МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ

ДЕРЖАВНИЙ УНІВЕРСИТЕТ «ЖИТОМИРСЬКА ПОЛІТЕХНІКА»

Кафедра комп’ютерних наук

**КУРСОВА РОБОТА**

з дисципліни: «Програмування»

на тему:

**«Розробка програми імітації онлайн-месенджера ''Skype''»**

студента І курсу групи ІСТ-23-1

спеціальності 126 «Інформаційні системи та технології»

Ячменьової Софії Олександрівни

(прізвище, ім’я та по-батькові)

Керівник ассистент кафедри КН

Оринчак А.І

Члени комісії: Віталій ЛЕВКІВСЬКИЙ \_

Галина МАРЧУК .

Андрій ОРИНЧАК \_

Житомир – 2024

ДЕРЖАВНИЙ УНІВЕРСИТЕТ «ЖИТОМИРСЬКА ПОЛІТЕХНІКА»

Факультет інформаційно-комп’ютерних технологій

Кафедра комп’ютерних наук

Освітній рівень: бакалавр

Спеціальність 126 «Інформаційні системи та технології»

«ЗАТВЕРДЖУЮ»

Завідувач кафедри комп’ютерних наук

Марина ГРАФ

3 квітня 2024 р.

ЗАВДАННЯ

НА КУРСОВУ РОБОТУ СТУДЕНТЦІ

Ячменьовій Софії Олександрівні

1. Тема роботи: «Розробка програми імітації онлайн месенджера "Skype"»,

керівник роботи: ассистент кафедри комп’ютерних наук, Оринчак Андрій Іванович

1. Строк подання студентом: “ 29 ” травня 2024р.
2. Вихідні дані до роботи: розробити програму імітації онлайн месенджера "Skype" - моделювання платформи для спілкування.
3. Зміст розрахунково-пояснювальної записки(перелік питань які підлягають розробці)
   * + 1. Постановка завдання
       2. Аналіз аналогічних розробок
       3. Алгоритми роботи програми

4. Опис роботи програми

5. Програмне дослідження

1. Перелік графічного матеріалу(з точним зазначенням обов’язкових креслень)

1. Презентація до курсової роботи\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ 2. Посилання на репозиторій: \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

1. Консультанти розділів роботи

| Розділ | Прізвище, ініціали та посади консультанта | Підпис, дата | |
| --- | --- | --- | --- |
| завдання  видав | завдання прийняв |
| 1,2 | Левківський Віталій, доцент кафедри комп’ютерних наук |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |

1. Дата видачі завдання “ 3 ” квітня 2024 р.

| № з/п | Назва етапів курсової роботи | Строк виконання етапів роботи | Примітки |
| --- | --- | --- | --- |
| 1 | Постановка задачі | 03.04.2024 | Виконано |
| 2 | Пошук, огляд та аналіз аналогічних розробок | 05.04.2024 | Виконано |
| 3 | Формулювання технічного завдання | 07.04.2024 | Виконано |
| 4 | Опрацювання літературних джерел | 10.04.2024 | Виконано |
| 5 | Проектування структури | 15.04.2024 | Виконано |
| 6 | Написання програмного коду | 30.04.2024 | Виконано |
| 7 | Відлагодження | 10.05.2024 | Виконано |
| 8 | Написання пояснювальної записки | 23.05.2024 | Виконано |
| 9 | Захист | 29.05.2024 |  |

**КАЛЕНДАРНИЙ ПЛАН**

**Студентка** Софія ЯЧМЕНЬОВА

**Керівник проекту** Андрій ОРИНЧАК

# РЕФЕРАТ

У першому розділі представлено результати аналізу методів та засобів реалізації програми імітації месенджера. На основі аналізу було виявлено принципи та особливості розробки застосунку чат-системи.

