

РЕФЕРАТ

Пояснювальна записка: ___ с., ___ рис., ___ табл., ___ дод., ___ джерел.

Об'єкт розробки: інформаційна система обробки архівних повідомлень.

Мета кваліфікаційної роботи: автоматизація процесу збору інформації про роботу банку, формування звітів, формування необхідних документів і звітів для проведення розслідувань та інших операцій з цінними паперами.

У вступі розглядається аналіз та сучасний стан проблеми, конкретизується мета кваліфікаційної роботи та галузь її застосування, наведено обґрунтування актуальності теми та уточнюється постановка завдання.

У першому розділі проведено аналіз предметної галузі, визначено актуальність завдання та призначення розробки, розроблена постановка завдання, задані вимоги до програмної реалізації, технологій та програмних засобів.

У другому розділі виконано аналіз існуючих рішень, обрано вибір платформи для розробки, виконано проектування і розробка програми, наведено опис алгоритму і структури функціонування програми, визначені вхідні і вихідні дані, наведені характеристики складу параметрів технічних засобів, описаний виклик та завантаження програми, описана робота програми.

В економічному розділі визначено трудомісткість розробленої інформаційної системи, проведений підрахунок вартості роботи по створенню програми та розраховано час на його створення.

Практичне значення роботи полягає в формуванні коректного повідомлення з архівної бази по заданому оператором критерієм пошуку. При цьому програма Messenger працює з усіма типами фінансових повідомлень оброблюваних банком.

Актуальність кваліфікаційної роботи визначається тим, що в ній реалізована можливість конкретизації пошуку повідомлень по їх типам, датою отримання/відправки, напрямку прямування, а також за сумою платежу, датою валютування з позначенням як відправника, так і одержувача повідомлення.

Список ключових слів: SWIF, ПОВІДОМЛЕННЯ, АРХІВ ДАНИХ, ІНФОРМАЦІЙНА СИСТЕМА, БАНК, АНАЛІЗ ДАНИХ.

ABSTRACT

Explanatory note: ___ pp., ___ fig., ___ table, __ appendix, ___ sources.

Object of development: information system for processing archival messages.

The purpose of the qualification work: automation of the process of collecting information about the bank, the formation of reports, the formation of the necessary documents and reports for investigations and other transactions with securities.

The introduction considers the analysis and the current state of the problem, specifies the purpose of the qualification work and the scope of its application, provides a justification for the relevance of the topic and clarifies the problem.

In the first section the analysis of the subject area is carried out, the urgency of the task and purpose of development are defined, the statement of the task is developed, requirements to software realization, technologies and software are set.

The second section analyzes the existing solutions, selects the platform for development, performs design and development of the program, describes the algorithm and structure of the program, determines the input and output data, provides characteristics of the parameters of hardware, describes the call and download program, describes the program .

The economic section determines the complexity of the developed information system, calculates the cost of work to create a program and calculates the time for its creation.

The practical significance of the work is to form a correct message from the archive database according to the search criteria specified by the operator. The Messenger program works with all types of financial messages processed by the bank.

The relevance of the qualification work is determined by the fact that it implements the ability to specify the search for messages by their types, date of receipt / sending, direction, as well as the amount of payment, value date with both sender and recipient of the message.

List of keywords: SWIF, MESSAGES, DATA ARCHIVE, INFORMATION SYSTEM, BANK, DATA ANALYSIS.

ПЕРЕЛІК УМОВНИХ ПОЗНАЧЕНЬ

- БД – база даних;
- ЕОД – електронний обмін даними;
- ІС – інформаційна система;
- ООП – об’єктно-орієнтоване програмування;
- ОС – операційна система;
- ПЗ – програмне забезпечення;
- ПК – персональний комп’ютер;
- СУБД – система управління базою даних;
- ВІС – Bank Identifier Codes, код ідентифікатора банку;
- МТ – Message Transaction, транзакція повідомлення;
- SWIFT – Society for Worldwide Interbank Financial Telecommunication,
товариство всесвітніх банківських комунікацій.

ЗМІСТ

РЕФЕРАТ.....	3
ABSTRACT.....	4
СПИСОК УМОВНИХ ПОЗНАЧЕНЬ.....	5
ВСТУП.....	8
РОЗДІЛ 1. АНАЛІЗ ПРЕДМЕТНОЇ ГАЛУЗІ ТА ПОСТАНОВКА ЗАВДАННЯ.....	10
1.1. Загальні відомості з предметної галузі	10
1.1.1. SWIFT - суспільство всесвітніх банківських комунікацій.....	10
1.1.2. Інтерфейс TurboSWIFT	12
1.1.3. SWIFT повідомлення	13
1.1.4. Передача повідомлень в форматі SWIFT через Інтернет.....	17
1.2. Призначення розробки та галузь застосування.....	18
1.3. Підстава для розробки.....	18
1.4. Постановка завдання.....	19
1.5. Вимоги до програми або програмного виробу.....	20
1.5.1. Вимоги до функціональних характеристик.....	20
1.5.2. Вимоги до інформаційної безпеки.....	20
1.5.3. Вимоги до складу та параметрів технічних засобів.....	21
1.5.4. Вимоги до інформаційної та програмної сумісності	22
РОЗДІЛ 2. ПРОЄКТУВАННЯ ТА РОЗРОБКА ІНФОРМАЦІЙНОЇ СИСТЕМИ.....	23
2.1. Функціональне призначення системи	23
2.2. Опис застосованих математичних методів.....	23
2.3. Опис використаних технологій та мов програмування.....	23
2.4. Опис структури програми та алгоритмів її функціонування ...	31
2.4.1. Проектування БД ІС.....	31
2.4.1.1.Опис логічної структури бази даних.....	31

2.4.1.2.Опис фізичної структури бази даних.....	33
2.4.1.3.Програмування БД системи.....	34
2.4.2. Структура системи.....	40
2.4.3. Взаємодія з іншими компонентами ПЗ банку.....	41
2.5. Обґрунтування та організація вхідних та вихідних даних програми.....	42
2.6. Опис розробленої системи	43
2.6.1. Використані технічні засоби.....	43
2.6.2. Використані програмні засоби.....	44
2.6.3. Виклик та завантаження програми.....	44
2.6.4. Опис інтерфейсу користувача.....	44
РОЗДІЛ 3. ЕКОНОМІЧНИЙ РОЗДІЛ.....	51
3.1. Розрахунок трудомісткості та вартості розробки програмного продукту.....	51
3.2. Розрахунок витрат на створення програми.....	54
ВИСНОВКИ.....	56
СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ.....	58
Додаток А. Код програми.....	61
Додаток Б. Відгук керівника економічного розділу.....	87
Додаток В. Перелік файлів на диску.....	88

ВСТУП

Одне з основних напрямків діяльності суспільства полягає в розробці уніфікованих засобів обміну фінансовою інформацією. З цією метою створена і продовжує удосконалюватися структурована система фінансових повідомлень, за допомогою якої можливо здійснити практично весь спектр банківських та інших фінансових операцій, включаючи операції, що здійснюються на валютних і фондових біржах.

Формати стандартизованих машинопечатних повідомлень розроблені таким чином, щоб зробити їх найбільш незалежними від національних особливостей банківської сфери в кожній конкретній країні. У той же час уніфіковані формати повідомлень, що використовуються для передачі інформації в мережі SWIFT, поряд з присвоюваними суспільством банківськими ідентифікаційними кодами (восьмизначний код, який є унікальною адресою банківських та інших фінансових інститутів), рекомендовані Міжнародною організацією зі стандартизації (ISO) в якості міжнародних стандартів. Стандарти SWIFT стали стандартами «де факто» для фінансових повідомлень транзакцій, роблячи усе більший вплив на банківську справу різних країн.

Уніфікація машиночитаних форматів значно полегшує контроль коректності відправлених повідомлень, що, з одного боку забезпечує захист від випадкових помилок, і, з іншого - підвищує пропускну здатність системи для правильно сформульованих повідомлень. Процеси підготовки і обробки повідомлень повністю піддаються автоматизації, що значно підвищує ефективність і рентабельність банківської діяльності.

Завдання автоматизації будь-якого процесу вимагає від фахівця чималих зусиль і кваліфікації. Це пояснюється тим, що для вирішення такого класу задач необхідно досконально розуміти весь процес, який підлягає автоматизації. В даному випадку, від фахівця потрібні знання технологій баз даних, мережевих

технологій, технологію обробки фінансових повідомлень і стандарти їх оформлення.

Автоматизована система обробки архівних банківських повідомлень призначена для проведення пошуку і вилучення повідомлень фінансової організації з архівних таблиць БД. При цьому повинна бути реалізована можливість конкретизації пошуку повідомлень по їх типам, датою отримання/відправки, напрямку прямування, а також за сумою платежу, датою валютування з позначенням як відправника, так і одержувача повідомлення.

Об'єктом розробки даної кваліфікаційної роботи є автоматизована система обробки архівних повідомлень.

Метою роботи є автоматизація процесу збору інформації про роботу банку, формування звітів, формування необхідних документів і звітів для проведення розслідувань та інших операцій з цінними паперами.

Для вирішення заданої мети необхідно забезпечити автоматичне формування повідомлень про роботу банку з архівних повідомлень, керуючись для пошуку різного роду даними, що вводяться оператором, а також автоматизувати процес отримання всіх необхідних платежів з банківської СУБД.

При розробці інформаційної системи автоматизовано процес формування різного роду звітів про фінансову діяльність банку.

Призначенням даного програмного виробу є формування коректного повідомлення з архівної бази по заданому оператором критерію пошуку. При цьому програма Messenger працює з усіма типами фінансових повідомлень оброблюваних банком.

Так як розроблена система служить для автоматизації процесу обробки архівних повідомлень, які зберігаються в базі даних керованих СУБД SQL AnyWhere 17.0, то для реалізації автоматизованої системи використовується зазначена СУБД для виключення програмної несумісності.

РОЗДІЛ 1

АНАЛІЗ ПРЕДМЕТНОЇ ГАЛУЗІ ТА ПОСТАНОВКА ЗАВДАННЯ

1.1. Загальні відомості з предметної галузі

1.1.1. SWIFT - суспільство всесвітніх банківських комунікацій

Товариство всесвітніх банківських комунікацій - Society for Worldwide Interbank Financial Telecommunication (SWIFT) було засновано групою з 250 банків Європи та Північної Америки в 1973 р 3 9 травня 1977 року, коли система почала функціонувати, і до сьогоднішнього дня число фінансових установ, що користуються послугами цієї організації, перевершило 3700 з 75 країн-членів і 92 країн-учасників. Число транзакцій, що пересилаються цими установами, становить майже 2 млн. щодоби і більше 400 млн. в рік. Обсяг транзакцій при односторонньої передачі становить 325 байт, а при двосторонній - 650 байт. Висока інтенсивність обмінів знижує їх вартість настільки, що вона виявляється - нижче вартості аналогічних передач по телексу і телеграфу. Час же доставки повідомлення по SWIFT становить близько 20 хв, але його можна скоротити до 1-5 хв за додаткову плату, що перекиває показники звичайних каналів зв'язку.

Вступ в SWIFT передбачає укладання контракту користувача, національних органів зі спільнотою.

SWIFT забезпечує єдину мову і єдину технологію організації обробки інформації; надійність передачі і обробки даних та їх захист; швидку передачу повідомлень; скорочення помилок і, отже, втрат; надійний дистанційний контакт між клієнтами; скорочення операційних витрат. У великій мірі ці переваги пов'язані з використанням стандартних повідомлень.

Система SWIFT здійснює:

- перекази клієнтів і банків;
- передачу повідомлень про отримання переказів;
- обмін валюти;

- запити і сповіщення про позики і вкладах;
- довгострокові позики і внески;
- платежі відсотків;
- підтвердження дебету або кредиту;
- виписки з рахунку.

Це дозволяє скоротити застосування паперової технології та перейти до електронного обміну даними (ЕОД) в фінансово-кредитній системі. Якщо спочатку планувалося обслуговування тільки банків, то незабаром система стала доступна і іншим установам (брокерським, дилерським, клірингових конторам та ін.).

Всі платіжні документи надходять в систему в стандартизованому форматі, що дозволяє автоматизувати обробку даних і виключити можливість різного розуміння змісту повідомлень відправником і одержувачем. Фіксація виконаних транзакцій дає можливість повного контролю (аудиту) всіх розпоряджень і щоденного автоматизованого формування звітів по ним. Передача повідомлення та його контроль здійснюються автоматично (в режимі online не довше ніж 20 с).

Багаторівнева комбінація фізичних, технічних і організаційних методів захисту забезпечує збереження і таємність переданих відомостей. Доручення клієнтів і формується звітність можуть використовуватися для оцінки стану рахунків і платежів. Застосування стандартів банківської документації, її уніфікація виключили розбіжності в оформленні документів різними користувачами.

В даний час виділено сім категорій повідомлень, які включають понад 70 їх типів. Макет повідомлення являє собою пронумеровану сукупність обов'язкових і необов'язкових полів. Обов'язкові поля містять інформацію, необхідну для правильної обробки транзакції. Додаткова інформація розташовується в опціональних кодах. При цілодобовій роботі 20% угод SWIFT здійснює без втручання операторів. У разі надходження в SWIFT неякісної інформації, яка не відповідає категоріям, групам і типам встановлених

повідомлень, може застосовуватися експертна система, що здійснює ідентифікацію повідомлення, - ARTRANS (Telex Reader). Вона витягує з неструктурованих повідомлень інформацію на природну мову, що стосується переказів коштів між банками і клієнтами, діючи за типом демонів.

Система може переводити документи з однієї форми в іншу, перераховувати вартість. Вона складається з очищувача повідомлень (виявлення неточностей і описок); аналізатора тексту (ядро); інтерпретатора повідомлень (перевірка номерів рахунків, адрес, підбір форми платежу) і формалізатора змісту транзакції в SWIFT.

Повідомлення кодуються відповідно до розробленого стандартом для різних їх типів. Повідомлення, що надійшли в операційний центр, перевіряються на правильність формату документа, терміну та адреси відправника та одержувача, а також статусу пристрою - джерела повідомлення. У зашифрованому вигляді повідомлення передаються в регіональний процесор країни, де розташований банк-одержувач, а звідти - в банк-одержувач. Термін зберігання протоколів передачі становить 14 днів.

1.1.2. Інтерфейс TurboSWIFT

TurboSWIFT є одним з найпопулярніших і недорогих інтерфейсів класу Gateway, який отримав акредитацію S.W.I.F.T. Інтерфейс здатний відправляти до 10000 - 20000 повідомлень на годину. Враховувалося також і те, що продукт існує з 1991 року і працює практично на всіх існуючих платформах. Операція була дуже важливою для Netik.com, оскільки вона принесла продуктам компанії більш широке поширення і забезпечила міжнародну дистриб'юторську мережу.

