додаток повинен витримувати навантаження, коли реєструється велика кількість користувачів, наприклад протягом літнього періоду вступу

- Критерії прийнятності(Acceptance criteria)

Ситуація: Створення нового профілю

Given: Користувач є новим, незареєстрованим студентом

When: Користувач реєструється

Then: Користувач може створити новий акаунт, обрати групу та користуватися доступними функціями

Ситуація: Взаємодія із дошкою

Given: Користувач має завдання

When: Користувач взаємодіє із дошкою

Then: Користувач може читати умову завдання, додавати дедлайн, змінювати його статус

Ситуація: Отримання сповіщення

Given: У користувача є не виконане завдання, яке необхідно здати у найближчі 24 год

When: дедлайн близько

Then: Користувач отримує сповіщення із нагадуванням

Ситуація: Отримання розкладу

Given: Користувач зареєстрований і вказав свою групу

When: Користувач відкриває вкладку "Розклад"

Then: Розклад автоматично завантажується із сайту університету та відображається у додатку

Ситуація: Користувач додає нове завдання

Given: Користувач авторизований і знаходиться на дошці завдань

When: Користувач натискає кнопку "Додати завдання" та заповнює форму (назва, дедлайн, опис)

Then: Завдання зберігається у системі та відображається на дошці

Ситуація: Користувач змінює статус завдання

Given: Користувач бачить завдання на дошці

When: Користувач змінює статус (наприклад, з "To Do" на "In Progress" або "Done")

Then: Система оновлює статус в реальному часі та синхронізує зміни для всіх учасників команди

Ситуація: Користувач знаходиться офлайн

Given: Користувач відкрив додаток без інтернет-з'єднання

When: Користувач переходить на дошку або розклад

Then: Користувач бачить останні синхронізовані дані (кешовану версію) з попереднього сеансу, із позначкою, що дані можуть бути неактуальні

Ситуація: Помилка завантаження розкладу

Given: Система не змогла отримати дані з сайту університету

When: Користувач відкриває вкладку "Розклад"

Then: Користувач бачить повідомлення про помилку та пропозицію спробувати оновити розклад пізніше

Ситуація: Доступ на кількох пристроях

Given: Користувач авторизований на кількох пристроях (телефон, планшет, ПК)

When: Користувач змінює завдання або розклад на одному пристрої

Then: Зміни синхронізуються у реальному часі та відображаються на всіх пристроях