ЗМІСТ

ВСТУП ...……………………………………………………………………………6

РОЗДІЛ 1. АНАЛІЗ ПРОБЛЕМАТИКИ, МЕТОДІВ ТА ЗАСОБІВ ВИРІШЕННЯ ЗАДАЧІ ……………………………………………………………………………..7

1.1. Аналіз задачі, засобів та методів її вирішення …........................................7

1.2. Аналіз існуючого програмного забезпечення за тематикою курсового проекту…………………………………………………………………………….8

1.3. Постановка завдання на курсовий проект…...............................................10

РОЗДІЛ 2. ПРОЕКТУВАННЯ ТА РОЗРОБКА ПРОГРАМНОГО ЗАБЕЗПЕЧЕННЯ…………………………………………………………………...12

2.1. Проектування загального алгоритму роботи програми…..........................12

2.2. Розробка функціональних алгоритмів роботи програми…........................15

2.3. Розробка програмного забезпечення ….......................................................17

РОЗДІЛ 3. ОПИС РОБОТИ З ПРОГРАМНИМ ДОДАТКОМ ТА ЙОГО ТЕСТУВАННЯ

3.1. Опис роботи з програмним додатком

3.2. Тестування роботи програмного забезпечення

ВИСНОВКИ

СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ

ДОДАТКИ

# ВСТУП

Мова програмування С#, в даний час, є одним з найпопулярніших

інструментів написання програмного забезпечення, з широким вибором технологій розробки. Була створена Андерсом Гейлсбергом, Скотом Вілтамутом та Пітером Гольде під егідою Microsoft Research у 2000 році.

**Актуальність.** На відміну від своїх попередників C# має можливість легшої взаємодії, з кодом програм, написаних на інших мовах, що є важливим при створенні великих проєктів та є необхідним у сучасних умовах розробки.

**Метою курсового проекту** є закріплення та отримання теоритичних знань, професійних навиків з програмування мовою С# та розробки програмного забезпечення комп’ютерних технологій на основі інтегрованого середовища Visual Studio.

**Завданням курсового проекту** є створення програмного додатку імітації онлайн-месенджера «Skype», що реалізує обробку даних з використанням Windows Forms та Windows Presentation Forms. Основний фукціонал додатку полягає в базі адресантів та адресатів, можливості переписок і спілкування.

**Об’єктом дослідження** даної практики є програмування мовою С#.

**Предмет дослідження** - основні принципи та способи стоворення візуальних додатків мовою С#.

# РОЗДІЛ 1. АНАЛІЗ ПРОБЛЕМАТИКИ, МЕТОДІВ ТА ЗАСОБІВ ВИРІШЕННЯ ЗАДАЧІ

## 1.1 Аналіз задачі, засобів та методів її вирішення

На початковому етапі проектування програми імітації месенджера потрібно провести попереднє дослідження та аналіз. Насамперед, потрібно переглянути інформацію про представлення чат-систем та існуючих аналогів.

Головна задача будь-якого месенджера - це швидкий обмін інформацією у вигляді повідомлень, що можуть бути представлені у формі: графічних данних, відео та аудіо записів, текстових наборів та файлів різного розширення.

Для створення зручного додатку необхідно реалізувати надійну систему обміну інформацією та комфортний у користуванні інтерфейс. Користувач має мати можливість входу до системи, вибору чату, обміну сповіщеннями, пошук повідомлень або контактів.

Система обміну інформації має бути подана у вигляді чату, де є зазначені користувачем контакти. У середині переписки має бути представлений функціонал.

Для того, щоб користувач знайшов необхідний контакт чи повідомлення потрібно реалізувати інформаційно-пошукову систему. Це має бути список з можливістю переходу до повідомлення чи чату.

Розробка імітації месенджера вимагає технічних знань, а саме: розуміння графічного інтерфейсу, логіки програми та потреб користувача.

Для розробки програми імітації месенджера буде використана мова C#, орієнтована для роботи з платформою .NET, об'єктно-орієнтованим підходом,, мультиплатформеністю та інтеграцією для Microsoft, що робить її популярним вибором для розробки різних типів програм.