Об'єднуючи лідируючі інтеграційні eBusiness-рішення з сучасними розробками компанії MIC Data в області SWIFT і надійно захищених B2B/B2C-адаптерів, отримана можливість виконання більш досконалих проектів по інтеграції програмного забезпечення всередині фінансових організацій і

реалізації STP-рішень.

TurboSWIFT є унікальним рішенням для банків, які тільки починають роботу в мережі SWIFT і мають спочатку невеликий трафік.

TurboSWIFT є ідеальною заміною інтерфейсів ST400 і Merva. Вже сьогодні розроблені інструменти, що дозволяють перенести при міграції не тільки ключі аутентифікації, але і конвертувати в TurboSWIFT самі повідомлення та їх історію.

Функціональні можливості TurboSWIFT були сильно розширені в липні 2000 року, коли в інтерфейсі з'явилася можливість обміну повідомленнями не тільки через мережу SWIFT, а й через IP-мережу, яка отримала назву B2B.

B2B-інтерфейс забезпечує механізм безпечного обміну повідомленнями SWIFT з іншими банками, минаючи мережу SWIFT і використовуючи IP-мережі. Доставка гарантується підтвердженнями (ACK), підтвердження справжності - технологією аутентифікації, аналогічної SWIFT (MAC). Ключі аутентифікації вводяться вручну або створюються банками-кореспондентами за стандартною технологією BKE через SWIFT.

Оскільки TurboSWIFT вже зараз має можливість роботи через IP-мережі, він зможе використовувати і нову IP-мережу S.W.I.F.T.

Однак вибір інтерфейсу TurboSWIFT означав не тільки його подальше використання в якості комунікаційного адаптера в складі продуктів сімейства xNetik або інтерфейсу до SWIFT. TurboSWIFT був використаний в новітній розробці компанії - InterLink.

1.1.3. SWIFT повідомлення

Одне з основних напрямків діяльності суспільства полягає в розробці уніфікованих засобів обміну фінансовою інформацією. З цією метою створена і продовжує удосконалюватися структурована система фінансових повідомлень, за допомогою якої можливо здійснити практично весь спектр банківських та інших фінансових операцій, включаючи операції, що здійснюються на

валютних і фондових біржах.

Формати стандартизованих машинопечатних повідомлень розроблені таким чином, щоб зробити їх найбільш незалежними від національних особливостей банківської сфери в кожній конкретній країні. У той же час уніфіковані формати повідомлень, що використовуються для передачі інформації в мережі SWIFT, поряд з присвоєваними суспільством банківськими ідентифікаційними кодами (восьмизначний код, який є унікальною адресою банківських та інших фінансових інститутів), рекомендовані Міжнародною організацією зі стандартизації (ISO) в якості міжнародних стандартів. Стандарти SWIFT стали стандартами «де факто» для фінансових повідомлень транзакцій, роблячи усе більший вплив на банківську справу різних країн.

Уніфікація машиночитаних форматів значно полегшує контроль коректності відправлених повідомлень, що, з одного боку забезпечує захист від випадкових помилок, і, з іншого - підвищує пропускну здатність системи для правильно сформульованих повідомлень. Процеси підготовки і обробки повідомлень повністю піддаються автоматизації, що значно підвищує ефективність і рентабельність банківської діяльності.

В даний час використовується 11 категорій, що охоплюють понад 130 типів повідомлень (Message Transaction - MT), побудованих таким чином, щоб забезпечувати виконання фінансових операцій з великою точністю.

Повідомлення, як правило, передаються від одного користувача до іншого, але існує категорія системних повідомлень, які дозволяють користувачеві взаємодіяти з мережею (0 категорія). Системні повідомлення використовуються для запиту певних дій і отримання спеціальних звітів, пошуку повідомлень у базі даних, для навчальних і тренувальних цілей. Користувач може отримувати від мережі запити, або вона може інформувати його про свій поточний стан, оновлення, нові послуги і т.д.

Системні повідомлення користуються найвищим пріоритетом, оскільки вони містять інформацію, що стосується функціонування мережі.

Банківські повідомлення ділять на термінові і звичайні, причому за відправку термінових повідомлень стягується спеціальний тариф.

Всім іншим типам повідомлень, які належать до категорій 1-9 і n, привласнені тризначні цифрові коди, причому перша цифра відповідає категорії операції.

Категорія n - повідомлення загальної групи. Кожне повідомлення з загальної групи може використовуватися в будь-який з категорій 1-9. Коди повідомлень загальної групи виглядають як n9M, де n - замінюється номером тієї категорії, яка найкращим чином відповідає меті повідомлення, 9 вказує на особливий характер повідомлення в кожній з категорій, M визначає конкретний тип повідомлення (наприклад, 0 - повідомлення, 2 - вимоги про анулювання, 5 і 6 - відповідно, запити і відповіді).

Повідомлення всіх типів побудовані по загальному принципу. Вони складаються з початкової частини, в яку входить мітка початку повідомлення (Start of Message), заголовок (Header), мітка початку тексту повідомлення (Start of Text), тексту повідомлення (Text of Message) і закінчення повідомлення, в яке входять мітка кінця тексту (End of Text), параметри (Trailer) і мітка кінця повідомлення (End of Message).

Початкова частина і закінчення утворюють «конверт», в якому пересилаються повідомлення і який містить інформацію необхідну для управління руху повідомлення в мережі. Тема містить одинадцятизначний код-ідентифікатор одержувача повідомлення (Bank Identifier Codes, BIC, який є адресою в мережі), код терміналу відправника, поточний п'ятизначний номер, що виконує контрольну і захисну функції, тризначний код повідомлення і з двозначним кодом пріоритету. У параметрах вказується код аутентифікації й інші повідомлення, наприклад, попередження банку-одержувача про можливість затримки в передачі повідомлення, попередження про можливість подвійного платежу і т.д. При передачі повідомлення текст повідомлення система «не бачить».

Текст повідомлення складається з полів, позначених двозначним цифровим кодом. Наприклад, код 57 означає банк, в якому ведеться рахунок, 69- беніфіціари, 71 - за чий рахунок проводиться платіж і сума комісії, поле 32 - сума платежу. У текст повідомлення інформація вноситься в строгій послідовності. Заповнення частини полів є обов'язковим. Для кожного типу повідомлення визначений свій набір заповнених полів. Обов'язкові поля містять інформацію, необхідну для правильної обробки повідомлення.

Необхідно мати на увазі, що хвостовик генерується автоматично термінальним обладнанням підключення SWIFT. Текст і заголовок повідомлення вводяться фахівцями банку відповідно до правил заповнення полів.

Приклад трансформації платіжного доручення в форматі Банку в повідомлення MT103 для передачі по системі SWIFT наведено на рис. 1.1-1.2.

Поступ. в банк плат.		Списано со сч. плат.		0401060
ПЛАТЕЖНОЕ ПОРУЧЕНИЕ № 234		21 августа 2020 г.	електронно	
		Дата	Вид платежа	
Сумма прописью	<i>Четыреста десять тысяч семьсот тридцать один рубль 80 копеек</i>			
ИНН 7741070139	КПП	Сумма	410731-80	
Надежный банк		Сч. №	30214810000000000989	
Плательщик		БИК	044583258	
НКО Расчетная палата РТС (ООО)		Сч. №	30103810200000000258	
Банк плательщика		БИК	044583789	
Надежный банк		Сч. №	30101810700000000001	
Банк получателя		Сч. №	40702810400000000005	
ИНН 7705070139	КПП	Сч. №	40702810400000000005	
Отличные инвестиции ООО		Вид оп.	01	Срок плат.
		Наз. пл.		Очер. плат. 6
Получатель		Код		Рез. поле
Перечисление средств от продажи акций по Договору 35 от 21.03.20. НДС не облагается.				
Назначение платежа				
Подписи Отметки банка				
М.П.				

Рис. 1.1. Вихідний документ Клієнта

:20 SENDER'S Reference : **+012345678901234**

:23B BANK OPERATION CODE : **CRED**

:32A Value Date/Currency INERBANK SETTLED AmOUNT : **030821RUB410731,8**

:50K Ordering Customer : **/30214810000000000989**

Inn7741070139

NADoJNYi BANK

:52D ORDERING INSTITUTION : **//RU044585258.3010381020000000258**

NKO RAScoTNAa PALATA RTS (OOO)

:57D Account With Institution : **//RU044583789.30101810700000000001**

NADoJNYi BANK

:59 Beneficiary Customer : **/40702810400000000005**

Inn7705070139

OTLicNYE INVESTICII OOO

:70 REMITANCE INFORMATION : **PEREcISLENIE SREDSTV OT PRODAJI**

AKCii PO DOGOVORU 35 OT 21.03.20.

NDS NE OBLAGAETSa.

:71A DETAILS OF CHARGES : **OUR**

:72 Sender to Receiver Information : **/RPP/234.030821.6.ELEK.030821**

Рис. 1.2. Swift- повідомлення в форматі mt103, сформоване на підставі вихідного документа клієнта

1.1.4. Передача повідомлень в форматі SWIFT через Інтернет

Використання InterLink дозволяє фінансовим установам посилати всі типи фінансових повідомлень, таких, як SWIFT, ISITC, XML, FIX і EDIFACT, поряд з неструктурованими повідомленнями, використовуючи найбільш зручний канал доставки - Інтернет, SWIFT або іншу мережу.

Вперше InterLink. був аносований 19 серпня 2000 р. Пізніше, на

щорічній виставці SIBOS з 11 по 15 вересня Interlink був продемонстрований учасникам виставки. В цей час було оголошено про пілотній програмі, протягом якої організації могли отримати у тимчасове користування продукт Turbo-SWIFT. Учасники програми могли протестувати обмін фінансовими повідомленнями через Інтернет як між собою, так і з Application Service Provider (ASP). Крім цієї, ASP дозволяє організувати підтвердження з відправникам в разі вдалої доставки повідомлень одержувачам. Це є реалізацією власного протоколу - Positive Acceptance Protocol (PAP). Повідомлення зберігаються в ASP 12 місяців, що гарантує користувачам дозвіл можливих конфліктних ситуацій. Для забезпечення безпеки при передачі повідомлень використані RSA-алгоритми і пристрої читання смарт-карток (Smart Card Readers). За допомогою ASP створена також інфраструктура відкритих ключів - PKI.

1.2. Призначення розробки та галузь застосування

Призначенням даної інформаційної системи є запит та формування коректного повідомлення з архівної бази комерційного банку по заданому оператором критерію пошуку. При цьому програма Messenger працюватиме з усіма типами фінансових повідомлень оброблюваних банком.

Система призначена для використання в комерційному банку.

1.3. Підстава для розробки

Підставою для розробки кваліфікаційної роботи бакалавра на тему «Розробка інформаційної системи обробки платіжних повідомлень комерційного банку» є наказ по Національному технічному університету «Дніпровська політехніка» від __.__.2021р. № ____-__.

1.4. Постановка завдання

Метою кваліфікаційної роботи є автоматизація процесу збору статистичної інформації про роботу банку, формування звітів, надання необхідних документів і звітів для проведення розслідувань та інших операцій з цінними паперами.

Для вирішення заданої мети необхідно забезпечити автоматичне формування повідомлень про роботу банку з архівних повідомлень, керуючись для пошуку різного роду даними, що вводяться оператором, а також автоматизувати процес отримання всіх необхідних платежів з банківської СУБД.

При розробці інформаційної системи необхідно забезпечити автоматизацію процесу формування різного роду звітів про фінансову діяльність банку.

Для виконання роботи необхідно виконати наступні етапи проектування та розробки ІС:

1. Проаналізувати предметну галузь об'єкту розробки.
2. Обґрунтувати структуру, характер і організацію вхідних і вихідних даних.
3. Визначити функціональні характеристики розробленої системи.
4. Розробити алгоритм роботи системи, методи, відповідно до яких проводиться необхідний розрахунок та її логічну структуру.
5. Забезпечити можливість зв'язку програми з іншим ПЗ, яке впроваджене в банку.
6. Програмно реалізувати задану систему.
7. Протестувати систему та виправити можливі недоліки.

1.5. Вимоги до програми або програмного виробу

1.5.1. Вимоги до функціональних характеристик

В даній роботі необхідно розробити інформаційну систему для проведення економічних розслідувань з архівної банківської документації. Система повинна надавати наступні функціональні можливості:

- формування запиту в архівну базу банку з метою отримання необхідної інформації про дане повідомлення;
- формування повідомлення з архівної таблиці з метою отримання необхідної інформації по даному запиту оператора;
- проведення розслідувань з нез'ясованих платіжних повідомлень шляхом проведення запитів в архівну БД;
- формування звітів про роботу банку за певний період часу: кількість проведених платіжних повідомлень, кількість проведених транзакцій грошових сум, що списані чи зараховані на рахунки банку;
- збереження і друк звітів;
- збереження і друк повідомлень.

Введення інформації повинно здійснюватися за допомогою консолі з використанням діалогових віконних систем, доступ до яких вимагає аутентифікації користувача.

Вивід знайденої інформації повинен бути реалізований у вигляді звітів. Звіти повинні бути представлені користувачеві у вигляді структурованих документів в форматі, затвердженому в даній фінансовій установі, і повинен бути доступний для перегляду, друку і збереження в файловий вид.

1.5.2. Вимоги до інформаційної безпеки

Безпека даних включає їх цілісність і захист. Цілісність даних - стійкість даних, що зберігаються до руйнування і знищення, пов'язаних з

несправностями технічних засобів, системними помилками і помилковими діями користувачів.

Вона передбачає:

- відсутність неточно введених даних або двох однакових записів про одне й те ж факт;
- захист від помилок при оновленні бази даних;
- неможливість видалення (або каскадне видалення) пов'язаних даних різних таблиць;
- збереження даних при збої техніки (відновлення даних).

Цілісність даних полягає у логічній і фізичній цілісності даних.

Логічна цілісність - виражається у вигляді обмежень або правил збереження даних, несуперечність яких не повинна порушуватися в базі. Можна вказати межі значень при описі структури бази даних або таблиць бази.

Фізична цілісність виражається в захисті даних від фізичного руйнування в результаті збоїв і відмов устаткування. Для цього використовують контрольні копії даних, тобто запам'ятовування станів бази даних у контрольних точках, а також ведення системного журналу, в якому відображаються усі маніпуляції над базою даних.

Одним з видів захисту бази даних являється різноманітність прав доступу, що гарантує доступ до бази даних тільки санкціонованих користувачів.

Введення інформації в базу даних повинно бути мінімізоване, тобто інформація вводиться однократно, а її зміни відображаються тільки у виведених даних.

База даних повинна забезпечувати швидкий і гнучкий доступ до даних.

1.5.3. Вимоги до складу та параметрів технічних засобів

При роботі з системою повинні дотримуватися умови, що забезпечують нормальний режим експлуатації IBM PC-сумісних машин. Кількість персоналу, що працює з програмним продуктом - одна людина. Працювати з програмою

повинен висококваліфікований оператор баз даних.

