НАЦІОНАЛЬНИЙ ТЕХНІЧНИЙ УНІВЕРСИТЕТ УКРАЇНИ

«Київський політехнічний інститут імені Ігоря Сікорського»

ФАКУЛЬТЕТ ПРИКЛАДНОЇ МАТЕМАТИКИ

Кафедра програмного забезпечення комп’ютерних систем

**КУРСОВА РОБОТА**

***з дисципліни "Основи програмування"***

Виконала: Грабовська Анастасія Дмитрівна

Група: КП-01

Допущено до захисту

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

2 семестр 2020/2021

НАЦІОНАЛЬНИЙ ТЕХНІЧНИЙ УНІВЕРСИТЕТ УКРАЇНИ

«Київський політехнічний інститут імені Ігоря Сікорського»

ФАКУЛЬТЕТ ПРИКЛАДНОЇ МАТЕМАТИКИ

Кафедра програмного забезпечення комп’ютерних систем

|  |  |
| --- | --- |
| Узгоджено  Керівник роботи    \_\_\_\_\_\_\_\_\_\_/Гадиняк Р.А./ | ЗАХИЩЕНА "\_\_"\_\_\_\_\_\_\_\_\_2021р.  з оцінкою\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_  \_\_\_\_\_\_\_\_\_\_/Гадиняк Р.А./ |

**Програмний додаток електронної бази питань і відповідей**

Виконавець роботи

Грабовська Анастасія Дмитрівна

\_\_\_\_\_\_\_\_\_\_\_\_\_\_2021р.

Зміст

[**Вступ**](#_cr2eoqpdh4mm) **4**

[**1. Аналіз предметної галузі**](#_b73pd3816f8p) **5**

[1.1. Аналіз вимог до функціональності програмних засобів](#_qz6vreit6wm5) 5

[**2. Аналіз мов програмування та технологій розроблення**](#_ewtylcuh8bvc) **6**

[2.1. Мова програмування С#](#_fh0j2unhluf2) 6

[2.2. Фреймворк .NET](#_ayd32pcv0jlc) 7

[2.3. База даних SQLite](#_hul0g052l08h) 7

[2.4. Бібліотека Terminal.Gui](#_6jqqk9eelxe)  7

[2.5. Бібліотека Microsoft.Data.Sqlite](#_28bxse9w7ext)  7

[2.6. Бібліотека ScottPlot](#_6jqqk9eelxe)  7

[**3. Опис розроблених програмних засобів**](#_kbi4mqs5pjyt) **8**

[3.1. Загальна структура системи](#_nlu4ckytvy2k) 8

[3.2. Структура бази даних](#_ny4p6qq0v8a9) 10

[3.3. Модулі репозиторіїв даних](#_p5yku72ilzdo) 11

[3.4. TCP сервер і опис RPC взаємодії](#_ddryso9qa52o) 13

[3.5. Модуль експорту](#_bl0vwivvre05) 13

[3.6. Модуль імпорту](#_bl0vwivvre05) 13

[3.7. Модуль аутентифікації](#_i6br8reh68e2) 13

[3.8. Бібліотека ClassLibrary](#_vkb2811r6l9f) 13

[**4. Аналіз розроблених програмних засобів**](#_g0l8gqqu407h) **14**

[4.1. Особливості реалізації](#_5ajysoaxe2s5) 14

[4.2. Інтерфейс користувача](#_c8eyxdz8sp3u) 15

[**Висновки**](#_6dbqg2jsslr5) **21**

# Вступ

В процесі розроблення програмного забезпечення часто виникає необхідність пошуку додаткової інформації, яка допомагає реалізувати бажаний функціонал.

Проблемами пошуку допоміжної інформації є ненадійність джерел, неструктурованість та застарілість інформації. Виникає необхідність створення бази даних та додатку питань і відповідей, яка б спростила процес пошуку інформації та надала б можливість ставити питання іншим користувачам, якщо дане питання не було розглянуте раніше. Саме тому пропонується розробка програмних засобів програмного додатку електронної бази даних питань і відповідей.

Створення системи, що забезпечує доступ до електронної бази даних питань і відповідей, є актуальною задачею.

Даний проект присвячено розробленню програмного додатку електронної бази даних питань і відповідей, який призначено для управління питаннями, відповідями на них, а також користувачами, що можуть публікувати ці питання та відповіді.