Як середовище розробки буде використано технології Windows Forms та Windows Presentation Forms, що є популярними фреймворками для розробки програмного забезпечення. Windows Forms та Windows Presentation Forms містять набір основних графічних елементів: кнопки, тексти, списки, таблиці, панелі, вікна, тощо - що надає можливість розробнику зробити зручний та зрозумілий інтерфейс для користувача.

1.2 Аналіз існуючого програмного забезпечення за тематикою курсового проекту

Існує різноманітна кількість реалізацій поданого застосунка. Тому, сучасні месенджери можна поділити на такі групи:

* - Інтегровані в соціальну мережу або в іншу програму. Мають значний функціонал з підтримкою всіх стандартних можливостей, в тому числі обмін файлами та набір власних смайлів. Для посилання та отримання повідомлень не потрібне попереднє звантаження;
* - Самостійні програми. Щоб почати користуватися таким месенджером його необхідно завантажити та встановити на свій пристрій. Можуть мати додаткові унікальні функції, відеозв'язок, голосові дзвінки, але це необов'язково. Наприклад, WhatsApp, призначений виключно для спілкування в текстовому форматі;



Рис. 1.1 - Франмент з WhatsUp

* - Інтегровані частково. Мають функцію «перехоплення» списку контактів з інших сервісів або соціальних мереж. Також вимагають попереднього звантаження. Найбільш поширеним прикладом частково інтегрованою програми є Facebook Messenger;



Рис. 1.2 - Франмент з Facebook Messenge

* - Мобільні додатки. Доступні для використання в мобільних пристроях. Популярність виключно таких месенджерів поступово згасає, бо основний ухил робиться на мультиплатформеність. Нерідкі випадки, коли такі програми надалі адаптують для роботи в інших операційних системах, що дозволяє значно розширити кількість користувачів;
* - Призначені для установки на будь-який пристрій під управлінням різних ОС, в тому числі на персональні комп'ютери. Як правило, такі месенджери є інтегрованими, але поступово до їх списку стали відноситися Telegram, Viber, Skype та інші.[1]



Рис. 1.3 - Франмент з Skype

Незважаючи на велику кількість видів та модифікацій месенджерів, можна описати спільні риси роботи застосунків. Робота будь-якого додатку переписок розпочинається з створення аккаунта для обміну повідомленнями між зареєстрованими користувачами. Для реєстрації програма потребує загальну інформацію про особу, щоб забезпечити унікальний індетифікатор для пересилання повідомлень зазначеному користовачу.

Далі сервіс відкриває для користувача панель чатів, де присутні усі соціальні групи та переписки. Месенджер дозволяє обрати необхідні з них та почати обмін інформацією: картинками, відео, смайлами, текстовими повідомленнями, звуковими записами, тощо.

Важливо зазначити, що будь-який користувач може редагувати свій список переписок, що складається з обраних контактів, у панелі: додавати нових співбесідників, видаляти непотрібні чати, тощо.

Подані месенджери допомагають користувачам спілкуватися швидко та зручно. При створенні своєї власної системи можна врахувати їхні функції та особливості для створення зручного та цінного інструменту для споживачів.

1.3 Постановка завдання на курсовий проект

Технічне завдання для програми імітаціє месенджера є створення таких частин:

1. Головного вікна з чат-системами;

2. Вікна входу в аккаунт;

3. Реєстраційної форми;

4. Панелей налаштування;

5. Внутрішніх компонентів роботи з чат-системою (кнопки, списки, тощо)

Головне вікно міститеме елементи керування чатами та необхідну область для обміну повідомленнями. При взаємодії з елементами головного вікна можна буде викликати вікно зміни аккаунта або панель налаштувань.

Форма входу міститиме поля для введення логіну і пароля або викликатиме форму реєстрації. Панель налаштувань дозволятиме організовувати відбраження головної області та зміну параметрів роботи застосунку.