Для технічних засобів повинна бути обрана конфігурація, що забезпечує цілодобову роботу програмних комплексів з резервуванням даних. Для цього повинно бути використано два сервера, один з яких виконує функцію прийому і відправки повідомлень за допомогою програмного продукту TurboSwift, а також здійснює всю пост обробку повідомлень:

- аналіз повідомлень і маршрутизацію прийнятих повідомлень;
- аналіз і перевірку на наявність помилок відправляються банком SWIFT - повідомлень;

Другий сервер виконує функцію резервування даних і є копією першого.

1.5.4. Вимоги до інформаційної та програмної сумісності

Розроблена система повинна взаємодіяти з банківською БД і автоматизувати процес обробки архівних повідомлень, які зберігаються в базі даних керованих СУБД SQL AnyWhere 17.0.

РОЗДІЛ 2

ПРОЄКТУВАННЯ ТА РОЗРОБКА ІНФОРМАЦІЙНОЇ СИСТЕМИ

2.1. Функціональне призначення системи

Інформаційна система обробки архівних повідомлень призначена для проведення пошуку і вилучення повідомлень фінансової організації з архівних таблиць БД. При цьому реалізована можливість конкретизації пошуку повідомлень по їх типам, датою отримання / відправки, напрямку прямування, а також за сумою платежу, датою валютування з позначенням, як відправника, так і одержувача повідомлення.

Вивід знайденої інформації реалізований у вигляді звітів, які представлені користувачеві у вигляді структурованих документів в форматі, затвердженому в даній фінансовій установі, і доступні для перегляду, друку і збереження в файловий вид.

2.2. Опис застосованих математичних методів

Ця інформаційна система не вимагає розробки математичної моделі і застосування математичних методів.

2.3. Опис використаних технологій та мов програмування

Для реалізації заданої роботи було обрано візуальне середовище програмування Microsoft Visual Studio 2017 і сервери баз даних Sybase: SQL Anywhere 17.0, і SYBASE System 11.5 Adaptive Server. Вибір був зроблений виходячи з наявних на підприємстві програмних засобів. Для організації обміну даними між різними системами управління БД, використовуємо програмну модель доступу до БД ADO.NET.

Програма написана мовою C ++. Вона широко використовується для розробки програмного забезпечення, будучи одним з найпопулярніших мов програмування. Область його застосування включає створення операційних систем, різноманітних прикладних програм, драйверів пристроїв, додатків для вбудованих систем, високопродуктивних серверів, а також ігор. Існує безліч реалізацій мови C ++, як безкоштовних, так і комерційних і для різних платформ. Наприклад, на платформі x86 це GCC, Visual C ++, Intel C ++ Compiler, Embarcadero (Borland) C ++ Builder і інші. C ++ зробив величезний вплив на інші мови програмування, в першу чергу на Java і C #.

Мова виникла на початку 1980-х років, коли співробітник фірми Bell Labs Бйорн Страуструп придумав ряд удосконалень до мови C під власні потреби. Коли в кінці 1970-х років Страуструп почав працювати в Bell Labs над завданнями теорії черг (в додатку до моделювання телефонних викликів), він виявив, що спроби застосування існуючих в той час мов моделювання виявляються неефективними, а застосування високоефективних машинних мов занадто складно через їх обмеженою виразності.

Стандарт C ++ складається з двох основних частин: опис ядра мови і опис стандартної бібліотеки.

Розвиток мови супроводив розвиток крос-компілятора cfront. Нововведення в мові відбивалися в зміні номера версії крос-компілятора. Ці номери версій крос-компілятора розповсюджувалися і на саму мову, але стосовно до теперішнього часу мова про версії мови C++ не ведуть. Лише в 1998 році мова стала стандартизованим.

Стандартна бібліотека C++ включає стандартну бібліотеку Cі з невеликими змінами, які роблять її більш відповідною для мови C ++. Інша велика частина бібліотеки C++ заснована на Стандартній Бібліотеці Шаблонів (STL). Вона надає такі важливі інструменти, як контейнери (наприклад, вектори і списки) і ітератори (узагальнені вказівники), що надають доступ до цих контейнерів як до масивів. Крім того, STL дозволяє схожим чином працювати і

з іншими типами контейнерів, наприклад, асоціативними списками, стеками, чергами.

Доступ до можливостей стандартної бібліотеки C++ забезпечується за допомогою включення в програму (за допомогою директиви `#include`) відповідних стандартних заголовків файлів. Всього в стандарті C++ 11 визначено 79 таких файлів. Засоби стандартної бібліотеки оголошуються які входять в простір імен `std`. Заголовки, імена яких відповідають шаблону «X», де X - ім'я заголовки стандартної бібліотеки C без розширення (`cstdlib`, `cstring`, `cstdint` та ін.). Містять оголошення, відповідні даної частини стандартної бібліотеки C. Стандартні функції бібліотеки C також знаходяться в просторі назв `std`.

C++ містить засоби розробки програм контрольованої ефективності для широкого спектра задач, від низькорівневих утиліт і драйверів до вельми складних програмних комплексів. Зокрема висока сумісність з мовою Cі: код на Cі може бути з мінімальними переробками скомпільована компілятором C++. Візуально-мовний інтерфейс є прозорим, так що бібліотеки на Cі можуть викликатися з C++ без додаткових витрат, і більш того при певних обмеженнях код на C++ може експортуватися зовні не відрізнятися від коду на Cі.

До числа недоліків мови можна віднести:

- відсутність системи модулів C++ успадкував від Cі підключення заголовних файлів за допомогою препроцесора. Це змушує дублювати опису об'єктів, породжує неочевидні вимоги до коду і збільшує обсяг компілюемого тексту, а значить і час компіляції;

- успадковані від Cі небезпечні і провокують помилки можливості (макроозначення, адресна арифметика, неявне приведення типів, можливість прямого управління розподілом пам'яті).

- відсутність вбудованих механізмів статичної валідації часу життя об'єктів, що приводить до раптового краху програм через звернення до знищеної змінної, або через неправильну багатопотокової роботи з об'єктами.

– шаблони породжують об'ємний і не завжди оптимальний код. Часткове визначення шаблонів ускладнює як сама мова, так і програми, де воно використовується.

Єдиним прямим нащадком C ++ є мовою D, задуманий як переробка C ++ для усунення найбільш очевидних його проблем. Автори відмовилися від сумісності з Cі, зберігши синтаксис і багато базові принципи C ++ і ввівши в мову можливості, характерні для нових мов. У D немає препроцесора, заголовків файлів, множинного спадкоємства, але є система модулів, інтерфейси, асоціативні масиви, підтримка unicode в рядках, прибирання сміття (при збереженні можливості ручного управління пам'яттю) вбудована багатопоточність, висновки типів, явне оголошення чистих функцій і незмінних значень. Використання D вельми обмежена, вважати його реальним конкурентом C ++ можна [17].

Sybase SQL AnyWhere є повнофункціональну СУБД на Intel-платформі для мобільних і невеликих груп користувачів. Дана СУБД дозволяє розробляти програми на основі технології "клієнт-сервер" на платформах Windows NT, Windows 7, OS/2, NetWare, Solaris / Sparc, HP-UX, AIX, DOS і QNX.

СУБД SYBASE SQL Anywhere є складовим елементом системи SYBASE System 11 і її подальшої версії SYBASE System 11.5 Adaptive. Основним елементом цих систем є потужна СУБД SYBASE SQL Server. Вона дозволяє зберігати величезні обсяги інформації і обробляти запити до баз даних із застосуванням технології клієнт-сервер. СУБД SYBASE SQL Server і SYBASE SQL Anywhere взаємно доповнюють один одного. Розробники SYBASE SQL Anywhere намагалися досягти максимальної сумісності баз даних, що створюються цієї СУБД, з базами даних SYBASE SQL Server. Саме цим пояснюється включення в SQL Anywhere деяких елементів SYBASE SQL Server.

SYBASE SQL Anywhere може поставлятися як в мережевому варіанті СУБД, так і автономному варіанті. В останньому випадку всі компоненти СУБД функціонують на тому ж комп'ютері, що і додаток. Для додатків не має

значення, який варіант СУБД вони використовують. В рамках локальних мереж взаємодію з додатками-клієнтами здійснюється за допомогою таких мережевих протоколів як NetBIOS, TCP/IP і IPX. Бази даних, створені засобами SQL Anywhere, сумісні з подальшими версіями і стерпні між платформами, на яких може функціонувати дана СУБД.

SYBASE SQL Anywhere має вбудований інтерфейс з рядом програмних систем, наприклад таких як Powersoft PowerBuilder, Optima C++ і Power Designer. Взаємодія з іншими системами і додатками здійснюється за допомогою наступних програмних інтерфейсів:

1. Специфікація ODBC. Дана специфікація надає користувачам уніфікований інтерфейс між додатками і реляційними базами даних, в тому числі і базами даних SQL. Цей інтерфейс являє собою сукупність функцій. Ці функції організують взаємодію додатків з базами даних в операційних системах Windows 3.x, Windows 2003, OS/2 і Windows NT. Специфікація ODBC представляє собою інтерфейс нижнього рівня.

2. Вбудований SQL (Embedded SQL interface) дозволяє "вбудовувати" SQL-оператори прямо в тексти програм-додатків на мовах C або C++. Згодом такі додатки піддаються обробці спеціальним препроцесором, який замінює SQL-оператори на виклики відповідних функцій і процедур. Після цього перетворені тексти програм представляють собою тексти програм на мові програмування C++.

3. DDE-інтерфейс. Це технологія динамічного зв'язування об'єктів використовується в операційній системі Windows 3.x .. Для застосування цієї технології потрібне використання відповідних програмних засобів, що підтримують функціонування DDE-клієнта. З іншого боку, в якості серверів необхідно використовувати додатки, що забезпечують режим DDE-сервера. До числа останніх відносяться такі програмні додатки, як Microsoft Access, Microsoft Excel і ряд інших. В СУБД SYBASE SQL Anywhere роль DDE-сервера виконує WSQL (Watcom SQL) DDE Server.

4. Високорівневий власний інтерфейс WSQL HLI (Watcom SQL High-

level interface) забезпечує на "високому" рівні взаємодію між SQL Anywhere і додатками в середовищах операційних систем Windows 3.x, Windows 7, OS/2 і Windows NT. Даний інтерфейс реалізований для систем програмування аналогічних Visual Basic (для Windows 7 і Windows NT) і REXX (для OS/2).

Для реалізації даної роботи, найкраще підходить специфікація ODBC, що надає уніфікований інтерфейс між додатками і реляційними базами даних.

СУБД SYBASE SQL Anywhere має в своєму складі мережевий і автономний варіанти СУБД. Ці варіанти реалізовані у вигляді компонент SQL Anywhere server/client і SQL Anywhere engine. SQL Anywhere server/client представляє мережевий варіант СУБД. Він організовує взаємодією додатків з базами даних в рамках локальної обчислювальної мережі за технологією "клієнт-сервер".

Крім локальної мережі, СУБД SYBASE SQL Anywhere надає можливість використовувати технологію "клієнт-сервер" і в рамках мережі Internet. Для цього необхідно, щоб на комп'ютері з сервером бази даних був організований веб-сервер, і до складу його програмного забезпечення входив додаток SYBASE NetImpact Dynamo.

Мережевий варіант включає в себе віддалений сервер і клієнтську компоненту; SQL Anywhere server і SQL Anywhere client, відповідно. Функціонування СУБД в рамках мережі полягає в наступному. На одній ЕОМ, що грає роль сервера бази даних, запускається віддалений сервер - SQL Anywhere server. Його завдання полягає в очікуванні запитів від клієнтських ЕОМ, їх обробці та відправленні результатів клієнтів. Під час роботи сервера на клієнтських ЕОМ функціонує компонента SQL Anywhere client. Вона приймає від клієнтських додатків запити. Ці запити являють собою оператори мови SQL, наприклад, SELECT (вибрати дані), UPDATE (модифікувати дані), CALL (викликати збережену процедуру) та інші. Отримані запити компонента SQL Anywhere client відправляє по мережі сервера бази даних і очікує повернення від нього результатів цих запитів.

Компонента SQL Anywhere engine є автономним варіантом даної СУБД і

являє собою локальний сервер баз даних. Він реалізує технологію "клієнт-сервер" в умовах, коли сервер і клієнти функціонують на одній ЕОМ. База даних зберігається на цій же ЕОМ. Сервер приймає запити безпосередньо від додатків, обробляє їх і відправляє додаткам результати запитів.

Клієнтські програми не розрізняють, з яким варіантом СУБД вони працюють. Одні і ті ж запити виконуються для користувача абсолютно однаково за одними і тими ж правилами.

Компоненти SQL Anywhere server/client і SQL Anywhere engine складають основу СУБД. Однак поряд з ними до складу СУБД входить і ряд інших компонентів. Всі разом ці компоненти складають комплект поставки SQL Anywhere, в який входять:

- SQL Anywhere server / client - мережевий варіант СУБД;
- SQL Anywhere engine - автономний варіант СУБД;
- SQL Remote - компонента підтримки режиму реплікації;
- Sybase SQL Central - утиліта адміністрування баз даних;
- ISQL (Interactive SQL) - утиліта інтерактивної взаємодії з базами даних;
- набір додаткових утиліт для створення баз даних, їх стиснення, знищення і т.д.

Наведені компоненти є елементами стандартної поставки. Крім стандартної поставки СУБД може пропонуватися у вигляді професійної поставки. Вона має ще більше число складових.

Бази даних, що створюються СУБД SYBASE SQL Anywhere, являють собою реляційні бази даних. Ці бази даних складаються з сукупності об'єктів. Такими об'єктами є:

- Таблиці даних - зберігають дані, що становлять основний зміст БД;
- Ключі - сукупності атрибутів, що утворюють ключі (первинні і зовнішні), призначені для здійснення прискореного пошуку даних і забезпечення обмежень посилальної цілісності;
- Індeksi - спеціальні таблиці, призначені для швидкого пошуку

необхідної інформації в таблицях даних;

- Представлення (Views) - пов'язані сукупності підмножин таблиць даних, що надаються користувачам для обмеження їх доступу до таблиць даних. При цьому, до одних таблиць у доступі повністю, а в інших таблицях доступ дозволяється тільки до деяких записів цих таблиць;

- Збережені процедури і функції: збережені в базі даних підпрограми на мові SQL, скористатися які може будь-який користувач, який має на це право;

- Тригери - підпрограми, що активізуються при настанні певних подій, наприклад, видалення запису з таблиці, модифікація записів і т.д. Тригери є потужним засобом забезпечення цілісності даних;

- Користувальницькі типи даних - типи даних, створювані користувачем на підставі базових типів даних СУБД;

- Системні таблиці - зберігають всю інформацію про схему бази даних і що містяться в ній об'єктах.