# 1. Аналіз предметної галузі

## 1.1. Аналіз вимог до функціональності програмних засобів

В процесі аналізу вимог до системи було виділено такі функціональні вимоги:

1. Наявний консольний інтерфейс користувача. Команди використані відповідно до їхнього основного призначення.
   1. Наявно декілька станів інтерфейсу користувача.
   2. Дотримується єдиний стиль для всіх команд
2. Наявні стани для створення і редагування даних.
   1. Можна видаляти дані (з підтвердженням цієї дії).
   2. 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.
   1. 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 експорту-імпорту даних у форматі даних XML. Є можливість зберігання даних у файлах, та передача даних до інших процесів через протокол TCP.
6. Програма взаємодіє з реляційною базою даних. SQL запити до бази даних не конкатенуються, а їх виконання винесено у спеціальний окремий модуль (вид репозиторія).
   1. База даних містить мінімум 3 таблиці. Таблиці в БД приведені до 1-ї та 2-ї нормальних форм.
   2. Між таблицями бази даних є one-to-many та/або many-to-many зв'язки
7. Наявна реєстрація користувачів у системі. Паролі користувачів не зберігаються у відкритому виді, а хешуються обраним алгоритмом хешування рядків.
   1. Наявний стан аутентифікації. Пароль користувача при введені прихований. Цей стан має сповіщення про некоректність введених користувачем даних.
   2. Програма авторизує доступу до даних по користувачах. У базі є дані, що належать конкретному користувачу (наприклад, авторство сутностей).
8. Система дозволяє генерувати зображення та файлові звіти на основі даних, що зберігаються у БД.
9. Використання окремих потоків виконання для паралельних фонових дій.
10. Використання клієнт-серверної архітектури програм та TCP сокетів (розділення системи на клієнт і сервер з передачею даних у одному із стандартних форматів даних)
11. Частина проекту винесена у власні бібліотеки класів. Винесення саме цього коду обгрунтоване.

# 2. Аналіз мов програмування та технологій розроблення

## 2.1. Мова програмування С#

Мова C# має багато можливостей, що полегшують реалізацію програмного забезпечення. При розробленні програми було використано:

1. Платформу .NET
2. Базу даних SQLite, яка підтримується мовою C#
3. Бібліотеку для реалізації інтерфейсу Terminal.GUI
4. Бібліотеку для взаємодії з базою даних SQLite
5. Бібліотеку ScottPlot, яка реалізує генерацію графіків

## 2.2. Фреймворк .NET

## Фреймворк .NET має такі можливості:

## Кожна бібліотека в .NET має свідчення про свою версію, що дозволяє усунути можливі конфлікти між різними версіями збірок

## .NET — крос-платформова технологія, в цей час існує реалізація для платформи Microsoft Windows і варіант технології для ОС Linux

## Компілятори для .NET випускаються багатьма фірмами.

## .NET поділяється на дві основні частини — середовище виконання та інструментарій розробки.

## 2.3. База даних SQLite

SQLite є ідеальним рішенням для багатьох завдань, в тому числі для зберігання даних розробленого додатка. Серед них:

1. База даних зберігається в одному крос-платформовому файлі
2. Підтримка великих розмірів баз даних
3. Швидший за популярні рушії клієнт-серверних баз даних для найпоширеніших операцій
4. Простий, легкий у використанні API
5. Доступні прив'язка для мови C#
6. Крос-платформовість
7. Поставляється з автономним клієнтом інтерфейсу командного рядка, який може бути використаний для управління базами даних SQLite

## 2.4. Бібліотека Terminal.Gui

Terminal.Gui - це бібліотека, призначена для створення консольних програм на основі C#. Структура розроблена для спрощення написання програм, які працюватимуть на монохромних терміналах, а також сучасних кольорових терміналах з підтримкою миші. Ця бібліотека працює в ОС Windows, Linux та MacOS. Вона надає текстовий набір інструментів, який працює подібним чином до графічних. Існує багато елементів керування, які можна використовувати для створення програм, заснованих на подіях.

## 2.5. Бібліотека Microsoft.Data.Sqlite

Microsoft.Data.Sqlite - це легкий постачальник ADO.NET для SQLite. Ця бібліотека реалізує загальні абстракції ADO.NET для з'єднань, команд, зчитувачів даних тощо.

## 2.6. Бібліотека ScottPlot

ScottPlot - це безкоштовна бібліотека для побудови графіків із відкритим кодом для .NET, яка полегшує інтерактивне відображення великих наборів даних. Лінійні графіки, стовпчасті діаграми, кругові графіки, діаграми розсіювання та багато іншого можна створити лише за допомогою декількох рядків коду.

# 3. Опис розроблених програмних засобів

## 3.1. Загальна структура системи

Програмні засоби реалізовані у вигляді десктопного додатку з графічним інтерфейсом користувача. Структурна схема системи зображена на рис.

