

Функціональні вимоги

Опис програми

StudyHub - це платформа для студентів, яка поєднує індивідуальну роботу та взаємодію в команді. Кожен користувач може мати власний простір для завдань, а також приєднуватися до команд, де доступні спільна дошка Kanban та сховище матеріалів. Крім того, система підтримує групові розклади, які можна синхронізувати з Google Calendar, щоб зручно відстежувати заняття або терміни виконання завдань. Комунікація будується за допомогою повідомлень: запрошення до груп, нагадування про терміни або підтвердження виконаних завдань надсилаються до додатка та дублюються в боті Telegram. Для мотивації користувачів впроваджено систему балів: виконані завдання та похвала від інших учасників приносять додаткові бонуси. Це створює атмосферу здорової конкуренції та допомагає підтримувати високий рівень залученості до роботи команди.

Система складається з таких вікон:

- **Task Board** - дошка у форматі Kanban для роботи з завданнями.
- **My Schedule** - розклад занять із синхронізацією з сайтом університету.
- **Storage** - файлове сховище для навчальних матеріалів.
- **Profile** - профіль користувача з особистими даними та підключенням Telegram-бота.
- **Головна сторінка** - стартове меню з навігацією.

1. Реєстрація та автентифікація

1. Реєстрація нового користувача

Як користувач, я можу створити новий обліковий запис, щоб працювати з додатком.

Необхідні умови:

- а. Не існує облікового запису з таким e-mail.
- б. Обов'язкові поля: Ім'я, Прізвище, e-mail, Пароль, вибір групи, Підтвердження пароля.
- с. Реєстрація при стабільному з'єднанні займає <2 сек

- d. При неправильних даних, користувач бачить валідаційні помилки
- 2. Авторизація
 - Як користувач, я можу увійти у свій обліковий запис.
 - Необхідні умови:
 - a. Користувач не авторизований.
 - b. У системі вже існує обліковий запис із введеним e-mail.
 - c. Пароль перевіряється на відповідність.
 - d. Вхід у систему при стабільному з'єднанні займає <2 сек
 - e. Після 3 неправильного введення паролю, користувачу блокується спроба входу на 5 хв
- 3. Відновлення пароля
 - Як користувач, я можу скинути пароль, отримавши код підтвердження.
 - Необхідні умови:
 - a. Введено дійсний e-mail.
 - b. Код підтвердження надсилається на e-mail.
 - c. Код підтвердження при стабільному з'єднанні надходить <60 сек
 - d. Після введення коду можна задати новий пароль.
 - e. Зміна пароля займає <2 сек
 - f. Після неправильно введеного коду, користувач немає можливості змінити пароль протягом 10 хв

2. Task Board

- 4. Як користувач, я можу переглядати Kanban-дошку зі завданнями.
 - Необхідні умови:
 - a. Відображаються колонки: *To Do*, *In Progress*, *For Review*, *Done*.
 - b. Доступні кнопки:
 - i. Фільтр / Пошук - відбір завдань за статусом, датою чи текстом.
 - c. Відображення всіх колонок і задач, при стабільному з'єднанні займає <10 сек
- 5. Як користувач, я можу створювати нові завдання.
 - Необхідні умови:
 - a. Доступна кнопка Create Task.

- b. Поля: Назва, Опис, Предмет/Група, Дата дедлайну.
 - c. Після збереження завдання з'являється в колонці *To Do*.
 - d. Створення нового завдання займає <2 сек
 - e. При неправильних даних, користувач бачить валідаційні помилки
- 6. Як користувач, я можу редагувати власні завдання.
Необхідні умови:
 - a. Поля в вікні завдання можна змінювати в будь який час.
 - b. Редагування завдання займає <2 сек
 - c. При неправильних даних, користувач бачить валідаційні помилки
- 7. Як користувач, я можу змінювати статус завдання перетягуванням між колонками.
Необхідні умови:
 - a. Колонки повинні бути drag and drop.
 - b. Збереження змін відбуваються <2 сек
- 8. Як користувач, я можу переглядати детальну інформацію про завдання.
Необхідні умови:
 - a. Відображаються: назва, опис, дати, група, список вкладень, історія змін.
 - b. Відображення завдання відбувається <5 сек

3. My Schedule

- 9. Як користувач, я можу експортувати розклад у форматі PDF.
Необхідні умови:
 - a. Доступна кнопка "Import schedule as PDF".
 - b. Експорт розкладу <120 сек
 - c. При неправильному форматі файлу, користувач отримує валідаційну помилку

4. Storage

- 11. Як користувач, я можу завантажувати та зберігати навчальні файли.
Необхідні умови:
 - a. Кнопка Add files для додавання файлу.

- b. Можливість перегляду та видалення завантажених матеріалів.
 - c. Завантаження всіх файлів при стабільному з'єднанні <120 сек
 - d. Завантаження файлів при стабільному з'єднанні (lazy loading) <15 сек
 - e. При завантаженні файлу, більшого ніж 100Мб користувач отримує валідаційну помилку
12. Як користувач, я можу шукати та фільтрувати файли за назвою або типом.