У даній СУБД використовується діалект мови SQL - Watcom SQL. Він відповідає стандартам ANSI SQL/89 Level 2 і IBM SAA. Крім того, використовуваний діалект SQL підтримує нові можливості і розширення стандартів ANSI SQL/92 і IBM's DB2

Вся інформація бази даних може розміщуватися в декількох областях. Областю є файл з розширенням db, в якій зберігається вся база даних або один з її фрагментів. Кожна область характеризується своїм ім'ям і файлом, відповідним цій галузі. Спочатку база даних займає тільки одну область з ім'ям SYSTEM, якій відповідає базовий файл (root file). Потім, у міру необхідності, простір зовнішньої пам'яті бази даних може розширюватися за рахунок додавання нових областей. Ці файли областей можуть бути розміщені в будь-якому каталозі на будь-якому диску і будь-якому вузлі локальної мережі.

Фізично кожен файл (область) складається зі сторінок фіксованого обсягу. Розмір сторінок встановлюється при створенні бази даних. У сторінках розміщуються об'єкти бази даних. Залежно від розміру об'єктів кожна сторінка може містити кілька примірників об'єктів або фрагмент одного з об'єктів.

Поряд з файлами областями до складу бази даних входить файл для зберігання журналу змін БД - файл з розширенням log. Наявність журналу змін є основою маніпулювання даними в даній СУБД з використанням механізму транзакцій. Такий підхід забезпечує високий захист баз даних від виникаючих програмних і апаратних збоїв і відмов.

Для реалізації бази даних, в якій буде зберігатися вся необхідна для роботи програми інформація, в даній роботі використана технологія роботи з БД на основі SQL-AnyWhere 17.0.

Для реалізації взаємозв'язку з БД банку використана технологія доступу до баз даних ADO.NET, використовуючи при цьому специфікацію мови SQL - Watcom.

Для реалізації клієнтської частини інформаційної системи використана середу візуального програмування Microsoft Visual 2010.

2.4. Опис структури системи та алгоритмів її функціонування

2.4.1. Проектування БД ІС

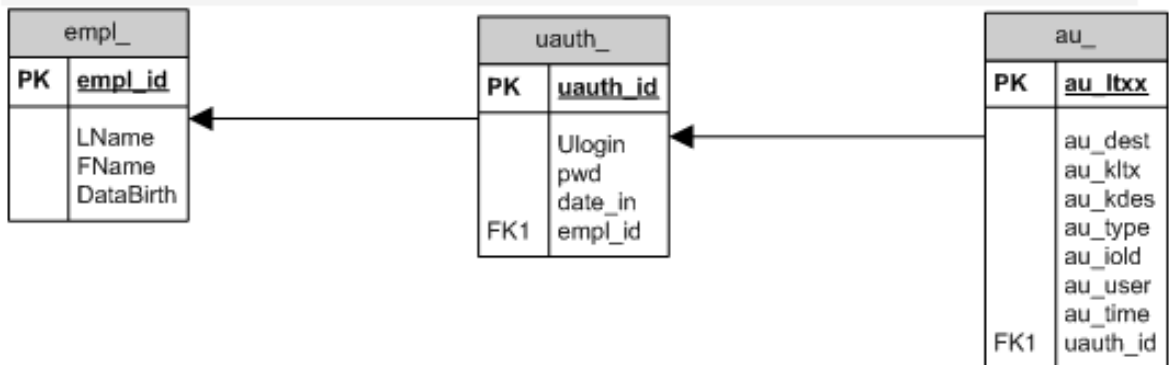
2.4.1.1. Опис логічної структури бази даних

У даній логічній моделі наведений зв'язок таблиць в реляційної моделі БД архівних повідомлень. Логічно всю БД програми можна розділити на три складових:

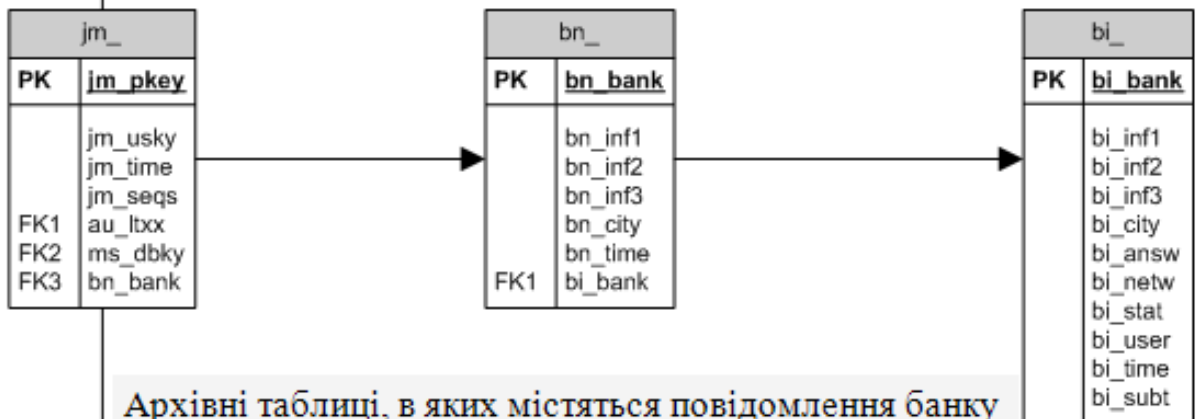
1. Таблиці для зберігання інформації про користувача і про те ким і коли було модифіковано повідомлення в БД.
2. Таблиці журналювання, що описують мнемоніку повідомлень, коли і якою філією вони створені тощо.
3. Архівні таблиці, в яких містяться повідомлення банку.

Таблиці для зберігання інформації про користувачів

та час й місце зміни повідомлення



Таблиці журналювання, що описують мнемоніку повідомлень,
коли і якою філією вони створені тощо.



Архівні таблиці, в яких містяться повідомлення банку

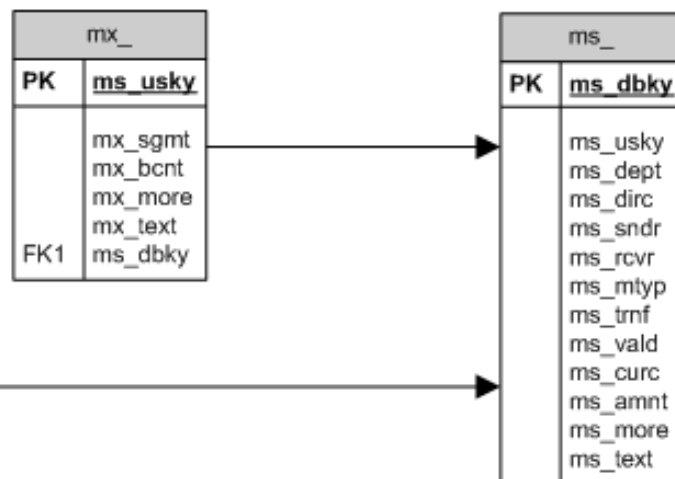


Рис. 2.1. Логічна структура бази даних

2.4.1.2. Опис фізичної структури бази даних

В наведених нижче таблицях (табл. 2.1, 2.2) представлена фізична структура БД розробленої системи.

Таблиця 2.1

Таблиця «Користувачі»

Name	Data Type	Comment
empl_id	integer	ID
LName	varchar(30)	Ім'я
FName	varchar(30)	Прізвище
DataBirth	char(8)	Дата народження

Таблиця 2.2

Таблиця «Дані»

Name	Data Type	Comment
uauth_id	Integer	ID
Ulogin	varchar(30)	Логін
pwd	char(8)	Пароль
date_in	char(8)	Дата входу

Інші поля і типи даних, визначені в БД:

– tamnt, дані цього типу служать для представлення інформації про суму платежу та суму запитуваних комісій, тип даних реалізований від базового char с розмірністю 17 байт;

– tcurc, в змінних цього типу зберігається валюта, тип даних char (3);

- tid, змінна - генератор унікальних значень цілого типу, використовується для формування первинного ключа в БД, базовий тип-integer;
- ms_dbku, в змінній зберігається посилання з бази даних back_office повідомлення, наприклад поле 20 і 21, базовий тип char;
- tsndr, дані цього типу служать для представлення інформації про відправника повідомлення, базовий тип char;
- tvald, - дата валютування, базовий тип char.

2.4.1.3. Програмування БД системи

```

CREATE TABLE "DBA"."au_"
(
    "au_ltxx"          char(8) NOT NULL ,
    "au_dest"         char(8) NOT NULL ,
    "au_kltx"         char(8) NULL ,
    "au_kdes"         char(8) NULL ,
    "au_type"         char(2) NULL ,
    "au_iold"         char(32) NULL ,
    "au_oold"         char(32) NULL ,
    "au_olda"         char(6) NULL ,
    "au_icur"         char(32) NULL ,
    "au_ocur"         char(32) NULL ,
    "au_cura"         char(6) NULL ,
    "au_inew"         char(32) NULL ,
    "au_onew"         char(32) NULL ,
    "au_newa"         char(6) NULL ,
    "au_user"         char(10) NULL ,
    "au_time"         char(14) NULL ,
    PRIMARY KEY ("au_ltxx", "au_dest"),
)

```



```

CREATE TABLE "DBA"."bi_"
(
    "bi_bank"          char(11) NOT NULL ,
    "bi_inf1"         char(35) NULL ,
    "bi_inf2"         char(35) NULL ,
    "bi_inf3"         char(35) NULL ,
    "bi_city"         char(35) NULL ,
    "bi_answ"         char(35) NULL ,
    "bi_netw"         char(5) NULL ,
    "bi_stat"         char(1) NULL ,
    "bi_user"         char(10) NULL ,
    "bi_time"         char(14) NULL ,
    "bi_subt"         char(4) NULL ,
    PRIMARY KEY ("bi_bank"),
)

```

```

CREATE TABLE "DBA"."bn_"
(
    "bn_bank"          char(6) NOT NULL ,
    "bn_inf1"         char(35) NULL ,
    "bn_inf2"         char(35) NULL ,
    "bn_inf3"         char(35) NULL ,
    "bn_city"         char(35) NULL ,
    "bn_time"         char(14) NULL ,
    PRIMARY KEY ("bn_bank"),
)

```

```

CREATE TABLE "DBA"."jm_"
(
    "jm_pkey"          char(18) NOT NULL ,

```

"jm_dbky"	char(18) NULL ,
"jm_bran"	char(8) NULL ,
"jm_bkey"	char(18) NULL ,
"jm_usky"	char(20) NULL ,
"jm_dept"	char(8) NULL ,
"jm_time"	char(14) NULL ,
"jm_seqs"	char(2) NULL ,
"jm_ltxx"	char(8) NULL ,
"jm_term"	char(1) NULL ,
"jm_appl"	char(1) NULL ,
"jm_dirc"	char(1) NULL ,
"jm_mtyp"	char(3) NULL ,
"jm_prio"	char(1) NULL ,
"jm_sndr"	char(12) NULL ,
"jm_rcvr"	char(12) NULL ,
"jm_trnf"	char(16) NULL ,
"jm_vald"	char(8) NULL ,
"jm_curc"	char(3) NULL ,
"jm_amnt"	char(17) NULL ,
"jm_user"	char(10) NULL ,
"jm_arg1"	char(40) NULL ,
"jm_arg2"	char(20) NULL ,
"jm_arg3"	char(20) NULL ,
"jm_arg4"	char(20) NULL ,
"jm_arg5"	char(20) NULL ,
"jm_arg6"	char(20) NULL ,
"jm_arg7"	char(20) NULL ,
"jm_arg8"	char(20) NULL ,
"jm_actn"	char(3) NULL ,
	PRIMARY KEY ("jm_pkey"),

)

CREATE TABLE "DBA"."ms_"

(

"ms_dbky"	char(18) NOT NULL ,
"ms_bran"	char(8) NULL ,
"ms_bkey"	char(18) NULL ,
"ms_usky"	char(20) NULL ,
"ms_note"	char(144) NULL ,
"ms_murx"	char(16) NULL ,
"ms_dept"	char(8) NULL ,
"ms_time"	char(14) NULL ,
"ms_flag"	char(16) NULL ,
"ms_user"	char(10) NULL ,
"ms_orig"	char(10) NULL ,
"ms_netw"	char(8) NULL ,
"ms_queu"	char(8) NULL ,
"ms_dual"	char(8) NULL ,
"ms_stat"	char(1) NULL ,
"ms_ltxx"	char(8) NULL ,
"ms_term"	char(1) NULL ,
"ms_appl"	char(1) NULL ,
"ms_sess"	char(4) NULL ,
"ms_seqn"	char(6) NULL ,
"ms_dirc"	char(1) NULL ,
"ms_sndr"	char(12) NULL ,
"ms_rcvr"	char(12) NULL ,
"ms_mtyp"	char(3) NULL ,
"ms_prio"	char(1) NULL ,
"ms_delm"	char(1) NULL ,

"ms_obsp"	char(3) NULL ,
"ms_itim"	char(4) NULL ,
"ms_mirx"	char(28) NULL ,
"ms_otim"	char(4) NULL ,
"ms_morx"	char(28) NULL ,
"ms_ackf"	char(1) NULL ,
"ms_nakr"	char(6) NULL ,
"ms_trnf"	char(16) NULL ,
"ms_vald"	char(8) NULL ,
"ms_curc"	char(3) NULL ,
"ms_amnt"	char(17) NULL ,
"ms_refp"	char(36) NULL ,
"ms_ref1"	char(56) NULL ,
"ms_aust"	char(2) NULL ,
"ms_auth"	char(8) NULL ,
"ms_pacs"	char(2) NULL ,
"ms_pacr"	char(8) NULL ,
"ms_encr"	char(36) NULL ,
"ms_chks"	char(12) NULL ,
"ms_trng"	char(1) NULL ,
"ms_pdex"	char(1) NULL ,
"ms_pdmx"	char(1) NULL ,
"ms_sysx"	char(1) NULL ,
"ms_dlmx"	char(1) NULL ,
"ms_pdeu"	char(36) NULL ,
"ms_dupi"	char(1) NULL ,
"ms_dupo"	char(1) NULL ,
"ms_dlvx"	char(1) NULL ,
"ms_dlvk"	char(18) NULL ,
"ms_tkrf"	char(8) NULL ,

```

"ms_tkst"          char(4) NULL ,
"ms_tkky"          char(10) NULL ,
"ms_cdbk"          char(18) NULL ,
"ms_cbra"          char(8) NULL ,
"ms_sdom"          char(16) NULL ,
"ms_sser"          char(20) NULL ,
"ms_ssig"          char(176) NULL ,
"ms_rdom"          char(16) NULL ,
"ms_rser"          char(20) NULL ,
"ms_rsig"          char(176) NULL ,
"ms_bcmt"          char(4) NULL ,
"ms_more"          char(1) NULL ,
"ms_text"          char(500) NULL ,
PRIMARY KEY ("ms_dbky"),
)

```

```

CREATE TABLE "DBA"."mx_"
(
    "mx_dbky"          char(18) NOT NULL ,
    "mx_sgmt"          char(2) NOT NULL ,
    "mx_bcmt"          char(4) NULL ,
    "mx_more"          char(1) NULL ,
    "mx_text"          char(500) NULL ,
    PRIMARY KEY ("mx_dbky", "mx_sgmt"),
)

```

```

CREATE TABLE "TS_DBA"."uauth"
(
    "uauth_id"          integer NOT NULL DEFAULT autoincrement ,
    "Ulogin"            varchar(30) NOT NULL ,

```

```

"pwd"          char(8) NOT NULL DEFAULT 'pwd' ,
"date_in"     char(8) NOT NULL DEFAULT 'NOW' ,
"empl_id"     integer NOT NULL ,
PRIMARY KEY ("uauth_id"),
)

```

```

CREATE TABLE "TS_DBA"."empl"

```

```

(
    "empl_id"      integer NOT NULL DEFAULT autoincrement ,
    "LName"        varchar(30) NOT NULL ,
    "FName"        varchar(30) NOT NULL ,
    "DataBirth"    char(8) NOT NULL DEFAULT 'NOW' ,
    PRIMARY KEY ("empl_id"),
)

```

2.4.2. Структура системи

Розроблена система взаємодіє з БД Sybase AnyWhere 17.0 за допомогою технології ADO.NET корпорації Microsoft, при цьому спроектовано структуру ПЗ, використовуючи ідеологію об'єктно-орієнтованого програмування. Структура програми приведена на рис. 2.2.

