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"***

Виконав: Тітов Єгор Павлович

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 соціальної мережі**

Виконавець роботи

Тітов Єгор Павлович

\_\_\_\_\_\_\_\_\_\_\_\_\_\_2021р.

Зміст

[**Вступ**](#_heading=h.30j0zll) **4**

[**1. Аналіз предметної галузі**](#_heading=h.1fob9te) **5**

[1.1. Аналіз вимог до функціональності програмних засобів](#_heading=h.3znysh7) 5

[**2. Аналіз мов програмування та технологій розроблення**](#_heading=h.2et92p0) **6**

[2.1. Мова програмування С#](#_heading=h.tyjcwt) 6

[2.2. Фреймворк .NET](#_heading=h.3dy6vkm) 6

[2.3. База даних SQLite](#_heading=h.1t3h5sf) 6

[2.4. Бібліотека Terminal.Gui](#_heading=h.4d34og8) 7

[2.5. Бібліотека Microsoft.Data.Sqlite](#_heading=h.2s8eyo1) 7

[Надає інструменти для роботи з базою даних SQLite](#_heading=h.x83rraaucxtz) 7

[2.6. Бібліотека ScottPlot](#_heading=h.17dp8vu) 7

[**3. Опис розроблених програмних засобів**](#_heading=h.26in1rg) **7**

[3.1. Загальна структура системи](#_heading=h.lnxbz9) 7

[3.2. Структура бази даних](#_heading=h.35nkun2) 9

[3.3. Модулі репозиторіїв даних](#_heading=h.6ebjxgcq2ndt) 10

[3.5. Модуль експорту](#_heading=h.z337ya) 12

[3.6. Модуль імпорту](#_heading=h.sfthkpngzabm)12

[3.7. Модуль аутентифікації](#_heading=h.cd461uc26fnj) 12

[3.8. Бібліотека ProcessData](#_heading=h.w9zqa3s6qmg8) 12

[3.9. Бібліотека ProcessXml](#_heading=h.byb3rh854bns) 12

[**4. Аналіз розроблених програмних засобів**](#_heading=h.1y810tw) **13**

[4.1. Особливості реалізації](#_heading=h.4i7ojhp) 13

[4.2. Інтерфейс користувача](#_heading=h.2xcytpi) 13

[**Висновки**](#_heading=h.iqe103w19p15) **19**

# Вступ

В процесі обміну даними і спілкуванні між людьми часто виникає необхідність висловлення думок і суджень, обговорення певних ідей і тверджень.

Проблемами обробки такої інформації є відсутність зручної платформи для обговорень і дискусій. Виникає необхідність створення програмного додатку, який має містити базу даних висловлювань будь-яких людей, яка б могла забезпечити вільний доступ до ведення дискусій і обговорень будь-яких тем. Саме тому пропонується розробка програмних засобів електронної соціальної мережі.

Створення системи, що забезпечує вільний обмін думками, є актуальною задачею.

Даний проект присвячено розробленню електронної бази соціальної мережі, який призначено для управління даними користувачів цієї соціальної мережі.

# 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 в таблиці не є лише відображенням однієї таблиці БД, а містять склеєні чи змінені дані.
   2. 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.
7. 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
8. 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.
9. Система дозволяє генерувати зображення на основі даних, що зберігаються у БД.

# 2. Аналіз мов програмування та технологій розроблення

## 2.1. Мова програмування С#

* Використання делегатів
* Використання обробки виключень
* Використання класів
* Використання різних типів змінних
* Використання стандартних бібліотек класів

## 2.2. Фреймворк .NET

* Компіляція і запуск C# проекту
* Створення рішень (solution) проекту
* Автоматичне управління пам’яттю

## 2.3. База даних SQLite

* Створення таблиць
* Використання foreign key для побудови зв’язків one2many і one2one між таблицями
* Приведення таблиць до 1-ї і 2-ї нормальної форми
* Використання запросів для отримання і запису даних в бд.

## 2.4. Бібліотека Terminal.Gui

* Призначена для створення консольних програм з графічним інтерфейсом на основі С#. Надає багато інструментів для роботи програм на основі подійно-орієнтованого програмування.

