# **Вимоги до системи**

Система повинна являти собою навчальну програмну систему. Повинна містити інформацію доступну для вивчення, мати модуль перевірки знань. Інтерфейс має бути інтуїтивно зрозумілим та зручним у використанні.

4.1 Вимоги до системи в цілому

4.1.1 Вимоги до структури і функціонування системи

4.1.1.1 Перелік підсистем, їх призначення та основні характеристики

В системі є такі підмодулі:

● практичне завдання на видобування вимог;

● практичне завдання на виявлення “poor words”.

Також вважатимемо, що в системі є такі надмодулі:

● авторизація;

● адміністрування.

4.1.1.2 Вимоги до способів і засобів зв'язку для інформаційного обміну між компонентами системи

Для використання програмного продукту необхідно підключення до мережі інтернет, оскільки система являє собою програмний продукт, який матиме віддалену базу даних (на базі MS SQL Server) і клієнтську частину. яким треба буде спілкуватись між собою через сервер.

4.1.2 Вимоги до чисельності та кваліфікації персоналу системи

В системі повинен бути мінімум один адміністратор бази даних (наприклад, викладач). Інші користувачі - це учасники лабораторії. Керування системою буде проводитись адміністратором, який буде слідкувати за правильною та коректною роботою бази даних. Адміністратор повинен бути фахівцем та розуміти навчальну структуру, а також повинен мати знання в області керування та адміністрування бази даних.

4.1.3 Показники призначення

Наповнення є динамічним, тобто таким, що постійно змінюється. Теоретичний та практичний матеріал системи поповнюватиметься. Цільове призначення – навчання аналізу – збережеться на протязі декількох років, до моменту, коли інформація в системі не перестане бути актуальною.

4.1.4 Вимоги до надійності

Збереження цілісності даних системи є дуже важливим завданням. Не можна допустити ні втрату завдань та теоретичного матеріалу, ні результатів студентів. Дані системи мають зберігатись в БД. Для забезпечення цілісності, запити до БД повинні виконуватись у вигляді транзакцій. Також потрібно регулярно створювати бекапи даних з БД.

4.1.5 Вимоги до безпеки

Система повинна забезпечувати такий функціонал для забезпечення безпеки ПС:

● відновлення даних під час фізичного збою системи;

● збереження та шифрування персональних даних користувачів;

● кешування даних, для полегшеного відновлення інформації при збоях.

4.1.6 Вимоги до ергономіки та технічної естетики

При розробці інтерфейсу потрібно дотримуватись правил:

● в першу чергу - витончений мінімалізм та простота (можна за основу брати Material Design);

● зручність та мінімальна кількість дій для отримання результату.

4.1.7 Вимоги до захисту інформації від несанкціонованого доступу

Захист даних буде реалізовано з допомогою системи авторизації. Користувач матиме можливість зареєструватись в системі, або, вже маючи акаунт, увійти в систему.

4.1.8 Вимоги до збереженості інформації при збоях

При збоях будуть використані бекапи (точки відновлення) для повернення і збереження втрачених даних. Бекапи мають здійснюватись раз на тиждень опівночі.

4.1.9 Вимоги до стандартів і уніфікації

Перед початком розробки команда обирає єдиний стандарт вигляду коду (code-style) та дотримується його протягом всієї розробки. Також обов’язково дотримуватись стандартів розробки графічного інтерфейсу.

4.1.10 Додаткові вимоги

● користувач може на вибір перемикати колірні теми додатку;

● час відгуку програми має бути не більшим ніж 1 секунда за умови ідеального під’єднання до мережі Інтернет.

4.2. Вимоги до функціоналу системи

4.2.1. Практичне завдання на видобування вимог

4.2.1.1. Опис і пріоритет

Система дозволяє користувачу пройти практичне завдання, пов’язане з видобуванням вимог. Користувач читає абзац тексту і шукає в ньому вимоги і вносить їх в “пісочницю”, яка моделює систему для управління вимогами. Система перевіряє, наскільки співпадають введені дані з еталоном.

Пріоритет: високий.

4.2.1.2. Шлях доступу до даної функції

З головної сторінки вибрати плитку меню, що відповідає за даний тест та натиснути на неї. Після цієї дії користувач отримає сторінку для проходження практичного завдання.

4.2.1.3 Функціональні вимоги

R1. Додавання вимог

Користувач вводить ID вимоги, її назву, текст, пріоритет, ключові слова в відповідні текстові поля. Він натискає кнопку “Додати”. Проводиться валідація введених даних. Введена вимога додається в користувацьку БД вимог.

R2. Перегляд списку вимог

Користувач обирає кнопку “Переглянути всі” й отримує список з назвами вимог та їх ID.

R3. Перегляд конкретної вимоги

Користувач натискає на потрібну вимогу з списку (див. R2) й потрапляє на сторінку з детальною інформацією про дану вимогу.

R4. Видалення вимоги

Переглядаючи конкретну вимогу (див. R3), користувач натискає на кнопку “Видалити” і дана вимога зникає з списку вимог.

R5. Редагування вимоги

Переглядаючи конкретну вимогу (див. R3), користувач натискає на кнопку “Редагувати” і отримує сторінку з формою для редагування параметрів вимоги. Він вводить змінені дані у відповідні поля форми. Тоді він натискає на кнопку “Зберегти”. Система проводить валідацію введених даних і вимога оновлюється в користувацькій БД.

