

НАЦІОНАЛЬНИЙ ТЕХНІЧНИЙ УНІВЕРСИТЕТ УКРАЇНИ
«Київський політехнічний інститут імені Ігоря Сікорського»

ФАКУЛЬТЕТ ПРИКЛАДНОЇ МАТЕМАТИКИ

Кафедра програмного забезпечення комп'ютерних систем

ТЕХНІЧНЕ ЗАВДАННЯ НА КУРСОВУ РОБОТУ
з дисципліни "Основи програмування"
на тему:
***“Програмний додаток електронної бази питань
і відповідей”***

Виконала: Грабовська Анастасія Дмитрівна
Група: КП-01

Затверджено

2 семестр 2020/2021

Зміст

1. Вступ	3
1.1. Найменування програми	3
1.2. Призначення і галузь застосування	3
2. Вимоги до програми	4
2.1. Вимоги до функціональних характеристик	4
2.2. Вимоги до надійності	5
2.2.1. Відмови через некоректні дії користувачів системи	5
3. Умови експлуатації	6
3.1. Кліматичні умови експлуатації	6
3.2. Вимоги до складу і параметрів технічних засобів	6
4. Вимоги до програмної документації	6
4.1. Попередній склад програмної документації	6
5. Техніко-економічні показники	6
5.1. Економічні переваги розробки	6
6. Стадії і етапи розробки	7
6.1. Стадії розробки	7
6.2. Етапи розробки	7
7. Порядок контролю і приймання	7
7.1. Види випробувань	7
7.2. Загальні вимоги до приймання роботи	8

1. Вступ

1.1. Найменування програми

Найменування програми: “Програмний додаток електронної бази питань і відповідей”.

1.2. Призначення і галузь застосування

Програма призначена для створення, керування вмістом бази даних, що містить такі дані:

1. Користувачів додатку
2. Питання
3. Відповіді

Програма надає консольний інтерфейс користувача для зручного керування записами в базі даних.

Зв'язки:

- Користувач-питання **one2many**
- Користувач-відповідь **one2many**
- Питання-відповідь **one2many**

Авторизація:

- Користувач є автором питань і відповідей
- Користувачі ролі “Модератор” можуть видаляти будь-які питання та відповіді

Додаткові функціональні вимоги:

- Автор питання може встановити у питанні ідентифікатор однієї відповіді, яка йому допомогла. Автор може очистити це значення.

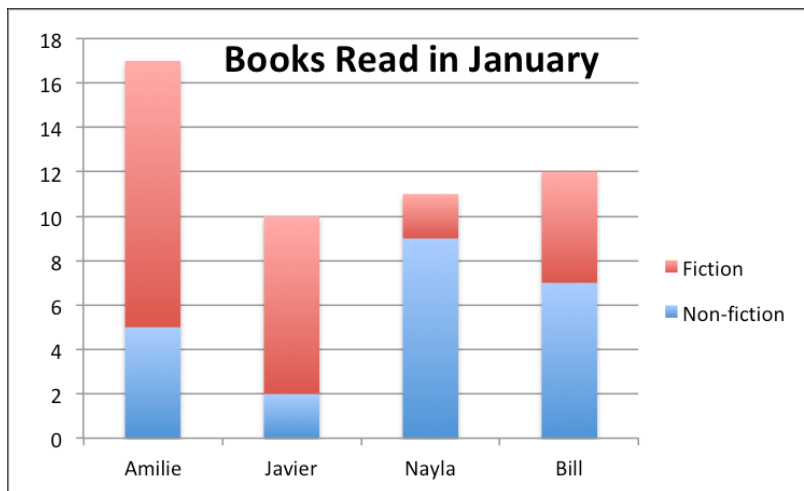
Експорт:

- Дані розбити на кілька XML файлів, у кожному файлі зберегти масив даних тільки одного типу, наприклад, **stores.xml**, **books.xml**.
- Згенеровані файли заархівувати у один архівний файл.
- Задати часовий проміжок і експортувати всі питання, у яких є прийнята відповідь, разом з цією відповіддю у форматі XML.

Імпорт:

- Якщо при імпорті у БД додатку вже є дані з такими ідентифікаторами - ігнорувати імпортовані дані

- Якщо при імпорті у даних є ідентифікатори на неіснуючі записи у БД, створити записи з такими ідентифікаторами та додати до них спеціальну помітку *"imported"*.
- Можна імпортувати експортовані питання з прийнятими відповідями.
Генерація зображень:
- Згенерувати **stacked bar** графік, що показує загальну кількість питань протягом певного проміжку часу і кількість питань, що мають закріплену відповідь:



Генерація звіту:

- З інформацією про питання і відповіді протягом часового проміжку:
 - Дата початку
 - Дата закінчення
 - Кількість питань протягом проміжку
 - Кількість питань протягом проміжку, що мають закріплену відповідь
 - Проміжок часу, коли було найбільше питань із їх кількістю
 - Проміжок часу, коли було найбільше закріплених відповідей із їх кількістю
 - Питання з найбільшою кількістю відповідей в цей період, що має закріплену відповідь
 - Зображення з графіком

2. Вимоги до програми

Перед входом у систему користувач проходить аутентифікацію.

