**Planora - Нефункціональні вимоги**

**Мова інтерфейсу**

* **Первинна мова інтерфейсу:** Українська.
* **Локалізація:** Архітектура системи повинна забезпечувати можливість додавання підтримки інших мов інтерфейсу в майбутньому.

**Вимоги до програмного забезпечення (ПЗ) для запуску клієнтської частини**

**Десктоп-версія**

* **Операційні системи:**
  + Windows 10 і новіші.
  + macOS 11 (Big Sur) і новіші.
  + Linux (Ubuntu 20.04 LTS і новіші або інші дистрибутиви з підтримкою сучасних веб-технологій).

**Мобільний додаток**

* **Операційні системи:**
  + Android 9.0 (Pie) і новіші.
  + iOS 15 і новіші.

**Вимоги до апаратного забезпечення клієнтської частини**

**Для десктоп-версії**

* **Процесор:** 2-ядерний, 1.5 ГГц або вище.
* **Оперативна пам’ять:** 4 ГБ (рекомендовано 8 ГБ).
* **Вільне місце на диску:** 500 МБ для встановлення додатку.
* **Інтернет-з’єднання:** стабільне підключення до мережі Інтернет.

**Для мобільних пристроїв**

* **Оперативна пам’ять:** від 2 ГБ.
* **Вільне місце на диску:** 200 МБ.

**База даних**

* **Тип:** реляційна база даних (RDBMS).
* **Конкретна СКБД:** PostgreSQL.
* **Цілісність даних:** забезпечення за допомогою обмежень (constraints), транзакцій та ACID-властивостей.

### Безпека

* **Зберігання паролів:** усі паролі зберігаються у хешованому вигляді (алгоритм bcrypt або аналогічний).
* **Авторизація:** відбувається за допомогою рольової моделі доступу (Адміністратор, Викладач, Студент).
* **Шифрування з’єднань:** всі з’єднання між клієнтом і сервером шифруються протоколом HTTPS/TLS.
* **Стійкість до атак:** система повинна бути стійкою до найбільш поширених атак: SQL-ін’єкції, XSS, CSRF, brute-force.
* **Контроль доступу:** передбачене обмеження кількості спроб входу, тайм-аут сесії (30 хв бездіяльності) та rate limiting для API-запитів.
* **Валідація даних:** виконується як на клієнтській, так і на серверній стороні.

**Працездатність та надійність**

* **Доступність:** система доступна не менше 99% часу протягом навчального семестру (окрім запланованих технічних робіт).
* **Обробка помилок:** система має коректно реагувати на збої (наприклад, втрату зв’язку із сервером, невірні дані) та інформувати користувача зрозумілими повідомленнями.
* **Автоматичне відновлення:** після відновлення роботи серверів або з’єднання система має продовжувати виконання без втрати критичних даних користувача.
* **Логування:** усі критичні помилки та збої повинні логуватись із зазначенням часу та контексту для подальшого аналізу.
* **Резервне копіювання:** дані бази повинні автоматично зберігатися в резервні копії з періодичністю не рідше одного разу на добу.
* **Тестування навантаженням:** перед впровадженням у навчальний процес система має пройти перевірку на стабільність під типовим і піковим навантаженням.

### Продуктивність

* **Швидке завантаження:**
  + запуск додатку — не більше 10 секунд;
  + завантаження головної сторінки після авторизації — не більше 5 секунд;
  + відгук на основні операції (пошук, фільтрація, перегляд розкладу) — не більше 3 секунд.
* **Алгоритм генерації розкладу:** повинен обробляти дані для університету розміру (до 25 000 студентів, 2000 викладачів, 500 аудиторій) протягом не більше ніж 4 хвилин.
* **Паралельна робота користувачів:** додаток повинен коректно працювати при одночасному використанні системи щонайменше 5000 активними користувачами без суттєвих затримок.

### Масштабованість

* **Розширюваність:** архітектура системи повинна підтримувати можливість розширення функціоналу без потреби повної перебудови.
* **Обробка даних:** система має бути здатна обробляти зростаючий обсяг даних (збільшення кількості користувачів, викладачів, груп, аудиторій) без суттєвого зниження продуктивності.
* **Хмарне розгортання:** передбачена можливість розгортання системи у хмарному середовищі для забезпечення динамічного масштабування ресурсів (CPU, пам’ять, база даних).
* **Інтеграція:** архітектура повинна дозволяти в майбутньому інтеграцію з іншими університетськими системами через API.
* **Модульність:** нові компоненти та сервіси мають додаватися до системи без зупинки або переривання роботи існуючого функціоналу.