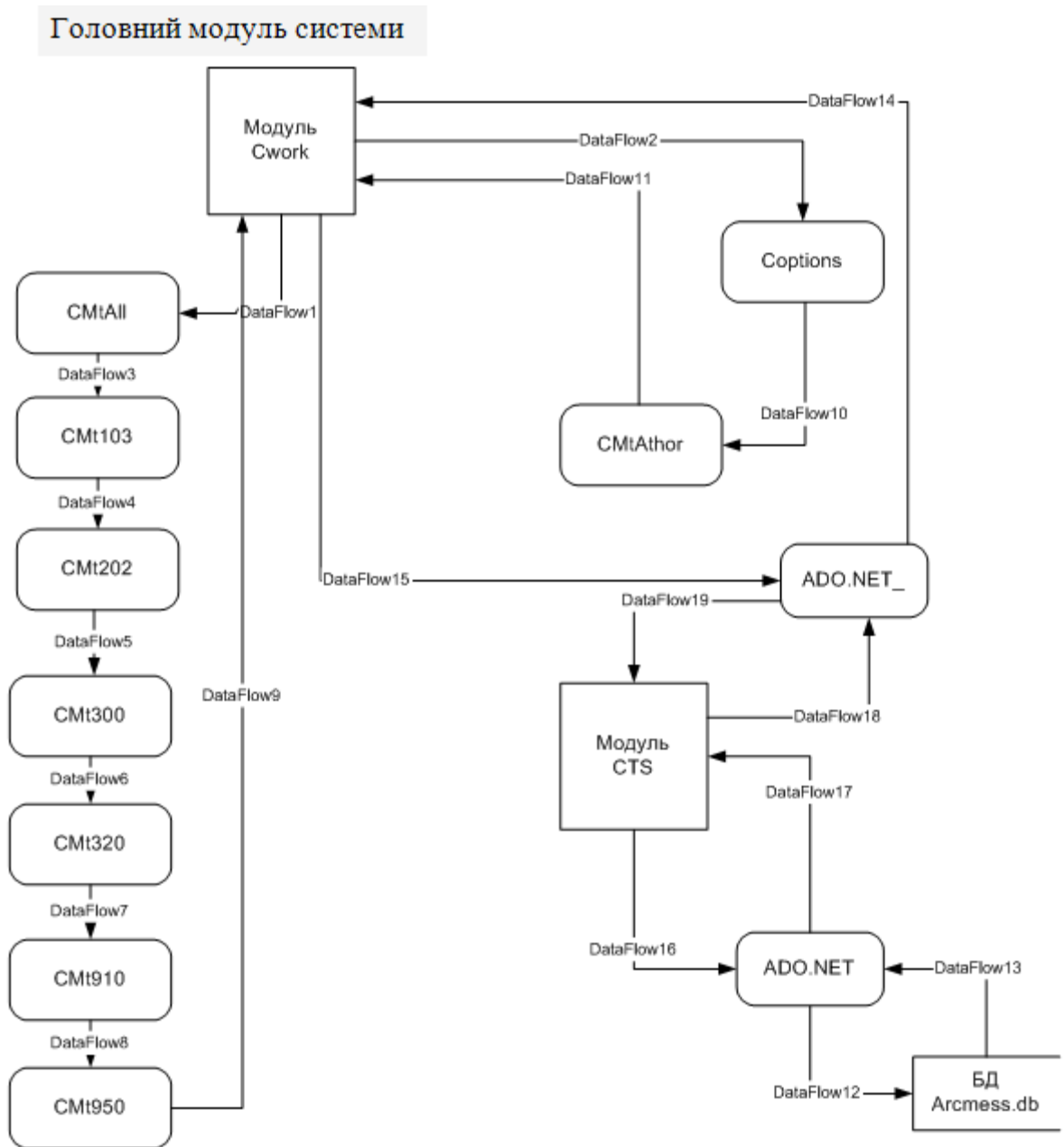


Рис. 2.2. Структура системи

2.4.3. Взаємодія з іншими компонентами ПЗ банку

Розроблена система обробляє інформацію, взаємодіючи при цьому з різними програмними модулями і системами. Для того, щоб здійснити аналіз даних, програма взаємодіє з базою архівних даних банку. Логічна схема взаємодії програм представлена на рис. 2.3.

Зв'язок додатку з іншими програмними модулями банку

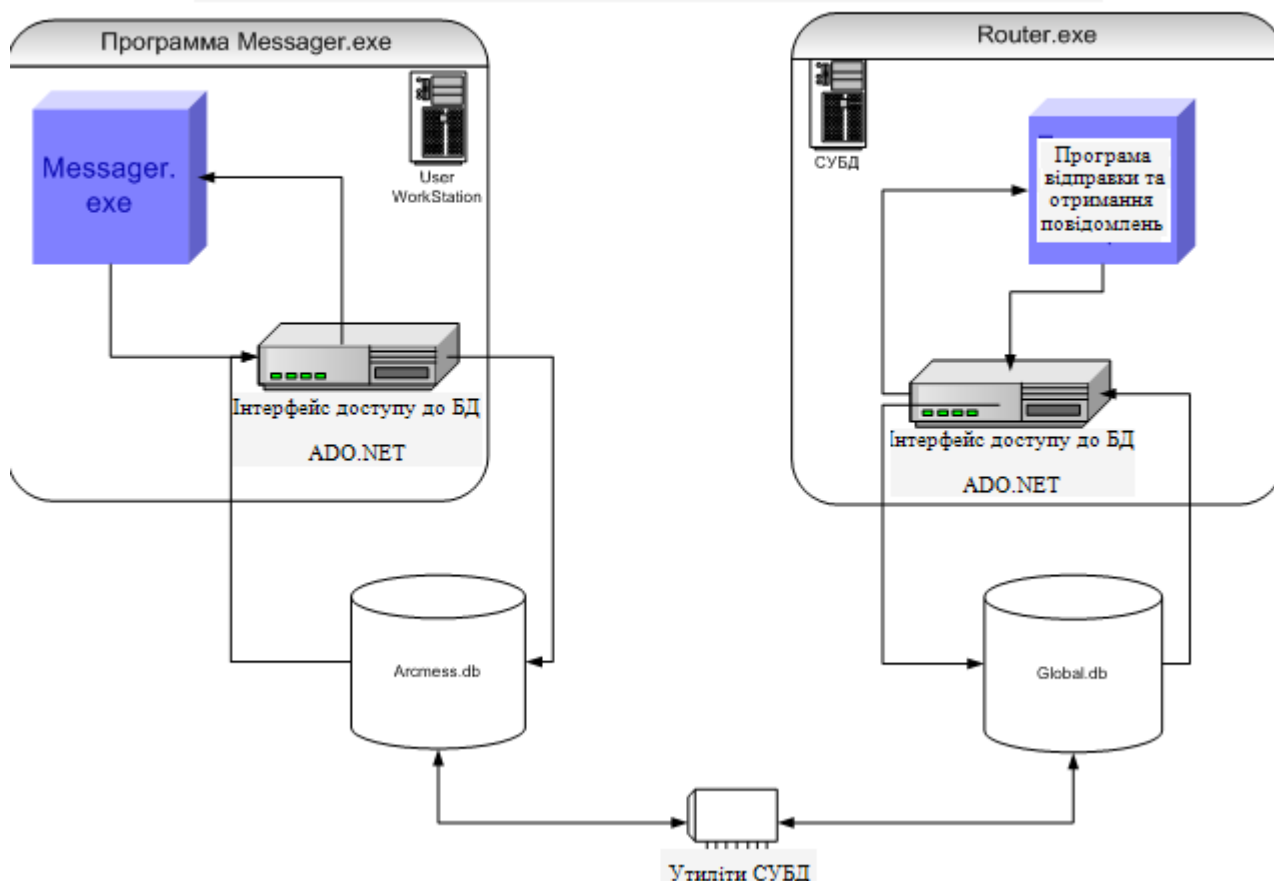


Рис. 2.3. Зв'язок додатку з іншими модулями

2.5. Обґрунтування та організація вхідних та вихідних даних програми

Для розроблюваної системи вхідними даними є набір отриманих критеріїв пошуку від оператора БД і дані, отримані за допомогою виконання внутрішнього SQL запиту до баз даних банківського сервера.

При проведенні пошуку, всі відсутні дані вводяться оператором в ручному режимі. Вихідні дані представлені в двох видах:

- електронному, у вигляді текстових файлів і звітів про проведену роботу;
- діалоговому, при ручному режимі проведення обробки.

Розроблена інформаційна система працює з локальною базою даних

inv_data.dbf під управлінням ОС Windows, версії не нижче 2000 і взаємодіє, за допомогою СУБД Sybase з базами даних DNH0, Syland, Sybridge, а також за допомогою файлового інтерфейсу з програмою TurboSwift.

2.6. Опис роботи розробленої системи

2.6.1. Використані технічні засоби

Для технічних засобів обрана конфігурація, що забезпечує цілодобову роботу програмних комплексів з резервуванням даних. Для цих засобів були обрані два сервера Dell PowerEdge 2600 Server, один з яких виконує функцію прийому і відправки повідомлень, а також здійснює всю постобробку повідомлень:

- аналіз повідомлень і маршрутизацію прийнятих повідомлень;
- аналіз і перевірку на наявність помилок відправлених банком SWIFT – повідомлень.

Другий сервер виконує функцію резервування даних і є копією першого.

Як було зазначено вище, розроблена система буде працювати спільно з програмним комплексом по відправки і прийому повідомлень, який на вимогу розробників, функціонує під ОС Windows 2016 Server, а база даних працює на Sybase SQL-Anywhere 17.0. Тому, інформаційна система обробки архівних повідомлень також працює під управлінням описаних вище програмних засобів. Технічні засоби були обрані, виходячи з необхідності забезпечити максимально відмовостійку систему для автоматизації процесу обробки банківських повідомлень. Для реалізації необхідної відмовостійкості, а також для забезпечення необхідної швидкості обробки отриманої інформації, були обрані два сервера Dell PowerEdge 2600 Server, один з яких виконує функцію резервування і даних.

2.6.2. Використані програмні засоби

Для реалізації бази даних, в якій буде зберігатися вся необхідна для роботи програми інформація, в даній роботі використана технологія роботи з БД на основі SQL-AnyWhere 17.0.

Для реалізації взаємозв'язку з БД банку використана технологія доступу до баз даних ADO.NET, використовуючи при цьому специфікацію мови SQL - Watcom.

Для реалізації клієнтської частини інформаційної системи використана середовище візуального програмування Microsoft Visual 2010.

2.6.3. Виклик та завантаження програми

Всі настройки програма, зберігає в файлі конфігурації * .ini, завдяки чому підвищується швидкість розгортання системи. Для коректної роботи інформаційної системи необхідна наявність все мережевих підключень і паролів, зазначених в ini. * файли, який знаходиться в кореневій папці проекту.

Запуск програми здійснюється або через меню «Пуск», «Програми» операційної системи, або шляхом активації .exe файлу програми.

2.6.4. Опис інтерфейсу користувача

Метою створення даної системи є автоматизація процесу пошуку інформації в архівах банку.

Для того, що б отримати доступ до програми необхідно пройти процедуру авторизації. Вид головного вікна програми під час першого запуску представлений на рис. 2.4.

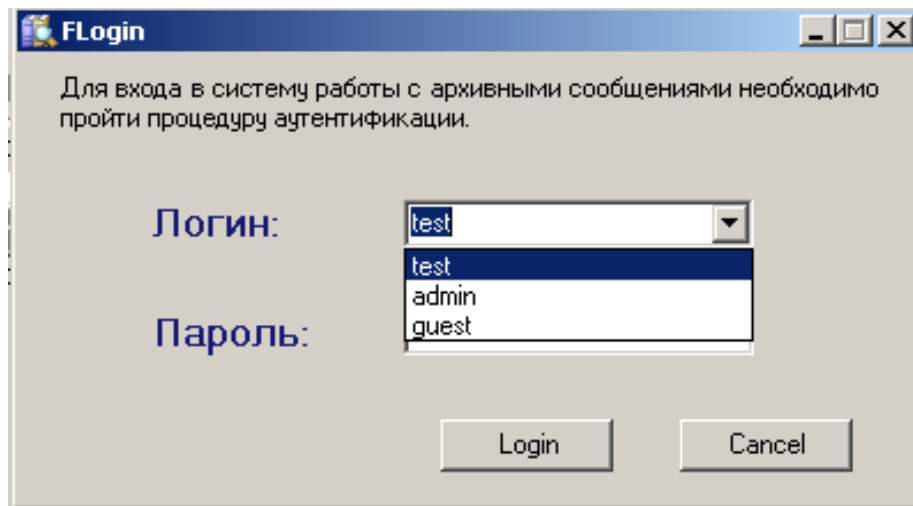


Рис. 2.4. Вікно аутентифікації.

Програма має досить багато типових повідомлень про помилки наступного характеру:

- повідомлення про відповідність вхідних даних необхідним форматам;
- про дублювання даних.

При неправильному введенні даних, система генерує повідомлення про помилку (рис. 2.5).



Рис. 2.5. Діалогове вікно помилки ідентифікації

Після проходження авторизації та успішної ідентифікації, оператору буде доступно головне меню програми, за допомогою якого здійснюється процес проведення розслідувань з нез'ясованих платіжними дорученнями (вказані ті функції, що розроблені в межах виконання даної кваліфікаційної роботи) (рис. 2.6).