UI client

Local Server

SQLite DB

Рис. Структурна схема системи

UI client +

Server

SQLite DB

Рис. Структурна схема системи

Сервер системи обробляє запити від користувачів та виконує такі функції:

1. Реєстрація
2. Авторизація
3. Пошук та пагінований перегляд списку сутностей
4. Перегляд, додавання, редагування і видалення сутностей
5. Отримання даних для експорту та генерації звіту
6. Імпорту даних в БД

Проект розбитий на такі модулі:

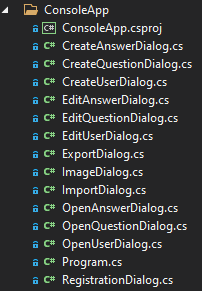
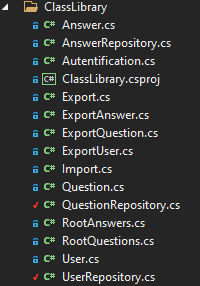
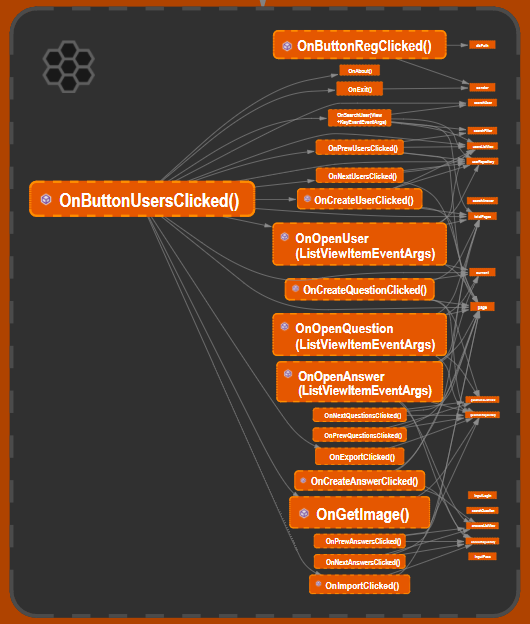


Рис. Модулі проекту

Діаграма залежностей модулів IDE:



## 3.2. Структура бази даних

На рис. представлена ERD-діаграма структури бази даних системи.

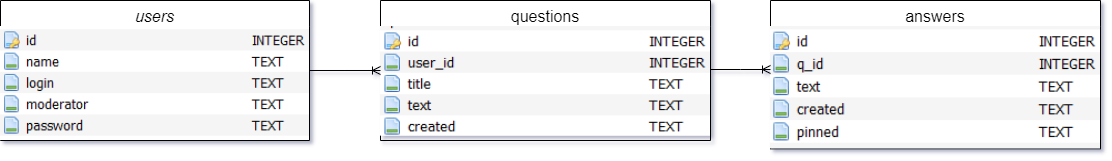


Рис. Структура бази даних

Таблиця користувачів містить ключ id (унікальне, ненульове автоінкрементоване значення), за яким здійснюється пошук, а також поля name, login, password i moderator.

Таблиця питань містить ключ id (унікальне, ненульове автоінкрементоване значення), за яким здійснюється пошук, а також поля user\_id, title, text i created. Поле user\_id забезпечує зв’язок з таблицею користувачів.

Таблиця відповідей містить ключ id (унікальне, ненульове автоінкрементоване значення), за яким здійснюється пошук, а також поля q\_id, text, created i pinned. Поле q\_id забезпечує зв’язок з таблицею питань.

## 3.3. Модулі репозиторіїв даних

Інтерфейс модуля:

|  |
| --- |
| **UserRepository.сs** |
| private SqliteConnection connection;  public int Insert(User user);  public User GetById(int id);  public User FindLogin(string login);  public int DeleteById(int id);  public int Update(int userId, User user);  public int GetTotalPages();  public List<User> GetPage(int pageNumber);  public List<User> GetAll();  public List<User> GetSearchUsers(string filter);  public int GetSearchPages(string filter);  public List<User> GetSearchPage(int pageNumber, string filter); |

## Модуль дозволяє взаємодіяти з таблицею користувачів. Він виконує базові CRUD операції, а також додаткові, які забезпечують пагінацію і пошук.

Інтерфейс модуля:

|  |
| --- |
| **QuestionRepository.сs** |
| private SqliteConnection connection;  public int Insert(Question question);  public Question GetById(int id);  public int DeleteById(int id);  public int Update(int questionId, Question question);  public Question[] GetAllQuestions(int id);  public List<Question> GetExportPinned(string start, string end);  public int GetImagePinned(string start, string end)  public int GetImageAll(string start, string end)  public int GetTotalPages();  public List<Question> GetPage(int pageNumber); |

## Модуль дозволяє взаємодіяти з таблицею питань. Він виконує базові CRUD операції, а також додаткові, які забезпечують імпорт, експорт, генерацію звіту і зображення, пагінацію і пошук.

Інтерфейс модуля:

|  |
| --- |
| **AnswerRepository.сs** |
| private SqliteConnection connection;  public int Insert(Answer answer);  public Answer GetById(int id);  public int DeleteById(int id);  public int Update(int answerId, Answer answer);  public Answer[] GetAllAnswers(int id);  public int GetTotalPages();  public List<Answer> GetPage(int pageNumber); |

## Модуль дозволяє взаємодіяти з таблицею питань. Він виконує базові CRUD операції, а також додаткові, які забезпечують імпорт, експорт, генерацію звіту і зображення, пагінацію і пошук.