## 2.5. Бібліотека Microsoft.Data.Sqlite

## Надає інструменти для роботи з базою даних SQLite

## 2.6. Бібліотека ScottPlot

* Надає інструменти для побудови графіків

# 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.

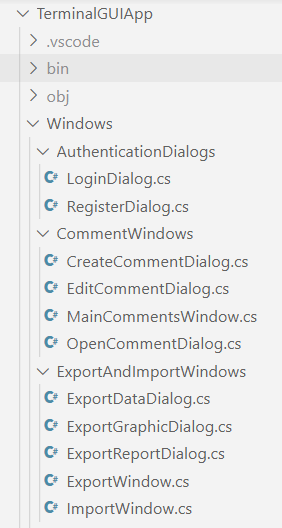
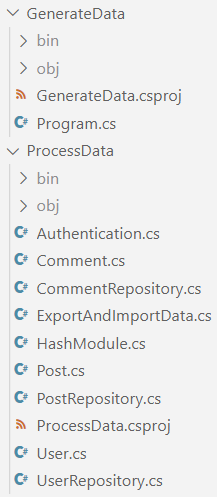
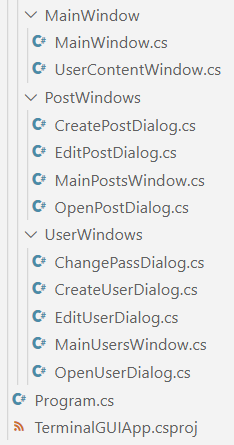


Рис. Структурна схема системи

Сервер системи обробляє запити від користувачів та виконує такі функції:

1. Реєстрація
2. Авторизація
3. Пошук і перегляд пагінованого списку всіх сутностей
4. Додавання, видалення, редагування і перегляд будь-якої сутності
5. Експорт і імпорт даних. Генерація графіку

Проект розбитий на такі модулі:



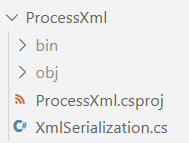
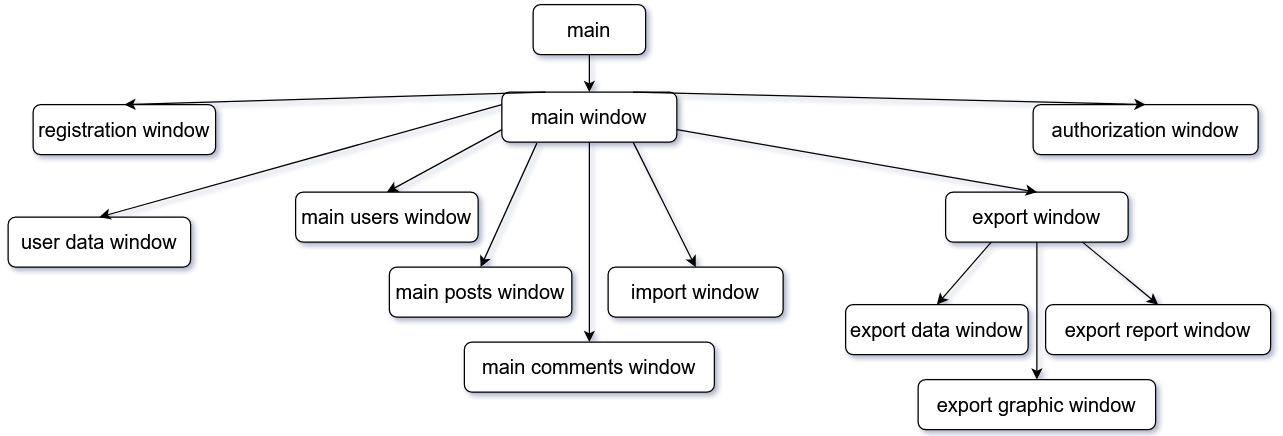


Рис. Модулі проекту

Діаграма залежностей модулів IDE:



## 3.2. Структура бази даних

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

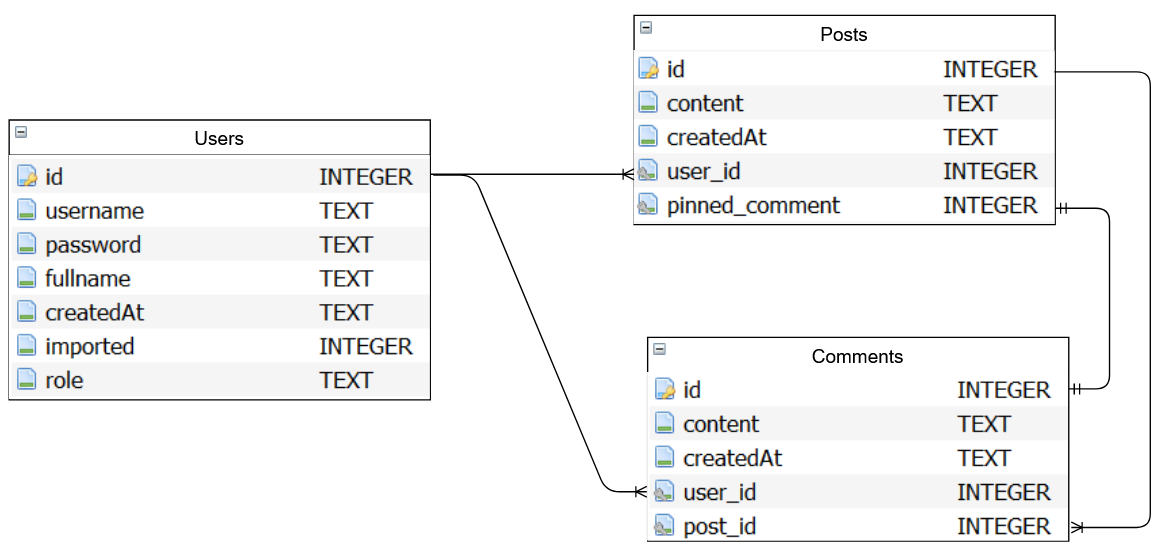


Рис. Структура бази даних

* Таблиця користувачів містить унікальний автоінкрементний id і унікальний username, за якими здійснюється пошук. Поле password містить захешований пароль для авторизації. Поля fullname, createdAt, role описують користувача. Поле imported вказує, чи даний користувач є дійсним, чи був імпортований.
* Таблиця постів містить унікальний автоінкрементний id за яким здійснюється пошук. Поля content, createdAt описують пост. Поле user\_id використовується для формування зв’язку один до багатьох між постами і користувачами. Поле pinned\_comment використовується для формування зв’язку один до одного з таблицею коментарів.
* Таблиця коментарів містить унікальний автоінкрементний id за яким здійснюється пошук. Поля content, createdAt описують коментар. Поле user\_id використовується для формування зв’язку з користувачами. Поле post\_id використовується для формування зв’язку з постами.

## 3.3. Модулі репозиторіїв даних

* Інтерфейс модуля користувачів:

|  |
| --- |
| **UserRepository.сs** |
| bool UserExists(string username)  bool UserExistsById(int id)  User[] GetAllUsers()  int[] GetAllUsersIds()  int GetSearchPagesCount(int pageSize, string searchValue)  List<User> GetSearchPage(string searchValue, int pageNum, int pageSize)  int Insert(User user)  int InsertImport(User user)  User GetByUsername(string username)  User GetById(int id)  bool Update(int userId, User user)  bool DeleteByUsername(string username) |

Модуль дозволяє взаємодіяти з таблицею користувачів для виконання CRUD операцій та операцій пошуку і пагінації.

* Інтерфейс модуля постів:

|  |
| --- |
| **PostRepository.сs** |
| int[] GetAllPostsIds()  bool PostExists(int id)  int GetSearchPagesCount(int pageSize, string searchValue)  List<Post> GetSearchPage(string searchValue, int pageNum, int pageSize)  List<Post> GetSearchUserPostsPage(int userId, string searchValue, int pageNum, int pageSize)  int GetSearchUserPostsPagesCount(int userId, int pageSize, string searchValue)  int Insert(Post post)  int InsertImport(Post post)  bool Update(int postId, Post post)  bool DeleteById(int postId)  List<Post> GetFiltredByTextPosts(string text)  List<Post> GetFiltredByTimeUserPosts(int userId, DateTime[] dateInterval)  List<Post> GetPostsByTimePeriod(DateTime[] dateInterval) |

Модуль дозволяє взаємодіяти з таблицею постів для виконання CRUD операцій та операцій пошуку і пагінації.

* Інтерфейс модуля коментарів:

|  |
| --- |
| **CommentRepository.сs** |
| int GetSearchPagesCount(int pageSize, string searchValue)  List<Comment> GetSearchPage(string searchValue, int pageNum, int pageSize)  int GetSearchUserCommentsPagesCount(int userId, int pageSize, string searchValue)  List<Comment> GetSearchUserCommentsPage(int userId, string searchValue, int pageNum, int pageSize)  List<Comment> GetFiltredByTimeUserComments(int userId, DateTime[] dateIntervals)  List<Comment> GetCommentsByTimePeriod(DateTime[] dateInterval)  int[] GetAllCommentsIds()  Comment GetById(int id)  int Insert(Comment comment)  int InsertImport(Comment comment)  bool CommentExists(int id)  List<Comment> GetByPostId(int postId)  bool Update(int commentId, Comment comment)  bool DeleteById(int commentId) |

Модуль дозволяє взаємодіяти з таблицею коментарів для виконання CRUD операцій та операцій пошуку і пагінації.