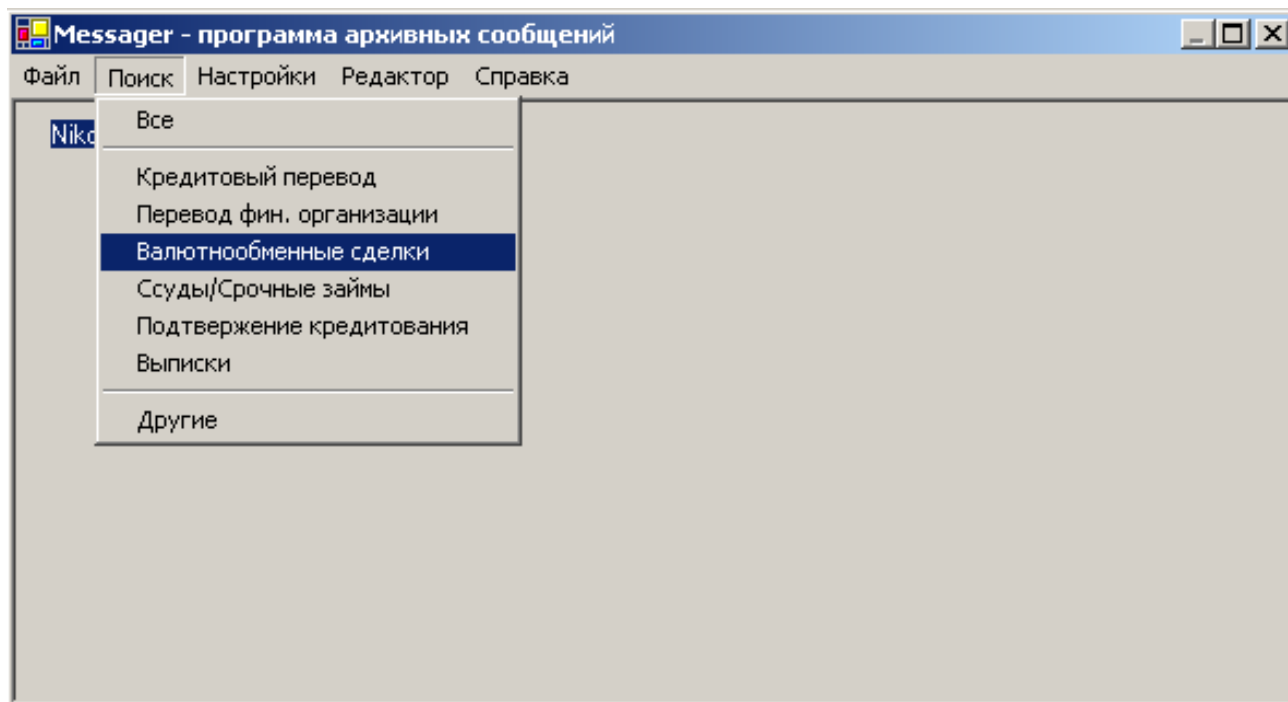


Рис. 2.6. Головне вікно програми.

Для того, щоб знайти архівне повідомлення необхідно скористатися головним меню програми та вибрати пункт меню «Валютні угоди».

Для організації пошуку платежу необхідно забезпечити введення ключових даних, за якими буде сформовано sql - запит в БД банку. Для початку проведення архівного пошуку документів, необхідно ввести ключові дані для пошуку в діалогове вікно «Налаштування пошуку» (рис. 2.7).

Настройки поиска.

Введите критерии поиска.

Начальная дата: Wednesday, June 03, 2021. Отправитель: PBANUA2X

Конечная дата: Tuesday, May 30, 2021. Получатель: DEUTDEFFXX

Направление: Сумма: 12345,56

Уникальный идентификатор:

Текст поиска:

Today

Search Cancel

Рис. 2.7. Форма введения параметров поиска документа

Після введення параметрів пошуку, на екрані монітора з'являється вікно результатів пошуку у форматі даних інтерфейсу TurboSWIFT (рис. 2.8) .

Окно результатов поиска.

!	ИО	Дата:	Тип:	Валюта:	Сумма:	Банк:	Файл:	Филиал:				
<input type="checkbox"/>	I	10.05.2021	103	USD	3 171.20	CHAS . US . 33 . XX	0510\$1J2.103		n	n	n	N
<input type="checkbox"/>	I	10.05.2021	103	USD	3 200.00	BKTR . US . 33 . XX	0510\$069.103		n	n	n	N
<input type="checkbox"/>	I	10.05.2021	103	USD	3 200.00	IRVT . US . 3N . XX	0510\$1LR.103		n	n	n	N
<input type="checkbox"/>	I	10.05.2021	103	USD	3 200.00	BKTR . US . 33 . XX	0510\$1MF.103		n	n	n	N
<input checked="" type="checkbox"/>	I	10.05.2021	910	USD		CHAS . US . 33 . XX	0510\$1MH.910		n	n	n	N
<input type="checkbox"/>	I	10.05.2021	103	USD		IRVT . US . 3N . XX	0510\$1P2.103		n	n	n	N
<input type="checkbox"/>	I	10.05.2021	910	USD		CHAS . US . 33 . XX	0510\$0SG.910		n	n	n	N
<input type="checkbox"/>	I	10.05.2021	103	USD		CHAS . US . 33 . XX	0510\$10A.103		n	n	n	N
<input type="checkbox"/>	I	10.05.2021	103	USD		CHAS . US . 33 . XX	0510\$1QK.103		n	n	n	N
<input type="checkbox"/>	I	10.05.2021	103	USD	3 300.00	CHAS . US . 33 . XX	0510\$04N.103		n	n	n	N
<input type="checkbox"/>	I	10.05.2021	103	USD	3 300.00	CHAS . US . 33 . XX	0510\$04O.103		n	n	n	N
<input type="checkbox"/>	I	10.05.2021	103	USD	3 300.00	CHAS . US . 33 . XX	0510\$04P.103		n	n	n	N
<input type="checkbox"/>	I	10.05.2021	103	USD	3 300.00	ABNA . GR . AP . XX	0510\$0LP.103		n	n	n	N
<input type="checkbox"/>	I	10.05.2021	910	USD	3 300.00	IRVT . US . 3N . XX	0510\$0JS.910		n	n	n	N
<input type="checkbox"/>	I	10.05.2021	910	USD	3 300.00	CHAS . US . 33 . XX	0510\$0LT.910		n	n	n	N
<input type="checkbox"/>	I	10.05.2021	103	USD	3 300.00	CHAS . US . 33 . XX	0510\$20G.103		n	n	n	N
<input type="checkbox"/>	I	10.05.2021	103	USD	3 311.52	CHAS . US . 33 . XX	0510\$0NF.103		n	n	n	N
<input type="checkbox"/>	I	10.05.2021	103	USD	3 319.00	CHAS . US . 33 . XX	0510\$1P9.103		n	n	n	N
<input type="checkbox"/>	I	10.05.2021	910	USD	3 322.38	CHAS . US . 33 . XX	0510\$0MO.910		n	n	n	N

Печать
Открыть
Сохранить
Сохранить Все

Рис. 2.8. Список найденных уведомлений в архиве банка

У файлового вигляді формуються повідомлення для подальшої подачі на друк. Вид обраного та сформованого за параметрами пошуку файл, наведено на рис. 2.9.

```

Окно результатов поиска.
-----
PRINTED BY TurboSwift AT:          15:38:32          FOR: PBANUA2XXXX
-----
Message Name: FREE FORMAT MESSAGE
Message Type: F 999      Sent/Received: DBXPAEADAXXX
Direction: Output      DUBAI BANK PJSC
Priority: Normal        DUBAI BANK BUILDING
Session: 1417          SHEIKH ZAYED STREET
Sequence: 432826      DUBAI
Input Output ACK   NAK   DUP   AUTH
Time   Time  Flag  Code  Flag  Code
MIR: 060417DBXPAEADAXXX1926086223   1638  1538  0      -----
MOR: 060417PBANUA2XXXX1417432826
MUR:
-----
1:      MESSAGE HEADER
        FO1PBANUA2XXXX1417432826
2:      APPLICATION HEADER
        O9991638060417DBXPAEADAXXX19260862230604171538N
4:      MESSAGE TEXT
:20:    Transaction Reference Number
TRN:                    FAX DT
:21:    Related Reference
Reference:              CASH ADVANCE.
:79:    Narrative
Free Format:           ATTN: CUSTOMER SUPPORT SERVICE
.
FURTHER TO YOUR FAX DATED 06.04.06 WITH REFERENCE
TO A CASH ADVANCE PAYMENT FOR YOUR CARD HOLDER
GEORGIY KHODACHENKO FOR USD 42000.
.
PLEASE BE ADVISED THAT WE DO NOT ACCEPT
  
```

Рис. 2.9. Результат відкриття знайденого за параметрами пошуку файлу

У діалоговому режимі виводиться інформація, що стосується розслідувань, що проводяться співробітниками відділу по нез'ясованим угодам.

Результат роботи представлений у вигляді діалогового вікна, в якому користувачеві представлена ключова інформація, яка однозначно ідентифікує SWIFT - повідомлення. Ескіз форми наведено на рис. 2.11.

USER	TYPE	CORRESPONDENT	TRN	VALUE	CUR	AMOUNT
PBANUA2X	IF950N	URCBUAUKXXXX	?0527/XR/0001	20050527	UAH	73667505,92
PBANUA2X	IF950N	DNSTUAUKXXXX	N0527/DT/0063	20050527	EUR	6972,19
PBANUA2X	IF950N	MPRIRUMDXXXX	?0527/PR/1001	20050527	UAH	3296994,77
PBANUA2X	IF950N	BLBBY2XXXX	?0527/BI/1001	20050527	UAH	136969,27
PBANUA2X	IF950N	PRTTLV22XXXX	N0527/PL/1007	20050527	RUB	20956,03
PBANUA2X	IF950N	PRTTLV22XXXX	N0527/PL/1063-1	20050527	EUR	86316,68
PBANUA2X	IF950N	PRTTLV22XXXX	N0527/PL/1063-2	20050527	EUR	20914,01
PBANUA2X	IF950N	MJSCUA22XXXX	?0527/MX/1002	20050527	UAH	373916,97
PBANUA2X	IF950N	CLHSUAUKXXXX	N0527/CL/1004	20050527	USD	12190,97
PBANUA2X	IF950N	EXMMMD22XXXX	?0527/EX/2003	20050527	USD	18017,84
PBANUA2X	IF950N	BSOCMD2XXXX	+0527/SO/2001	20050527	UAH	91446,31
PBANUA2X	IF950N	AVALUAUKXXXX	N0527/AV/2063	20050527	EUR	99944,43
PBANUA2X	IF950N	KUPBUAUKXXXX	N0527/PM/2063	20050527	EUR	63539,73
PBANUA2X	IF950N	UKTCUA2KXXXX	?0527/TK/2006	20050527	UAH	165316,61
PBANUA2X	IF950N	DNSTUAUKXXXX	N0527/DT/2072	20050527	EUR	0,00
PBANUA2X	IF950N	DJSBUA2XXXX	?0527/DO/2002	20050527	UAH	25428,57
PBANUA2X	IF950N	MOBBMD22XXXX	?0527/MB/3001	20050527	UAH	164365,15
PBANUA2X	IF950N	AKBBY2XXXX	?0527/AY/3001	20050527	UAH	1683127,13
PBANUA2X	IF950N	RSLBRUMDXXXX	?0527/RS/3001	20050527	UAH	4503,03
PBANUA2X	IF950N	PRTTLV22XXXX	N0527/PL/3072	20050527	EUR	34785,25
PBANUA2X	IF950N	ITGRUAUKXXXX	N0527/IG/3004	20050527	USD	72704,59
PBANUA2X	IF950N	UKTCUA2KXXXX	N0527/TK/3072	20050527	EUR	8804,65
PBANUA2X	IF950N	ELECUA2XXXX	N0527/EB/3072	20050527	EUR	32981,12
PBANUA2X	IF950N	EABUUAUKXXXX	N0527/EZ/3004	20050527	USD	121822,19
PBANUA2X	IF950N	UKCAUAUKXXXX	N0527/KU/3004	20050527	USD	86914,24

Рис. 2.11. Діалогове вікно результатів пошуку.

Також, в файловому вигляді формується інформація, отримана з розслідувань. В отриманому файлі, надаються довідкові дані про шукані SWIFT - повідомлення. Ескіз файлу наведено на рис. 2.10.

mt191rep - Notepad			
File Edit Format View Help			
BNLIITRRXXXX	:20:CHRG041801F	:21:38488053810040EA	:32B:EUR20.00
COBADEFFXXXX	:20:CHRG041801X	:21:FAAS510452512100	:32B:EUR20.00
BKTRUS33XXXX	:20:CHRG041801Y	:21:C7116400CP041905	:32B:USD20.00
ABNANL2AXXXX	:20:CHRG041805L	:21:EM1504071201788I	:32B:EUR20.00
CHASUS33XXXX	:20:CHRG041805Q	:21:9962100104JS	:32B:USD20.00
BKTRUS33XXXX	:20:CHRG0418072	:21:C701547BBK041805	:32B:USD20.00
BKTRUS33XXXX	:20:CHRG0418076	:21:C707939BBK041805	:32B:USD20.00
BKTRUS33XXXX	:20:CHRG0418078	:21:C652548BBK041805	:32B:USD20.00
DEUTDEFFXXXX	:20:CHRG041808D	:21:001KID0504144937	:32B:EUR20.00
GIBAATWXXXX	:20:CHRG041808L	:21:20050418PAY00359	:32B:EUR20.00
UBSWCHZH80A	:20:CHRG041808Z	:21:ZD81105E12551346	:32B:USD20.00
DEUTDEFFXXXX	:20:CHRG0418099	:21:001KID0504150679	:32B:EUR20.00
DEUTDEFFXXXX	:20:CHRG041809P	:21:004KID0504144514	:32B:EUR20.00
DEUTDEFFXXXX	:20:CHRG041809T	:21:004KID0504144687	:32B:EUR20.00
DEUTDEFFXXXX	:20:CHRG041809U	:21:004KID0504144690	:32B:EUR20.00
DEUTDEFFXXXX	:20:CHRG04180AD	:21:004KID0504154409	:32B:EUR20.00
DEUTDEFFXXXX	:20:CHRG04180BN	:21:002KID0504180295	:32B:EUR20.00
DEUTDEFFXXXX	:20:CHRG04180B0	:21:002KID0504180316	:32B:EUR20.00
DEUTDEFFXXXX	:20:CHRG04180BR	:21:002KID0504180369	:32B:EUR20.00
DEUTDEFFXXXX	:20:CHRG04180C0	:21:002KID0504180287	:32B:EUR20.00
CHASUS33XXXX	:20:CHRG04180CQ	:21:4807700108JS	:32B:USD20.00
MULTLV2XXXX	:20:CHRG04180CR	:21:20050415114006	:32B:EUR20.00
BSUICH22X120	:20:CHRG04180D5	:21:SW0762805000-MYL	:32B:USD20.00
COBADEFFXXXX	:20:CHRG04180D7	:21:FAAS510856380200	:32B:EUR20.00
HABAE2XXXX	:20:CHRG04180D9	:21:050417000657	:32B:USD20.00
RIKOLV2XXXX	:20:CHRG04180DP	:21:RMZ706808J-S	:32B:USD20.00
BBRUBEBB010	:20:CHRG04180DW	:21:H9050418K005480	:32B:EUR20.00
BBRUBEBB010	:20:CHRG04180DX	:21:H01504186105890	:32B:EUR20.00
HABAE2XXXX	:20:CHRG04180EF	:21:050416000980	:32B:EUR20.00
DEUTDEFFXXXX	:20:CHRG04180EN	:21:001KID0504181783	:32B:EUR20.00
COBADEFFXXXX	:20:CHRG04180ES	:21:FAAS510555678000	:32B:USD20.00
DEUTDEFFXXXX	:20:CHRG04180FA	:21:106KID0504180073	:32B:EUR20.00
COBADEFFXXXX	:20:CHRG04180FK	:21:FAAS510555076900	:32B:EUR20.00
GEBABEBB36A	:20:CHRG04180FM	:21:2005041800010251	:32B:EUR20.00
ABNANL2AXXXX	:20:CHRG04180FO	:21:XU18040031TD	:32B:EUR20.00
BCOMPTPLXXXX	:20:CHRG04180G9	:21:OEM0200521691242	:32B:EUR20.00
DEUTPLPKXXXX	:20:CHRG04180GG	:21:6408BT050418Z12	:32B:USD20.00
MKKBHUNBXXXX	:20:CHRG04180GV	:21:FT05108012000291	:32B:USD20.00
CHASDEFXXXX	:20:CHRG04180GZ	:21:2059737 80131	:32B:EUR20.00