## 3.4. TCP сервер і опис RPC протоколу

Спочатку потрібно запустити сервер. Сервер чекає на підключення і одразу ж очікує від клієнта вхідне повідомлення. Консольна програма-сервер чекає на вхідне підключення, приймає підключення, чекає на вхідні дані від клієнта, відключає клієнта. Для створення TCP сервера, використовується клас Socket. Клієнт підключається до сервера по визначеному порту, надсилає дані, приймає відповідь, відключається.

## 3.5. Модуль експорту

Модуль Export серіалізує список питань і відповідей, після чого записує їх у два різні .xml файли і зберігає в архів result.zip.

## 3.6. Модуль імпорту

Модуль Import імпортує сутності питань і відповідей з файлів до бази даних. Модуль десеріалізує .xml файли, перевіряє дані і, згідно з перевірками, зберігає дані у БД.

## 3.7. Модуль аутентифікації

Модуль Authentication має два основні методи: реєстрацію та авторизацію. Реєстрація створює нового користувача, попередньо перевіривши доступність логіна, а авторизація перевіряє коректність логіна і пароля та, на основі перевірки, дозволяє або не дозволяє вхід в програму.

## 3.8. Бібліотека ClassLibrary

Бібліотека ClassLibrary призначена для взаємодії користувача з базою даних. Бібліотека дозволяє виконує базові CRUD операції, пагінацію і пошук всіх сутностей. Також ця бібліотека надає додатковий функціонал, який забезпечує імпорт, експорт, генерацію звіту і зображення. Вона містить методи, які серіалізують і десеріалізують дані у форматі XML.

# 4. Аналіз розроблених програмних засобів

## 4.1. Особливості реалізації

Кожен користувач є модератором або звичайним користувачем. В залежності від ролі, йому доступний різний функціонал. Модератор може редагувати, додавати, переглядати і видаляти всіх користувачів, навідміну від звичайного користувача, який може редагувати, додавати і видаляти лише свої питання і відповіді.

Щоб авторизуватися, користувачу додатка потрібно ввести в поля коректний логін і пароль. При вводі пароль не показується.

Всі паролі в базі даних зберігаються в захешованому вигляді. Для хешування використовується алгоритм SHA256. Це забезпечує надійність зберігання паролів у базі даних.

Програма використовує TCP взаємодію для зв'язку клієнта і бази даних (наприклад при реєстрації та авторизації). Інший функціонал поєднує в собі клієнта і сервер (клієнт напряму звертається до бази даних). TCP взаємодія реалізується за допомогою мережевих сокетів.

Програма може обслуговувати одразу кілька клієнтів завдяки використанню багатопоточності.

Дані для експорту, імпорту, а також для передачі сутностей через сервер зберігаються в форматі XML. В мові C# є вбудована бібліотека, що дозволяє виконувати серіалізацію та десеріалізацію даних у форматі XML.

Всі користувачі можуть переглядати пагінований список всіх сутностей (користувачі, питання та відповіді), а також здійснювати пошук у кожному з вікон програми, яка забезпечує роботу з сутностями.

## 4.2. Інтерфейс користувача

Для показу даних система має інтерфейс користувача, що складається з:

1. Вікно входу в програму
2. Вікно реєстрації
3. Вікно перегляду користувачів
4. Вікно перегляду користувача
5. Вікно створення користувача
6. Вікно редагування користувача
7. Вікно перегляду питань
8. Вікно перегляду питання
9. Вікно створення питання
10. Вікно редагування питання
11. Вікно перегляду відповідей
12. Вікно перегляду відповіді
13. Вікно створення відповіді
14. Вікно редагування відповіді
15. Вікно імпорту
16. Вікно експорту
17. Вікно отримання статистики (звіту)