5. Go to my profile

13. Як користувач, я можу переглядати та редагувати особисті дані.

Необхідні умови:

- a. Відображаються: Ім'я, Прізвище, e-mail, фото профілю.
 - b. При кліку за межами форми, зміни зберігаються у базі <5 сек.
 - c. При введенні неправильних даних, користувача отримує валідаційні помилки
14. Як користувач, я можу підключити Telegram-бота для отримання сповіщень.

Необхідні умови:

- a. Генерується токен/QR-код для підключення <10 сек.
- b. Підтвердження успішного з'єднання <5 сек.

6. Офлайн-режим

15. Як користувач, я можу переглядати останні синхронізовані дані без інтернету.

Необхідні умови:

- a. Доступ до кешованих завдань і розкладу з позначкою «Дані можуть бути не актуальні».

7. Навігація через навбар

17. Як користувач, я можу переходити між розділами за допомогою бокового меню:

- a. Task Board
- b. My Schedule
- c. Storage
- d. Profile

Необхідні умови:

- e. Підсвітка поточної вкладки.

8. Основна сторінка

18. Як користувач, я можу потрапити на основну сторінку з навігацією
Необхідні умови:

- a. Користувач повинен бути авторизованим
- b. Відображаються сторінки для навігації і кнопка з допоміжною інформацією
- c. Всі вкладки відображаються <1сек

Пріоритети функцій

Must: Реєстрація, авторизація, створення завдання, перегляд завдань, експорт розкладу, власний кабінет

Should: інтеграція з Telegram, сховище з файлами користувачів

Could: можливість зміни теми додатку, гейміфікація і досягнення

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. Uptime системи повинен становити мінімум 99.5% щомісяця (максимум 3.6 години простою на місяць). Планові технічні роботи повинні проводитися в нічний час не частіше 2 разів на місяць.
2. Correctness: Додаток повинен точно відображати актуальний графік, вчасно надсилати дедлайни. Точність відображення розкладу повинна становити 99.9%; Сповіщення про дедлайни повинні надходити мінімум за 24 години до дедлайну.
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. Код повинен бути покритий юніт-тестами мінімум на 70%; Середній час виправлення критичної помилки ≤ 48 годин. Нефункціональні оновлення не повинні порушувати роботу основних функцій (регресійне тестування обов'язкове).
4. Usability: Інтерфейс додатку повинен бути простим та інтуїтивно-зрозумілим кожному користувачу. Новий користувач повинен мати змогу зареєструватися та створити перше завдання ≤ 5 хвилин; Кількість кліків для додавання нового завдання ≤ 5 ; Користувач повинен зрозуміти інтерфейс без додаткової документації (рівень успішності юзабіліті-тестування $\geq 85\%$).

5. Reliability: Додаток повинен бути стабільним та не виходити із ладу. Система повинна витримувати мінімум 72 години безперервної роботи без збоїв; Максимальний час відновлення після аварії (MTTR) — не більше 2 годин.
6. Scalability: Додаток повинен витримувати навантаження, коли реєструється велика кількість користувачів, наприклад протягом літнього періоду вступу. Система повинна підтримувати зростання бази користувачів у 10 разів без зміни архітектури; Система повинна підтримувати одночасну роботу не менше 10 000 активних користувачів.

- 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

Ситуація: Користувач створює нове завдання з дедлайном в минулому або залишає поле порожнім

Given: Користувач авторизований і відкрив форму створення завдання

When: Користувач вводить дедлайн у минулому або залишає поле пустим

Then: Система показує повідомлення про помилку і не дозволяє зберегти завдання

Ситуація: Користувач багаторазово вводить некоректний пароль

Given: Користувач намагається увійти в систему

When: Користувач вводить неправильний пароль 3 рази поспіль

Then: Обліковий запис тимчасово блокується на 5 хвилин, і користувач отримує повідомлення про блокування

Ситуація: Перевантаження системи

Given: Система одночасно обробляє велику кількість користувачів (пікове навантаження)

When: Кількість активних з'єднань перевищує допустимий рівень

Then: Система пріоритезує критичні операції (сповіщення, дедлайни, розклад), а другорядні (наприклад, аналітика) можуть виконуватись із затримкою

Ситуація: Втрата з'єднання

Given: Користувач відкрив додаток і має нестабільне або відсутнє інтернет-з'єднання

When: Користувач виконує дію (перегляд розкладу, оновлення завдання)

Then: Система показує кешовані дані з позначкою «можуть бути неактуальні» та автоматично синхронізує зміни після відновлення підключення

Ситуація: Користувач намагається створити акаунт з вже зареєстрованим email

Given: Користувач відкрив форму реєстрації

When: Користувач вводить email, який вже існує у системі

Then: Система показує повідомлення про помилку «Користувач із таким email вже існує» та пропонує відновити пароль