Рис. 2.10. Файл, що містить результати пошуку

РОЗДІЛ 3

ЕКОНОМІЧНИЙ РОЗДІЛ

3.1. Розрахунок трудомісткості та вартості розробки програмного продукту

Вхідні дані:

- передбачуване число операторів – 1500;
- коефіцієнт складності програми – 2;
- коефіцієнт корекції програми в ході її розробки – 0,08;
- годинна заробітна плата програміста, грн / год – 50.

Нормування праці в процесі створення ПЗ істотно ускладнено в силу творчого характеру праці програміста. Тому трудомісткість розробки ПЗ може бути розрахована на основі системи моделей з різною точністю оцінки.

Трудомісткість розробки ПЗ можна розрахувати за формулою:

$$t = t_u + t_a + t_n + t_{oml} + t_d, \text{ людино-годин,} \quad (3.1)$$

де t_o - витрати праці на підготовку й опис поставленої задачі (приймається 50),

t_u - витрати праці на дослідження алгоритму рішення задачі,

t_a - витрати праці на розробку блок-схеми алгоритму,

t_n - витрати праці на програмування по готовій блок-схемі,

t_{oml} - витрати праці на налагодження програми на ЕОМ,

t_d - витрати праці на підготовку документації.

Складові витрати праці визначаються через умовне число операторів у ПЗ, яке розробляється.

Умовне число операторів (підпрограм):

$$Q = q \times C \times (1 + p), \text{ людино-годин,} \quad (3.2)$$

де q - передбачуване число операторів,

C - коефіцієнт складності програми,

p - коефіцієнт кореляції програми в ході її розробки.

$$Q = 1500 \cdot 2 \cdot (1 + 0,08) = 3240 \text{ людино-годин.}$$

Витрати праці на вивчення опису задачі ти визначається з урахуванням уточнення опису і кваліфікації програміста:

$$t_u = \frac{QB}{(75...85)K}, \text{ людино-годин,} \quad (3.3)$$

де B - коефіцієнт збільшення витрат праці внаслідок недостатнього опису задачі; $B=1.2 \dots 1.5$,

k - коефіцієнт кваліфікації програміста, обумовлений від стажу роботи з даної спеціальності.

Витрати праці на розробку алгоритму рішення задачі:

$$t_u = \frac{3240 \cdot 1,3}{80 \cdot 1,2} = 44, \text{ людино-годин.}$$

Витрати на складання програми по готовій блок-схемі:

$$t_a = \frac{Q}{(20...25)K} \text{ людино-годин.} \quad (3.4)$$

$$t_a = \frac{3240}{22 \cdot 1,2} = 123 \text{ людино-годин.}$$

Витрати на складання програми по готовій блок-схемі:

$$t_n = \frac{Q}{(20..25)K} \quad \text{людино-годин.} \quad (3.5)$$

$$t_n = \frac{3240}{23 \cdot 1,2} = 117 \quad \text{людино-годин.}$$

Витрати праці на налагодження програми на ЕОМ:

$$t_{\text{отп}} = \frac{Q}{(4..5)K} \quad \text{людино-годин.} \quad (3.6)$$

$$t_{\text{отп}} = \frac{3240}{5 \cdot 1,2} = 540 \quad \text{людино-годин.}$$

Витрати праці на підготовку документації:

$$t_{\partial} = t_{\partial p} + t_{\partial o}, \quad \text{людино-годин,} \quad (3.7)$$

де $t_{\partial p}$ - трудомісткість підготовки матеріалів і рукопису.

$$t_{\partial p} = \frac{Q}{(15..20)K}, \quad \text{людино-годин.} \quad (3.8)$$

$$t_{\partial p} = \frac{3240}{18 \cdot 1,2} = 150 \quad \text{людино-годин,}$$

$t_{\partial o}$ - трудомісткість редагування, печатки й оформлення документації

$$t_{\partial o} = 0,75 \cdot t_{\partial p}, \quad \text{людино-годин.} \quad (3.9)$$

$$t_{\partial o} = 150 + 73 = 223 \quad \text{людино-годин.}$$

Отримуємо трудомісткість розробки програмного забезпечення:

$$t = 50 + 44 + 123 + 117 + 540 + 223 = 1097 \text{ людино-годин.}$$

3.2. Розрахунок витрат на створення програми

Витрати на створення ПЗ Кпо включають витрати на заробітну плату виконавця програми Зз/п і витрат машинного часу, необхідного на налагодження програми на ЕОМ.

$$K_{по} = З_{зп} + З_{мв}, \text{ грн,} \quad (3.10)$$

Заробітна плата виконавців визначається за формулою:

$$З_{зп} = t \cdot C_{пр}, \text{ грн,} \quad (3.11)$$

де t - загальна трудомісткість, людино-годин,

$C_{пр}$ - середня годинна заробітна плата програміста, грн/година.

$$C_{пр} = 1097 \cdot 50 = 54850 \text{ грн.}$$

Вартість машинного часу, необхідного для налагодження програми на ЕОМ:

$$З_{мв} = t_{отл} \times C_{м}, \text{ грн,} \quad (3.12)$$

де $t_{отл}$ - трудомісткість налагодження програми на ЕОМ, год,

$C_{м}$ - вартість машино-години ЕОМ, грн/год.

Визначені в такий спосіб витрати на створення програмного забезпечення є частиною одноразових капітальних витрат на створення АСУП.

$$З_{мв} = 540 \times 5 = 2700 \text{ грн.}$$

$$\hat{E}_{\text{п}} = 2700 + 54850 = 57550 \text{ }.$$

Очкуваний перод створення ПЗ:

$$T = \frac{t}{B_k \cdot F_p} \text{ мс.} \quad (3.13)$$

де B_k - число виконавцв,

F_p - мсячний фонд робочого часу (при 40 годинному робочому тижни $F_p=176$ годин).

$$T = \frac{1097}{1 \cdot 176} = 6.23 \text{ мс.}$$

Висновки: визначено трудомстксть розробленого додатку (1097 люд-год), проведений драхунок вартости роботи по створенню програми (57550 грн.) та розраховано час на його створення (6,2 мс).

ВИСНОВКИ

Об'єктом розробки даної кваліфікаційної роботи є автоматизована система обробки архівних повідомлень банку.

Метою роботи є автоматизація процесу збору інформації про роботу банку, формування звітів, формування необхідних документів і звітів для проведення розслідувань та інших операцій з цінними паперами.

Призначенням даного програмного комплексу є формування коректного повідомлення з архівної бази по заданому оператором критерію пошуку. При цьому програма Messenger працює з усіма типами фінансових повідомлень, оброблюваних банком.

В даній системі реалізована можливість конкретизації пошуку повідомлень по їх типам, датою отримання/відправки, напрямку прямування, а також за сумою платежу, датою валютування з позначенням як відправника, так і одержувача повідомлення.

Для вирішення заданої мети під час розробки системи забезпечено автоматичне формування повідомлень про роботу банку з архівних повідомлень, керуючись для пошуку різного роду даними, що вводяться оператором, а також автоматизовано процес отримання всіх необхідних платежів з банківської СУБД.

При розробці інформаційної системи автоматизовано процес формування різного роду звітів про фінансову діяльність банку.

Розроблена система буде працювати спільно з програмним комплексом по відправки і прийому повідомлень, який на вимогу розробників, функціонує під ОС Windows 2016 Server, а база даних працює на Sybase SQL-Anywhere 17.0. Тому, інформаційна система обробки архівних повідомлень також працює під управлінням описаних вище програмних засобів. Технічні засоби були обрані, виходячи з необхідності забезпечити максимально відмовостійку систему для автоматизації процесу обробки банківських повідомлень.

Для реалізації бази даних, в якій зберігається вся необхідна для роботи

програми інформація, в даній роботі використана технологія роботи з БД на основі SQL-AnyWhere 17.0.

Для реалізації взаємозв'язку з БД банку використана технологія доступу до баз даних ADO.NET, використовуючи при цьому специфікацію мови SQL - Watcom.

Для реалізації клієнтської частини інформаційної системи використана середовище візуального програмування Microsoft Visual 2010.

В економічному розділі визначено трудомісткість розробленого додатку (1097 люд-год), проведений підрахунок вартості роботи по створенню програми (57550 грн.) та розраховано час на його створення (6,2 міс).

СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ

1. Ананьєв, О.М. Інформаційні системи і технології в комерційній діяльності [Текст]: підручник / О.М. Ананьєв, В.М. Білик, Я.А. Гончарук. – Львів: Новий Світ-2000, 2006. – 584 с.
2. Антонов, В.М. Фінансовий менеджмент: сучасні інформаційні технології [Текст]: навчальний посібник / В.М. Антонов, Г.К. Яловий; ред. В.М. Антонов; Мін-во освіти і науки України, КНУ ім. Т.Г. Шевченка. – К.: ЦНЛ, 2005. – 432 с.
3. Арбузов С. Г., Колобов Ю. В., Міщенко В. І., Науменкова С. В. СВІФТ // Банківська енциклопедія. - Київ : Центр наукових досліджень Національного банку України : Знання, 2011. - 504 с. - (Інституційні засади розвитку банківської системи України). - ISBN 978-966-346-923-2.
4. Архангельский А.Я. С++ Builder 6. Справочное пособие. Книга 1. Язык С++. – М.: Бином-Пресс, 2002 г. – 544 с.: ил. ISBN 978-5-9518-0229-3.
5. Бусигін Б.С., Коротенко Г.М., Коротенко Л.М. Прикладна інформатика. Підручник для студентів комп'ютерних спеціальностей. – Дніпропетровськ: Видавництво НГУ, 2004. – 559 с. URL: <http://www.programmer.dp.ua/book-ua-k01.php>. дата звернення: 15.03.2019.
6. Бусыгин Б.С., Дивизинюк М.М., Коротенко Г.М., Коротенко Л.М. Введение в современную информатику. Учебник. – Севастополь: Издательство СНУЯЭиП, 2005. – 644 с. / URL: <http://www.programmer.dp.ua/book-ru-k02.php>. дата звернення: 15.01.2018.
7. Гужва В.М. Інформаційні системи і технології на підприємствах [Текст]: навчальний посібник / В.М. Гужва; Мін-во освіти і науки України, КНЕУ. – К.: КНЕУ, 2001. – 400 с.
9. Гуржій, А.М. Інформатика та інформаційні технології [Текст]: підручник / А.М. Гуржій, Н.І. Поворознюк, В.В. Самсонов. – Х.: Компанія СМІТ, 2003. – 352 с.

10. Давыдов В.Г. Технологии программирования C++. –СПб.:БХВ-Петербург, 2005.-672с.:ил.
11. ДСТУ 2394-94 Інформація та документація. Комплектування фонду, бібліографічний опис, аналіз документів. Терміни та визначення. – Чинний від 01.01.1995. - Київ : Держстандарт України, 1994. – 88 с.
12. Маслов, В.П. Інформаційні системи і технології в економіці [Текст]: навчальний посібник / В.П. Маслов; Мін-во освіти і науки України. – К.: Слово, 2003. – 264 с.
13. Методичні вказівки з виконання економічного розділу в дипломних проектах студентів спеціальності “Комп’ютерні системи ” / Укладачі О.Г. Вагонова, Нікітіна О.Б. Н.Н. Романюк – Дніпропетровськ: Національний гірничий університет. – 2013. – 23с.
14. Методичні рекомендації до виконання кваліфікаційних робіт бакалаврів напряму підготовки 122 «Комп’ютерні науки» галузі знань 12 Інформаційні технології/, Л.М. Коротенко , О.С. Шевцова; Нац. гірн. ун-т. – Д : ДВНЗ НГУ, 2019. – 65 с.
15. Олійник, А.В. Інформаційні системи і технології у фінансових установах [Текст]: навчальний посібник / А.В. Олійник, В.М. Шацька. – Львів: Новий Світ-2000, 2006. – 436 с.
16. Офіційний сайт середовища розробки Visual Studio Code URL: <https://code.visualstudio.com/docs>. дата звернення: 12.05.2021.
17. Пасічник, В.В. Організація баз даних та знань/ В.В.Пасічник, В.А.Резніченко.–К.: Видавнича група ВНУ, 2006.– 384 с.:іл.– ISBN966-552-156-X
18. Румянцев, М.И. Информационные системы и технологии финансово-кредитных учреждений [Текст]: учебное пособие для вузов / М.И. Румянцев; Западодонбасский ин-т экономики и управления. – Днепропетровск: ИМА-пресс, 2006. – 482 с.

19. Системы и технологии: приложения в экономике и управлении: Кн. 6 [Текст]: учебное пособие / Мин-во образования и науки Украины, Донецкий нац. ун-т; ред. Ю.Г. Лысенко. – Донецк: Юго-Восток, 2004. – 377 с.

20. Степанов Ю.Л. Разработка приложений баз данных для СУБД SYBASE SQL ANYWHERE.