Висновок до першого розділу

Перед проектуванням та реалізацією програми імітації месенджера необхідно провести попереднє дослідження існучих аналогів застосунку, сформувати технічне завдання і обрати технологію реалізації. Для створення чат-системи було обрано Windows Presentations Forms для прототипу і Windows Forms для реліз-версії застосунку, що мають необхідний для виконання задачі функціонал.

# РОЗДІЛ 2. ПРОЕКТУВАННЯ ТА РОЗРОБКА ПРОГРАМНОГО ЗАБЕЗПЕЧЕННЯ

## 2.1 Проектування загального алгоритму роботи програми

Продовжуючи етап проектування програми імітації месенджера потрібно визначити методи реалізації, структуру програми, оцінити обсяг функціонального наповнення програми. Використовуючи поставлені вище умови технічної задачі є необхідним узгодження і опис процесів роботи застосунку. Насамперед, спроектуємо діаграму сценаріїв використання, що узгодить функціональне наповнення, показана на рисунку 2.1.



Рис. 2.1 - Діаграма сценаріїв використання

Процес роботи програми імітації онлайн-месенджеру складається з взаємодіє декількох користувачів, що мають за мету передавання інформації в текстовому або файловому вигляді. Користувачі створюють чат, де надсилають один одному повідомлення, редагують їх, видаляють, переглядають, тощо. У кожного користувача може бути безліч таких чатів, які він може додавати і видаляти.

Кожна дія користувача є дією над певним об’єктом: повідомленням, чатом, профілем, тощо. Важливим є пропрацювання реалізації цих об’єктів, що у програмуванні відповідаючи створенню необхідних класів, методів, інтерфейсів. Прийнятним способом проектування для вище поставленоє задачі слугує діаграма класів, наведена на рисунку 2.2.

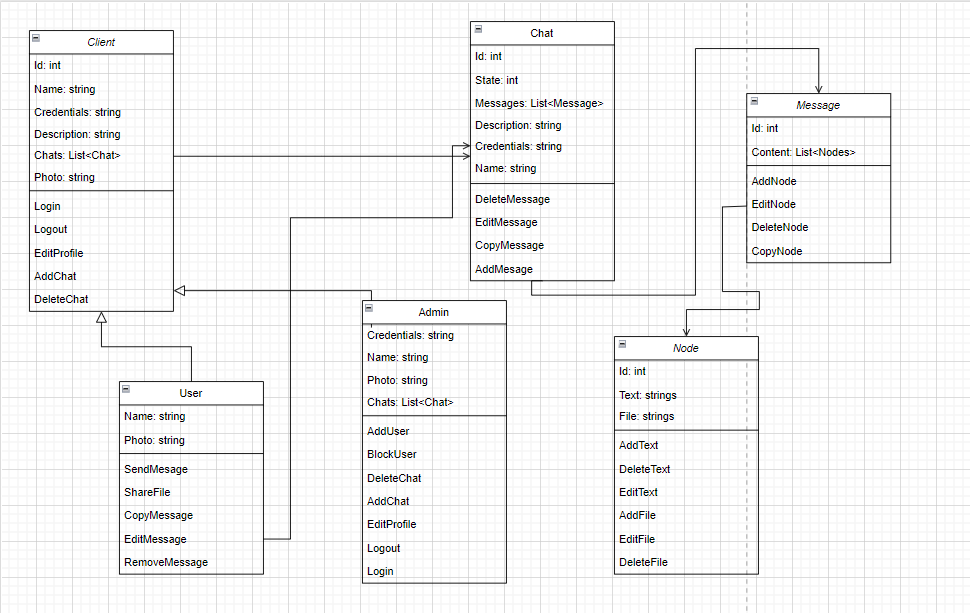


Рис. 2.2 - Діаграма класів

Основними класами програми є Client та Сhat, що реалізують задачу створення користувачів та чатів, які є основою усіх процесів програми. Похідні класи від клієнта - це Admin та User, відмінність яких у правах доступу та можливостях. Admin контролює стан чатів та поведінку користувачів, тому може їх видаляти чи додавати. Клас User відповідає за діє клієнтів у програмі, тобто надає їм можливість обміну повідомленнями. Процес обміну супроводжується змінами полів у класі Chat, що містить набір повідомлень. Повідомлення реалізуються через клас Messages, що має список типу Node - відповідає за текст повідомлення та файл надісланий користувачем.