Система повинна забезпечувати одночасну роботу з одним сервером мінімум двох користувачів.

2.1. Вимоги до функціональних характеристик

Програма має задовольняти такі вимоги:

1. Наявний консольний інтерфейс користувача. Команди використані відповідно до їхнього основного призначення.
 - a. Наявно декілька станів інтерфейсу користувача.
 - b. Дотримується єдиний стиль для всіх команд
2. Наявні стани для створення і редагування даних.
 - a. Можна видаляти дані (з підтвердженням цієї дії).
 - b. Наявний стан редагування даних має мати вже заповнені поля із попередніми значеннями. Є можливість відмінити оновлення сутності.
3. Використання списків та таблиць для показу колекцій даних.
 - a. Дані в таблиці не є лише відображенням однієї таблиці БД, а містять склеєні чи змінені дані.
 - b. Дані у списках та таблицях пагінуються, доступний пошук та фільтрація їх записів.
4. Всі вхідні дані від користувача та з файлів перевіряються, помилки обробляються, користувачу відображається інформація про помилки.
5. Програма має можливість завантаження зображень з файлової системи, їх збереження у сховище даних та їх показ у інтерфейсі користувача.
6. Програма має можливість експорту-імпорту даних у форматі даних XML. Є можливість зберігання даних у файлах, та передача даних до інших процесів через протокол TCP.
7. Програма взаємодіє з реляційною базою даних. SQL запити до бази даних не конкатенуються, а їх виконання винесено у спеціальний окремий модуль (вид репозиторія).
 - a. База даних містить мінімум 3 таблиці. Таблиці в БД приведені до 1-ї та 2-ї нормальних форм.

- b. Між таблицями бази даних є one-to-many та/або many-to-many зв'язки
- 8. Наявна реєстрація користувачів у системі. Паролі користувачів не зберігаються у відкритому виді, а хешуються обраним алгоритмом хешування рядків.
 - a. Наявний стан аутентифікації. Пароль користувача при введенні прихований. Цей стан має сповіщення про некоректність введених користувачем даних.
 - b. Програма авторизує доступу до даних по користувачах. У базі є дані, що належать конкретному користувачу (наприклад, авторство сутностей).
- 9. Система дозволяє генерувати зображення та файлові звіти на основі даних, що зберігаються у БД.
- 10. Використання окремих потоків виконання для паралельних фонових дій.
- 11. Використання клієнт-серверної архітектури програм та TCP сокетів (розділення системи на клієнт і сервер з передачею даних у одному із стандартних форматів даних)
- 12. Частина проекту винесена у власні бібліотеки класів. Винесення саме цього коду обґрунтоване.
- 13. Наявний проект з модульними тестами функціональності системи.

2.2. Вимоги до надійності

2.2.1. Відмови через некоректні дії користувачів системи

Відмови програми внаслідок некоректних дій користувача при взаємодії з програмою через командний інтерфейс неприпустимі.

3. Умови експлуатації

3.1. Кліматичні умови експлуатації

Кліматичні умови експлуатації, при яких повинні забезпечуватися задані характеристики, повинні задовольняти

вимогам, що пред'являються до технічних засобів в частині умов їх експлуатації.

3.2. Вимоги до складу і параметрів технічних засобів

До складу технічних засобів повинен входити персональний комп'ютер, що включає в себе:

1. Оперативну пам'ять об'ємом 1 Гігабайт, не менше.
2. Операційну систему Windows 10 або Ubuntu Linux версії 18.04 чи вище.

4. Вимоги до програмної документації

4.1. Попередній склад програмної документації

Склад програмної документації повинен містити:

1. Технічне завдання
2. Звіт виконання

5. Техніко-економічні показники

5.1. Економічні переваги розробки

Орієнтовна економічна ефективність не розраховується. Аналогія не проводиться зважаючи на унікальності пропонованих вимог до розробки.

6. Стадії і етапи розробки

6.1. Стадії розробки

Розробка повинна бути проведена в три стадії:

1. Розробка технічного завдання
2. Робоче проектування
3. Захист курсової роботи

6.2. Етапи розробки

На стадії розробки технічного завдання повинен бути виконаний етап розробки і затвердження даного технічного завдання.

На стадії робочого проектування повинні бути виконані перелічені етапи робіт:

1. Розробка програми
2. Розробка програмної документації (звіту)
3. Тестування програми.

На стадії захисту курсової роботи повинні бути виконані етапи підготовки результатів виконання і захист результатів з демонстрацією їх Викладачу.

7. Порядок контролю і приймання

7.1. Види випробувань

Приймально-здавальні випробування повинні проводитись на об'єкті Студента або Викладача в обумовлені терміни.

Приймально-здавальні випробування включають збірку розробленої програми із проекту репозиторія Студента та демонстрації всієї розробленої функціональності.

7.2. Загальні вимоги до приймання роботи

У випадку виникнення труднощів в реалізації функцій програми, ТЗ може бути відкориговане. Пропозиції щодо зміни і доповнення ТЗ можуть вносити як Студент так і Викладач.