## 3.5. Модуль експорту

Модуль Export серіалізує масив постів і коментарів, після чого записує їх у два різні .xml файли і зберігає у вибраний архів.

## 3.6. Модуль імпорту

## Модуль десеріалізує .xml файли, попередньо розархівуваши їх з вибраного архіву. Після десеріалізації відбувається імпорт даних до бд.

## 3.7. Модуль аутентифікації

Містить методи реєстрації та авторизації. Реєстрація - створює нового користувача. Авторизація - на основі введеного логіну і паролю дозволяє(чи не дозволяє) вхід у систему.

## 3.8. Бібліотека ProcessData

Містить модулі: взаємодії з бд, імпорту і експорту, аутентифікації.

Дозволяє виконувати CRUD операції, операції імпорту/експорту, входу в систему, реєстрації.

## 3.9. Бібліотека ProcessXml

Містить модуль xml взаємодії. Дозволяє серіалізувати і десеріалізувати будь-які дані.

# 4. Аналіз розроблених програмних засобів

## 4.1. Особливості реалізації

Кожен користувач є або модератором, або звичайним користувачем. Звичайні користувачі можуть лише переглядати інші записи і писати свої. Модератори можуть, крім звичайних можливостей, видаляти будь-які записи. Також є роль, яку може мати лише один користувач - адміністратор. Лише адміністратор може давати ролі і видаляти користувачів.

Паролі користувачів зберігаються в хешованому вигляді.

Дані серіалізуються за допомогою вбудованої в C# бібліотеку, через додаткову бібліотеку ProcessXml, яка оптимізує і спрощує код для серіалізації.

У кожного користувача є власна сторінка (доступна лише йому), на якій користувач може побачити і редагувати всі свої пости і коментарі.

## 4.2. Інтерфейс користувача

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

1. Головне вікно
2. Вікно реєстрації
3. Вікно входу
4. Вікно перегляду користувачів
5. Вікно перегляду постів
6. Вікно перегляду коментарів
7. Вікно експорту
8. Вікно імпорту
9. Вікно перегляду всіх постів і коментарів користувача
10. Вікно перегляду даних користувача
11. Вікно створення користувача
12. Вікно редагування даних користувача
13. Вікно перегляду даних поста
14. Вікно створення поста
15. Вікно редагування даних поста
16. Вікно перегляду даних коментаря
17. Вікно створення коментаря
18. Вікно редагування коментаря
19. Вікно імпорту даних
20. Вікно експорту даних
21. Вікно експорту графіка
22. Вікно експорту звіту

Знімки вікон

|  |  |
| --- | --- |
|  |  |
| Рис. Головне вікно | Рис. Вікно реєстрації |

|  |  |
| --- | --- |
|  |  |
| Рис. Вікно входу | Рис. Вікно перегляду користувачів |
|  |  |
| Рис. Вікно перегляду постів | Рис. Вікно перегляду коментарів |
|  |  |
| Рис. Вікно експорту | Рис. Вікно імпорту |

|  |  |
| --- | --- |
|  |  |
| Рис. Вікно перегляду всіх постів і коментарів користувача | Рис. Вікно перегляду даних користувача |
|  |  |
| Рис. Вікно створення користувача | Рис. Вікно редагування даних користувача |
|  |  |
| Рис. Вікно перегляду даних поста | Рис. Вікно створення поста |

|  |  |
| --- | --- |
|  |  |
| Рис. Вікно редагування даних поста | Рис. Вікно перегляду даних коментаря |
|  |  |
| Рис. Вікно створення коментаря | Рис. Вікно редагування коментаря |
|  |  |
| Рис. Вікно імпорту даних | Рис. Вікно експорту даних |

|  |  |
| --- | --- |
|  |  |
| Рис. Вікно експорту графіка | Рис. Вікно експорту звіту |

# Висновки

Метою даного курсового проекту було розроблення десктопного додатку з подійно-орієнтованим інтерфейсом користувача для програмного додатку електронної бази соціальної мережі.

Розроблена система:

* забезпечує авторизований доступ до електронної бази соціальної мережі;
* дозволяє перегляд, створення, видалення, редагування користувачів, постів, коментарів;
* дає можливість зареєструватися та авторизуватися в систему для використання всіх можливостей додатку;

Особливу увагу під час розроблення даного програмного продукту було приділено графічному інтерфейсу, обробці введення можливих некоректних даних користувачем та коректній взаємодії бд з інтерфейсом користувача.

Розробка виконана у повному обсязі, всі вимоги враховані, продукт протестовано.

Використання розробленого додатку дозволить вільно ділитися думками з іншими користувачами, спілкуватися, обговорювати різноманітні теми, ідеї, судження.