Подані вище класи - це моделі програми імітації онлайн-месенджера «Skype», що потребують додаткової візуальзації та функціонального представлення. Описаний спосіб реалізації відповідає патерну проектування Модель - представлення - контролер (MVC, [англ.](https://uk.wikipedia.org/wiki/Англійська_мова) Model View Controller).

Модель - представлення - контролер передбачає поділ системи на три взаємопов'язані частини: [модель даних](https://uk.wikipedia.org/wiki/Модель_даних), вигляд ([інтерфейс користувача](https://uk.wikipedia.org/wiki/Інтерфейс_користувача)) та модуль керування. Застосовується для розділення даних (моделі) від зовнішнього вигляду додатку - інтерфейсу користувача (подання) так, щоб зміни подання користувача мінімально впливали на роботу з даними, а зміни в моделі даних могли здійснюватися без змін інтерфейсу користувача.[2]

Визначивши моделі для роботи необхідно створити їх представлення. Подання даних у програмі імітації месенджеру «Skype» відбувається за допомогою створення та взаємодії форм, рисунок 2.3.

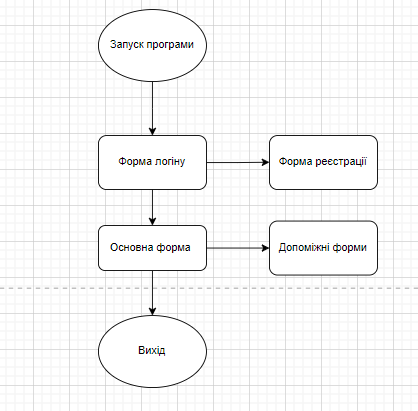


Рис. 2.3 - Схема форм застосунку

Першою формою, з якої користувач починає роботу, є форма логіну. Тобто, будь-який клієнт месенджеру має мати аккаунт для входу, щоб почати обмін сповіщеннями. Якщо такого аккаунту не має, то користувач створює аккаунт за допомогою форми реєстрації. Опісля, клієнт виконує усі діє над чатами, тобто спілкування в головному вікні. Якщо користувач редагує профіль, повідомлення, чи налаштування вікна, тощо, то викликається відповідна допоміжна форма. У випадку припеннення роботи клієнт виходить з головного вікна, закриваючи програму і припиняючи її роботу.

Головне вікно програми месенджера - це панель, де можна переглянути чати, відкрити їх, додати чи видалити. Ця панель має містити сектори меню та налаштувань роботи, область передання інфомації, де буде відображатися вибраний чат, профіль користувача, тощо.

Робота з головною формою - це основна функціональна частина застосунку, яка може бути описана у вигляді алгоритму дій - блок-схеми. Реалізуючи відповідну блок-схему необхідно врахувати усі можливі ситуації при роботі. Описаний нижче алгоритм є стислим відображенням дій з головною фомою, рисунок 2.4.