R6. Надсилання відповіді на перевірку

Користувач натискає на кнопку “Надіслати на перевірку” і йому відображається сторінка з результатом відсоткового співпадіння з еталоном.

4.2.2. Практичне завдання на виявлення “poor words”

4.2.2.1. Опис і пріоритет

Система дозволяє користувачу пройти практичне завдання, пов’язане з виявленням так званих “poor words”, тобто слів, які є небажаними в описі вимог. Користувач має БД з введеними вимогами. В цьому завданні користувач має інтерфейс, подібний до того ж, що й в завданні 4.2.2. Він переглядає список вимог, а також кожну конкретну вимогу. В цих вимогах користувач має змогу натискати на слова й позначати їх, як “poor words”.

Пріоритет: високий.

4.2.2.2 Шлях доступу до даної функції

З головної сторінки вибрати плитку меню, що відповідає за даний тест та натиснути на неї. Після цієї дії користувач отримає сторінку для проходження практичного завдання.

4.2.2.3. Функціональні вимоги

R7. Позначення слова, як “poor word”

Користувач, переглядаючи конкретну вимогу (див. R3), натискає на слово, яке він ідентифікує, як “poor word”. В модальному вікні він натискає кнопку “Позначити”. Позначене слово отримує червоне забарвлення для ідентифікації.

R8. Скасування позначки “poor word”

Користувач, переглядаючи конкретну вимогу (див. R3), натискає на слово, яке він ідентифікує, як “poor word”. В модальному вікні він натискає кнопку “Забрати позначку”. Слово втрачає червоне забарвлення.

R9. Надсилання відповіді на перевірку

Користувач натискає на кнопку “Надіслати на перевірку” і йому відображається сторінка з результатом відсоткового співпадіння з еталоном.

4.2.3. Профіль з історією проходження завдань

4.2.3.1. Опис і пріоритет

Система дозволяє користувачу переглянути таблицю з результатами попередніх проходжень завдань, а також дізнатись свій рівень.

Пріоритет: середній.

4.2.3.2 Шлях доступу до даної функції

З головної сторінки вибрати плитку меню, що відповідає за профіль користувача.

4.2.3.3. Функціональні вимоги

R10. Перегляд результатів проходження завдань

Користувач переглядає таблицю, в якій зображається інформація про назву завдання, його рівень складності, а також отриманий результат у відсотках.

R11. Перегляд рівня користувача

Користувач має змогу дізнатися свій рівень навичок в системі, від якого залежить складність доступних завдань.

4.2.4. Проходження тесту на визначення початкового рівня

4.2.4.1. Опис і пріоритет

Система дозволяє користувачу пройти тест, який допоможе визначити початковий рівень користувача, відповідно до якого користувач буде мати доступні завдання. Цей тест має проходитись користувачем, при його першому вході в систему.

Пріоритет: низький.

4.2.4.2 Шлях доступу до даної функції

Дана функція автоматично стає доступна й пропонується новим користувачам, при їх першому вході в систему.

4.2.4.3. Функціональні вимоги

R12. Проходження тесту

Користувач проходить тест, обираючи правильні відповіді серед запропонованих до заданого питання. Обравши відповідь серед опцій, він натискає кнопку “Наступне”.

R13. Надсилання відповідей для визначення рівня

Користувач, пройшовши тест та обравши відповіді на всі питання, натискає кнопку “Завершити” й отримує свій результат. Отриманий рівень визначає, завдання якої складності будуть доступні користувачу з початку.

4.3. Вимоги до забезпечення системи

4.3.1. Вимоги до алгоритмічного забезпечення системи

Система повинна мати в своєму розпорядку алгоритми пошуку та впорядкування інформації за певними критеріями.

4.3.2. Вимоги до інформаційного забезпечення системи

Система повинна бути реалізована дотримуючись архітектурного патерна MVC. Доступ до даних БД повинні здійснюватися окремо в конкретному модулі. Дані в БД повинні бути нормалізовані для уникнення функціональних залежностей між сутностями. Повинні бути розроблені транзакції для доступу до даних, щоб запобігти виникненню певних проблем на рівні доступу до даних. Повинні здійснюватися регулярні бекапи даних для уникнення можливих втрат.

4.3.3. Вимоги до лінгвістичного забезпечення системи

Серверна частина ПЗ буде розроблятися за допомогою технології компанії Microsoft ASP. NET Core мовою програмування С#. Клієнтська частина буде розроблена за допомогою фреймворка Angular з використанням технології React JS за допомогою мов HTML/CSS та мови програмування JavaScript.

4.3.4. Вимоги до програмного забезпечення системи

В якості СУБД використовуватиметься MSSQL Server 2019 та інструмент для роботи з нею Microsoft SQL Server Management Studio.

4.3.5. Вимоги до технічного забезпечення системи

Для користування програмним продуктом у користувача необхідний бути доступ до мережі інтернет.

4.3.6. Вимоги до організаційного забезпечення

Система повинна адекватно реагувати на помилковий ввід даних користувачами клієнтами, та сповіщати їх про це (з пропозиціями та порадами).