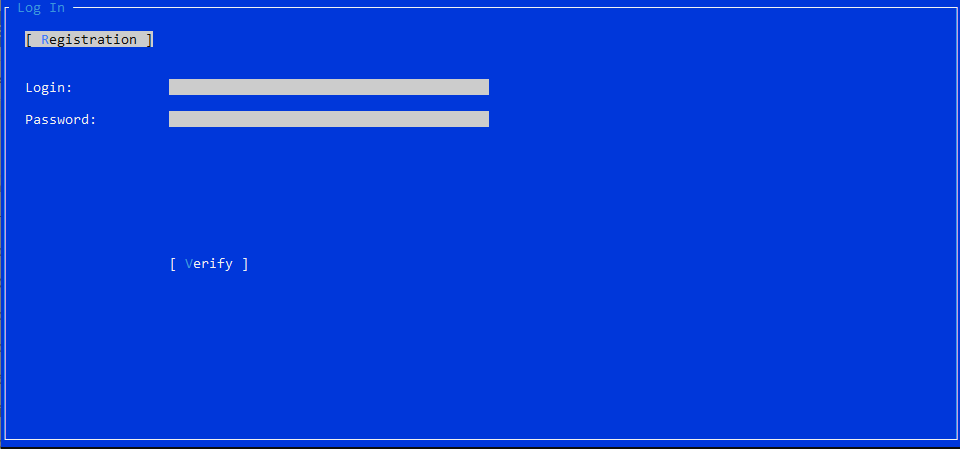


Рис. Вікно входу в програму

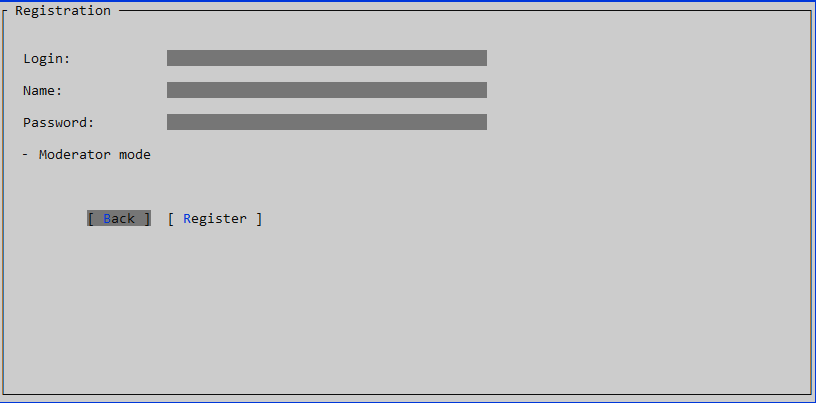


Рис. Вікно реєстрації

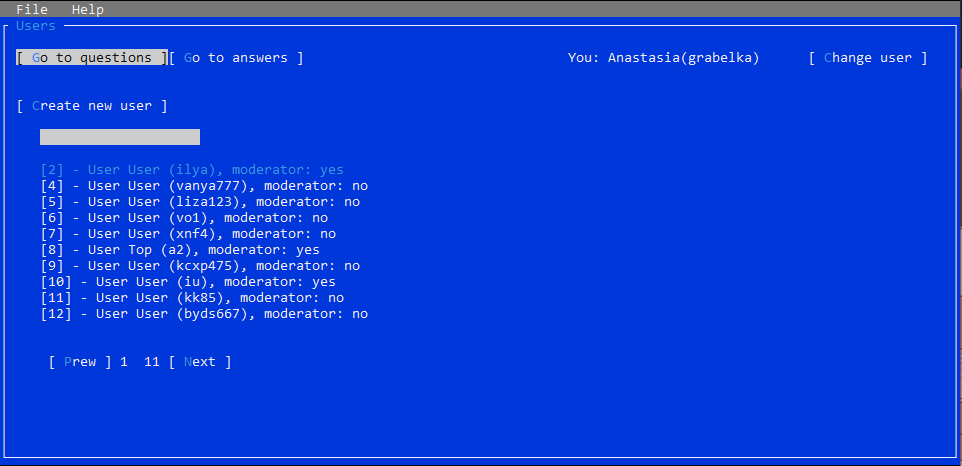


Рис. Вікно перегляду користувачів

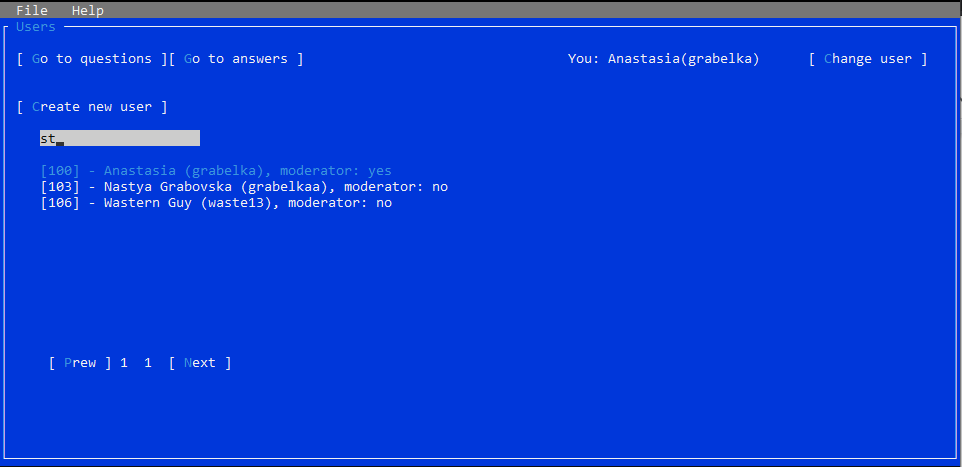


Рис. Вікно перегляду користувачів при здійсненні пошуку



Рис. Вікно створення користувача

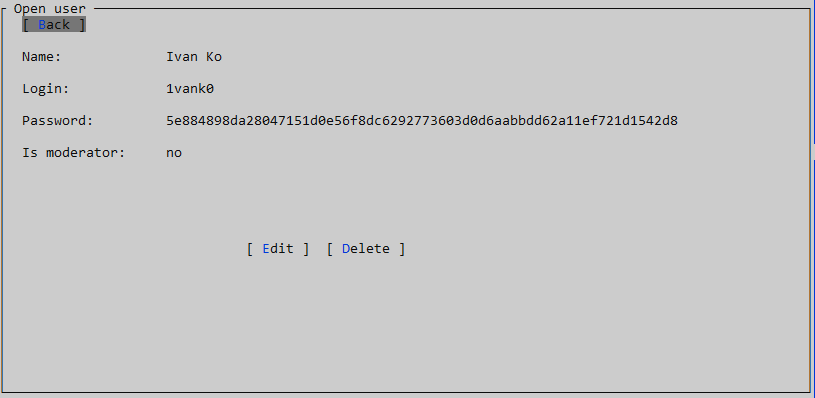


Рис. Вікно перегляду користувача

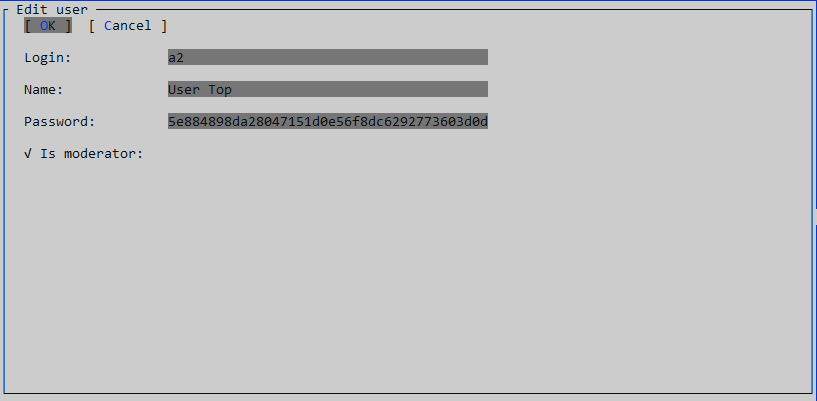


Рис. Вікно редагування користувача

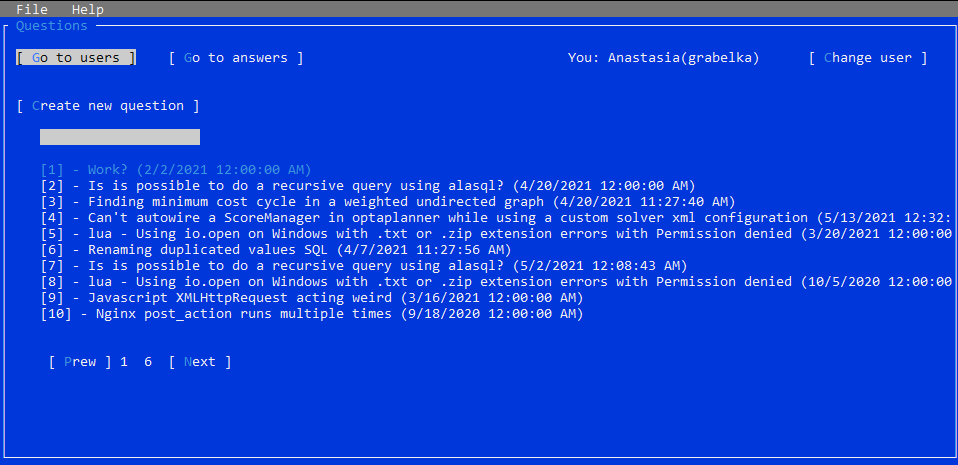