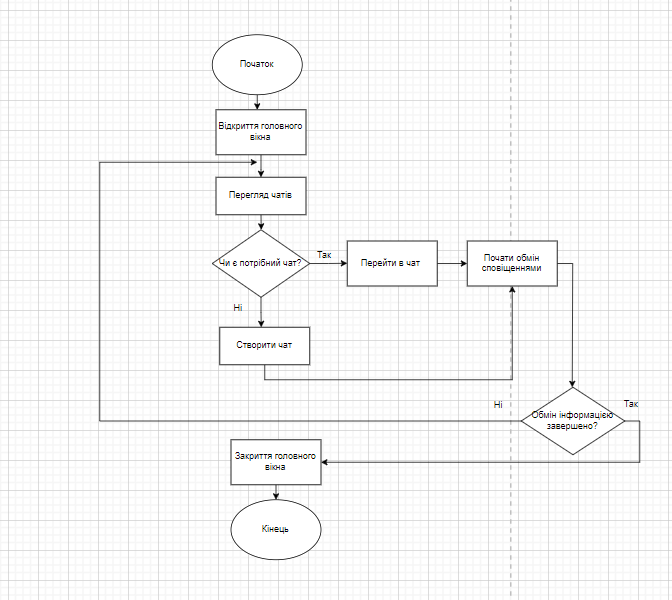


Рис. 2.4 - Блок-схема головної форми

Усі діє в програмі над будь-якою формою керуються необхідними контролерами, що представлені як методи, допоміжні класи та інтрфейси. Вони будуть реалізувати способи доступу до моделей.

2.2 Розробка функціональних алгоритмів роботи програми

Коли описано загальний алгоритм роботи програми, можна проаналізувати всі компоненти головної форми окремо, а також розглянути допоміжні класи і дії над ними.

Основна форма сладається з двох елементів Panel, одного компонента Listbox, чотирьох об’єктів ContextMenuStrip. Панелі використано для розміщення списку розмов та відображення вибраного користувачем чату, а також чотирьох контексних меню для налашування роботи з вікном.

У програмі головна форма має передавати данні різних користувачів під час обміну сповіщеннями. Тому, елемент Panel, що містить Listbox з набором чатів є дуже важливим. Чати повинні бути опрацьовані певним контролером, який зможе відлагодити їх взаємодію з користувачами. У цьому випадку доречним є використання поділення на серверну частини і клієнтську.

Серверна частина має відповідати за зв’язок клієнтів, тобто обмін сповіщеннями. Для реалізації цієї задачі доречним є використання фреймворку SignaIR, який дає можливість запуску серверу та його роботи.

SignaIR працює за допомогою TCP-протоколу, створює з'єднання між відправником і одержувачем (рисунок 2.5), а потім дані передаються пакетами. TCP також гарантує, що всі пакети доставляються в порядку. Крім того, він має механізми для повторної передачі втрачених пакетів і керування потоком.[3]



Рис. 2.5 - Схема роботи TCP-пакетів

Клієнтську частину з сервером буде поєднувати клас ChatHub, що використовуватиме інтерфейси: IClient та ICommand; і буде містити методи опрацювання запитів до сервера і надання доступу до даних у візуальній частині - формах. IClient відповідатиме за опрацювання дій клієнтів: підключення до сервера, надсилання повідомлень і зображень, тощо. ICommand реалізовуватиме синхронні та асинхронні команди, які міститеме IClient.

2.3 Розробка програмного забезпечення

Технічне завдання для програми імітаціє месенджера є створення таких частин:

Висновок до другого розділу

Перед проектуванням та реалізацією програми імітації месенджера необхідно провести попереднє дослідження існучих аналогів застосунку.

# СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ

1. Яким месенджерам віддають перевагу українці [Електронний ресурс] – 2022. – Режим доступу до ресурсу: <http://icoola.ua/blog/populiarni-mesendzhery-v-ukraini/>
2. Martin Fowler UML Distilled: A Brief Guide to the Standard Object Modeling Language. Third Edition. Addison-Wesley Object Technology Series, 2005 – 118 p.
3. Протоколи TCP та UDP - пояснення простою мовою [Електронний ресурс] – 2021. – Режим доступу до ресурсу: https://devzone.org.ua/post/protokoly-tcp-ta-udp-poiasnennia-prostoiu-movoiu

# ДОДАТКИ

# ДОДАТОК А

Лістинг програмного коду