Рис. Вікно перегляду питань



Рис. Вікно перегляду питання



Рис. Вікно створення питання

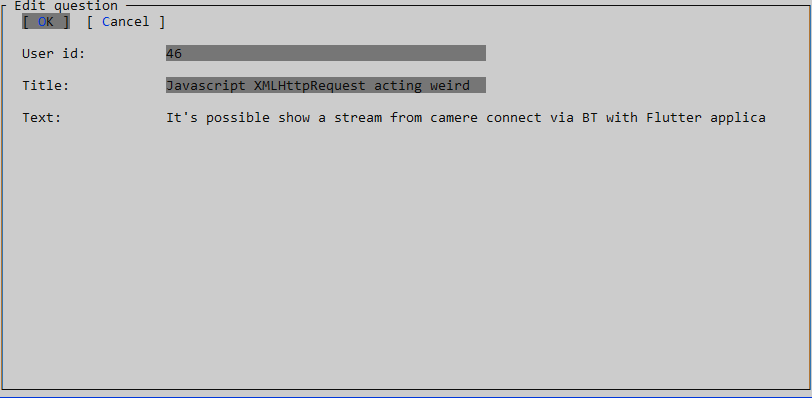


Рис. Вікно редагування питання

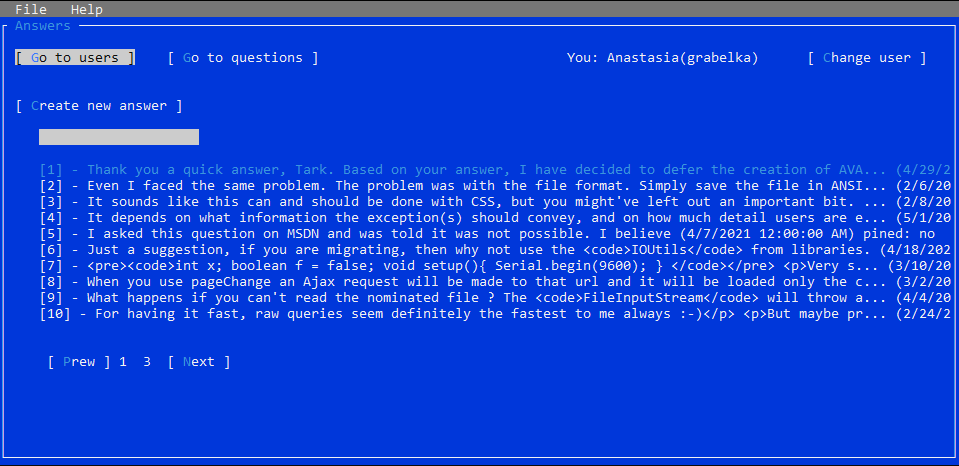


Рис. Вікно перегляду відповідей

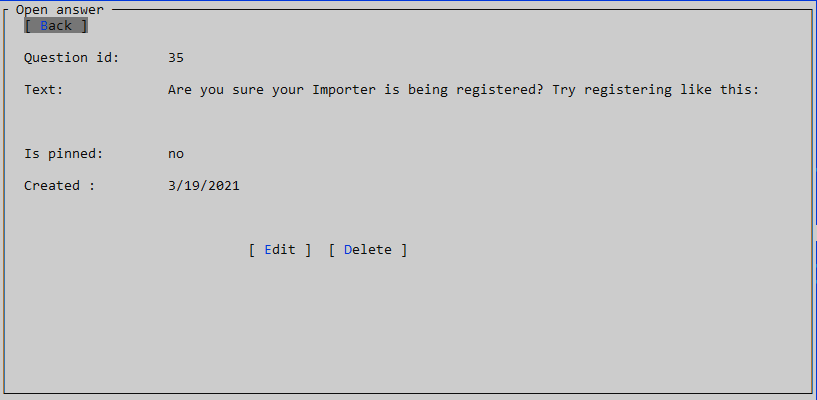


Рис. Вікно перегляду відповіді

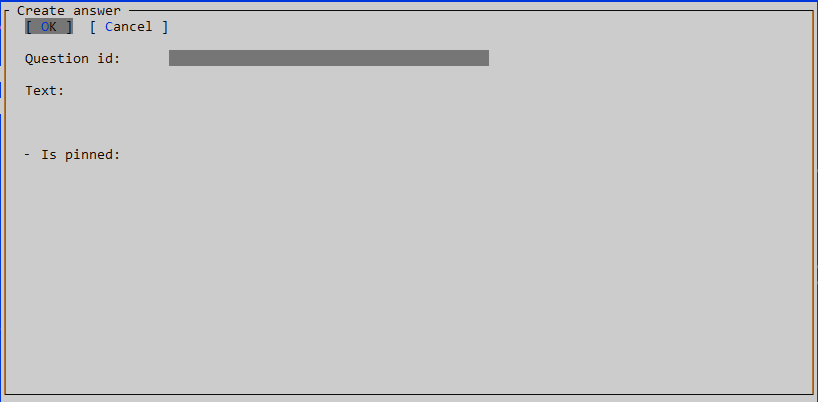


Рис. Вікно створення відповіді



Рис. Вікно редагування відповіді

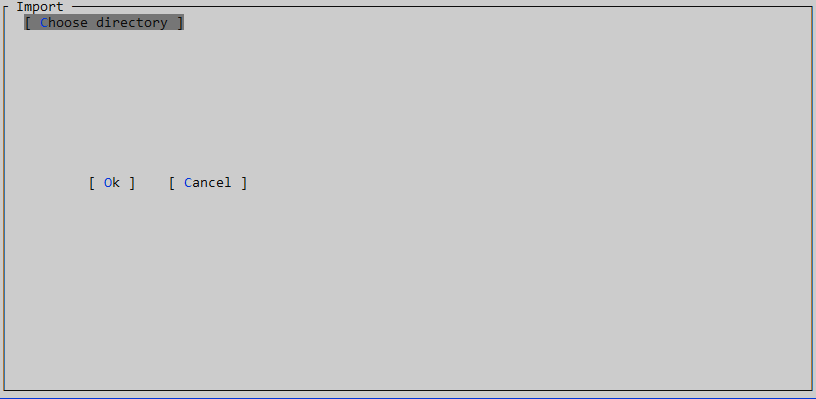


Рис. Вікно імпорту

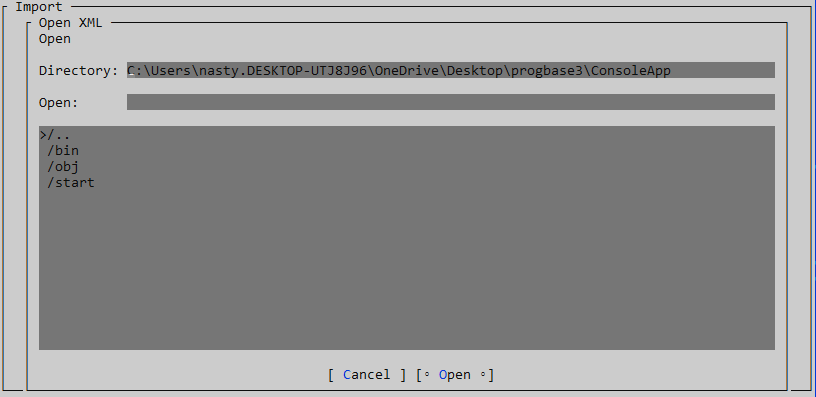


Рис. Вікно вибору директорії

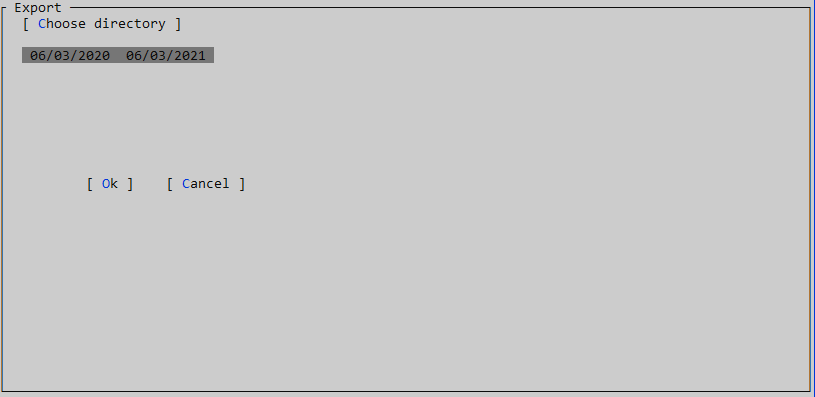


Рис. Вікно експорту

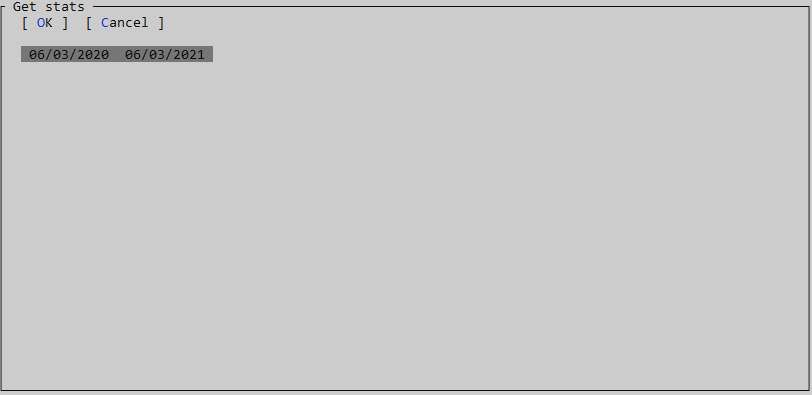


Рис. Вікно отримання статистики (звіту)

# 

# Висновки

Метою даного курсового проекту було розроблення десктопного додатку з подійно-орієнтованим інтерфейсом користувача для програмного додатку електронної бази питань і відповідей.

Розроблена система:

* забезпечує авторизований доступ до електронної бази питань та відповідей;
* дозволяє перегляд, створення, редагування та видалення питань та відповідей;
* дає можливість зареєструватися та увійти в додаток і використовувати його функціонал в залежності від ролі;

Особливу увагу під час розроблення даного програмного продукту було приділено графічному інтерфейсу користувача, взаємодії з базою даних та обробці помилок.

Використання розробленого додатку дозволить зареєструватися та увійти в додаток і використовувати його функціонал в залежності від ролі (переглядати, створювати, редагувати та видаляти питання і відповіді). Імпортувати та експортувати дані, а також отримувати статистику по питанням.