21. Скорочення слів в українській мові у бібліографічному описі. Загальні правила та вимоги : ДСТУ 3582-97. – Чинний від 1998-07-01. – К. : Держстандарт України, 1998. – 24 с. – (Державний стандарт України).

КОД ПРОГРАМИ

```
using System;
using System.Drawing;
using System.Collections;
using System.ComponentModel;
using System.Windows.Forms;
using System.Text;
using iAnywhere.Data.AsaClient;
namespace Messenger
{
    /// <summary>
    /// Summary description for FLogin.
    /// </summary>
    public class FLogin : System.Windows.Forms.Form
    {
        private System.Windows.Forms.Label label1;
        private System.Windows.Forms.Label label2;
        private System.Windows.Forms.Label label3;
        private System.Windows.Forms.TextBox txtBPwd;
        private System.Windows.Forms.Button btnCancel;
        private System.Windows.Forms.Button btnOk;
        /// <summary>
        /// Required designer variable.
        /// </summary>
        private System.ComponentModel.Container components = null;

        //Anywhere data types
        private AsaCommand asa_cmd;
        private AsaConnection asa_conn;
        private AsaDataAdapter asa_adapter;
        //Works elements
        private string str_conn;
        private string str_cmd;
        private string[] str_pwd;
        private System.Windows.Forms.ComboBox cbLogin;
        private bool flag = true;
        private StringBuilder str_info;
        public FLogin(StringBuilder str_info)
        {
            //
            // Required for Windows Form Designer support
            //
            InitializeComponent();
            //
            cbLogin.Focus();
            this.str_info =str_info;
            str_conn = "Data Source=TS;UID=TS_DBA;PWD=TS_DBA";
        }
    }
}
```

```

        flag=false;
        //
    }
    /// <summary>
    /// Clean up any resources being used.
    /// </summary>
    protected override void Dispose( bool disposing )
    {
        if( disposing )
        {
            if(components != null)
            {
                components.Dispose();
            }
        }
        base.Dispose( disposing );
    }
    #region Windows Form Designer generated code
    /// <summary>
    /// Required method for Designer support - do not modify
    /// the contents of this method with the code editor.
    /// </summary>
    private void InitializeComponent()
    {
        System.Resources.ResourceManager resources = new
System.Resources.ResourceManager(typeof(FLogin));
        this.label1 = new System.Windows.Forms.Label();
        this.label2 = new System.Windows.Forms.Label();
        this.label3 = new System.Windows.Forms.Label();
        this.txtBPwd = new System.Windows.Forms.TextBox();
        this.btnCanc = new System.Windows.Forms.Button();
        this.btnOk = new System.Windows.Forms.Button();
        this.cbLogin = new System.Windows.Forms.ComboBox();
        this.SuspendLayout();
        //
        // label1
        //
        this.label1.AccessibleDescription =
resources.GetString("label1.AccessibleDescription");
        this.label1.AccessibleName = resources.GetString("label1.AccessibleName");
        this.label1.Anchor =
((System.Windows.Forms.AnchorStyles)(resources.GetObject("label1.Anchor")));
        this.label1.AutoSize = ((bool)(resources.GetObject("label1.AutoSize")));
        this.label1.Dock =
((System.Windows.Forms.DockStyle)(resources.GetObject("label1.Dock")));
        this.label1.Enabled = ((bool)(resources.GetObject("label1.Enabled")));
        this.label1.Font = ((System.Drawing.Font)(resources.GetObject("label1.Font")));
        this.label1.Image =
((System.Drawing.Image)(resources.GetObject("label1.Image")));
        this.label1.ImageAlign =
((System.Drawing.ContentAlignment)(resources.GetObject("label1.ImageAlign")));
        this.label1.ImageIndex = ((int)(resources.GetObject("label1.ImageIndex")));

```

```

        this.label1.ImeMode =
((System.Windows.Forms.ImeMode)(resources.GetObject("label1.ImeMode")));
        this.label1.Location =
((System.Drawing.Point)(resources.GetObject("label1.Location")));
        this.label1.Name = "label1";
        this.label1.RightToLeft =
((System.Windows.Forms.RightToLeft)(resources.GetObject("label1.RightToLeft")));
        this.label1.Size = ((System.Drawing.Size)(resources.GetObject("label1.Size")));
        this.label1.TabIndex = ((int)(resources.GetObject("label1.TabIndex")));
        this.label1.Text = resources.GetString("label1.Text");
        this.label1.TextAlign =
((System.Drawing.ContentAlignment)(resources.GetObject("label1.TextAlign")));
        this.label1.Visible = ((bool)(resources.GetObject("label1.Visible")));
        //
        // label2
        //
        this.label2.AccessibleDescription = resources.GetString("label2.AccessibleDescription");
        this.label2.AccessibleName = resources.GetString("label2.AccessibleName");
        this.label2.Anchor =
((System.Windows.Forms.AnchorStyles)(resources.GetObject("label2.Anchor")));
        this.label2.AutoSize = ((bool)(resources.GetObject("label2.AutoSize")));
        this.label2.Dock =
((System.Windows.Forms.DockStyle)(resources.GetObject("label2.Dock")));
        this.label2.Enabled = ((bool)(resources.GetObject("label2.Enabled")));
        this.label2.Font = ((System.Drawing.Font)(resources.GetObject("label2.Font")));
        this.label2.ForeColor = System.Drawing.Color.MidnightBlue;
        this.label2.Image =
((System.Drawing.Image)(resources.GetObject("label2.Image")));
        this.label2.ImageAlign =
((System.Drawing.ContentAlignment)(resources.GetObject("label2.ImageAlign")));
        this.label2.ImageIndex = ((int)(resources.GetObject("label2.ImageIndex")));
        this.label2.ImeMode =
((System.Windows.Forms.ImeMode)(resources.GetObject("label2.ImeMode")));
        this.label2.Location =
((System.Drawing.Point)(resources.GetObject("label2.Location")));
        this.label2.Name = "label2";
        this.label2.RightToLeft =
((System.Windows.Forms.RightToLeft)(resources.GetObject("label2.RightToLeft")));
        this.label2.Size = ((System.Drawing.Size)(resources.GetObject("label2.Size")));
        this.label2.TabIndex = ((int)(resources.GetObject("label2.TabIndex")));
        this.label2.Text = resources.GetString("label2.Text");
        this.label2.TextAlign =
((System.Drawing.ContentAlignment)(resources.GetObject("label2.TextAlign")));
        this.label2.Visible = ((bool)(resources.GetObject("label2.Visible")));
        //
        // label3
        //
        this.label3.AccessibleDescription = resources.GetString("label3.AccessibleDescription");
        this.label3.AccessibleName = resources.GetString("label3.AccessibleName");
        this.label3.Anchor =
((System.Windows.Forms.AnchorStyles)(resources.GetObject("label3.Anchor")));
        this.label3.AutoSize = ((bool)(resources.GetObject("label3.AutoSize")));

```

```

        this.label3.Dock =
((System.Windows.Forms.DockStyle)(resources.GetObject("label3.Dock")));
        this.label3.Enabled = ((bool)(resources.GetObject("label3.Enabled")));
        this.label3.Font = ((System.Drawing.Font)(resources.GetObject("label3.Font")));
        this.label3.ForeColor = System.Drawing.Color.MidnightBlue;
        this.label3.Image =
((System.Drawing.Image)(resources.GetObject("label3.Image")));
        this.label3.ImageAlign =
((System.Drawing.ContentAlignment)(resources.GetObject("label3.ImageAlign")));
        this.label3.ImageIndex = ((int)(resources.GetObject("label3.ImageIndex")));
        this.label3.ImeMode =
((System.Windows.Forms.ImeMode)(resources.GetObject("label3.ImeMode")));
        this.label3.Location =
((System.Drawing.Point)(resources.GetObject("label3.Location")));
        this.label3.Name = "label3";
        this.label3.RightToLeft =
((System.Windows.Forms.RightToLeft)(resources.GetObject("label3.RightToLeft")));
        this.label3.Size = ((System.Drawing.Size)(resources.GetObject("label3.Size")));
        this.label3.TabIndex = ((int)(resources.GetObject("label3.TabIndex")));
        this.label3.Text = resources.GetString("label3.Text");
        this.label3.TextAlign =
((System.Drawing.ContentAlignment)(resources.GetObject("label3.TextAlign")));
        this.label3.Visible = ((bool)(resources.GetObject("label3.Visible")));
        //
        // txtBPwd
        //
        this.txtBPwd.AccessibleDescription =
resources.GetString("txtBPwd.AccessibleDescription");
        this.txtBPwd.AccessibleName = resources.GetString("txtBPwd.AccessibleName");
        this.txtBPwd.Anchor =
((System.Windows.Forms.AnchorStyles)(resources.GetObject("txtBPwd.Anchor")));
        this.txtBPwd.AutoSize = ((bool)(resources.GetObject("txtBPwd.AutoSize")));
        this.txtBPwd.BackgroundImage =
((System.Drawing.Image)(resources.GetObject("txtBPwd.BackgroundImage")));
        this.txtBPwd.Dock =
((System.Windows.Forms.DockStyle)(resources.GetObject("txtBPwd.Dock")));
        this.txtBPwd.Enabled = ((bool)(resources.GetObject("txtBPwd.Enabled")));
        this.txtBPwd.Font =
((System.Drawing.Font)(resources.GetObject("txtBPwd.Font")));
        this.txtBPwd.ImeMode =
((System.Windows.Forms.ImeMode)(resources.GetObject("txtBPwd.ImeMode")));
        this.txtBPwd.Location =
((System.Drawing.Point)(resources.GetObject("txtBPwd.Location")));
        this.txtBPwd.MaxLength = ((int)(resources.GetObject("txtBPwd.MaxLength")));
        this.txtBPwd.Multiline = ((bool)(resources.GetObject("txtBPwd.Multiline")));
        this.txtBPwd.Name = "txtBPwd";
        this.txtBPwd.PasswordChar =
((char)(resources.GetObject("txtBPwd.PasswordChar")));
        this.txtBPwd.RightToLeft =
((System.Windows.Forms.RightToLeft)(resources.GetObject("txtBPwd.RightToLeft")));
        this.txtBPwd.ScrollBars =
((System.Windows.Forms.ScrollBars)(resources.GetObject("txtBPwd.ScrollBars")));

```

```

        this.txtBPwd.Size =
((System.Drawing.Size)(resources.GetObject("txtBPwd.Size")));
        this.txtBPwd.TabIndex = ((int)(resources.GetObject("txtBPwd.TabIndex")));
        this.txtBPwd.Text = resources.GetString("txtBPwd.Text");
        this.txtBPwd.TextAlign =
((System.Windows.Forms.HorizontalAlignment)(resources.GetObject("txtBPwd.TextAlign")));
        this.txtBPwd.Visible = ((bool)(resources.GetObject("txtBPwd.Visible")));
        this.txtBPwd.WordWrap = ((bool)(resources.GetObject("txtBPwd.WordWrap")));
        this.txtBPwd.KeyPress += new
System.Windows.Forms.KeyPressEventHandler(this.txtBPwd_KeyPress);
        this.txtBPwd.TextChanged += new
System.EventHandler(this.txtBPwd_TextChanged);
        this.txtBPwd.KeyUp += new
System.Windows.Forms.KeyEventHandler(this.txtBPwd_KeyUp);
        //
        // btnCancel
        //
        this.btnCancel.AccessibleDescription = resources.GetString("btnCancel.AccessibleDescription");
        this.btnCancel.AccessibleName = resources.GetString("btnCancel.AccessibleName");
        this.btnCancel.Anchor =
((System.Windows.Forms.AnchorStyles)(resources.GetObject("btnCancel.Anchor")));
        this.btnCancel.BackgroundImage =
((System.Drawing.Image)(resources.GetObject("btnCancel.BackgroundImage")));
        this.btnCancel.Dock =
((System.Windows.Forms.DockStyle)(resources.GetObject("btnCancel.Dock")));
        this.btnCancel.Enabled = ((bool)(resources.GetObject("btnCancel.Enabled")));
        this.btnCancel.FlatStyle =
((System.Windows.Forms.FlatStyle)(resources.GetObject("btnCancel.FlatStyle")));
        this.btnCancel.Font =
((System.Drawing.Font)(resources.GetObject("btnCancel.Font")));
        this.btnCancel.Image =
((System.Drawing.Image)(resources.GetObject("btnCancel.Image")));
        this.btnCancel.ImageAlign =
((System.Drawing.ContentAlignment)(resources.GetObject("btnCancel.ImageAlign")));
        this.btnCancel.ImageIndex = ((int)(resources.GetObject("btnCancel.ImageIndex")));
        this.btnCancel.ImeMode =
((System.Windows.Forms.ImeMode)(resources.GetObject("btnCancel.ImeMode")));
        this.btnCancel.Location =
((System.Drawing.Point)(resources.GetObject("btnCancel.Location")));
        this.btnCancel.Name = "btnCancel";
        this.btnCancel.RightToLeft =
((System.Windows.Forms.RightToLeft)(resources.GetObject("btnCancel.RightToLeft")));
        this.btnCancel.Size = ((System.Drawing.Size)(resources.GetObject("btnCancel.Size")));
        this.btnCancel.TabIndex = ((int)(resources.GetObject("btnCancel.TabIndex")));
        this.btnCancel.Text = resources.GetString("btnCancel.Text");
        this.btnCancel.TextAlign =
((System.Drawing.ContentAlignment)(resources.GetObject("btnCancel.TextAlign")));
        this.btnCancel.Visible = ((bool)(resources.GetObject("btnCancel.Visible")));
        this.btnCancel.Click += new System.EventHandler(this.btnCancel_Click);
        //
        // btnOk
        //

```

```

        this.btnOk.AccessibleDescription = resources.GetString("btnOk.AccessibleDescription");
        this.btnOk.AccessibleName = resources.GetString("btnOk.AccessibleName");
        this.btnOk.Anchor =
((System.Windows.Forms.AnchorStyles)(resources.GetObject("btnOk.Anchor")));
        this.btnOk.BackgroundImage =
((System.Drawing.Image)(resources.GetObject("btnOk.BackgroundImage")));
        this.btnOk.Dock =
((System.Windows.Forms.DockStyle)(resources.GetObject("btnOk.Dock")));
        this.btnOk.Enabled = ((bool)(resources.GetObject("btnOk.Enabled")));
        this.btnOk.FlatStyle =
((System.Windows.Forms.FlatStyle)(resources.GetObject("btnOk.FlatStyle")));
        this.btnOk.Font = ((System.Drawing.Font)(resources.GetObject("btnOk.Font")));
        this.btnOk.Image =
((System.Drawing.Image)(resources.GetObject("btnOk.Image")));
        this.btnOk.ImageAlign =
((System.Drawing.ContentAlignment)(resources.GetObject("btnOk.ImageAlign")));
        this.btnOk.ImageIndex = ((int)(resources.GetObject("btnOk.ImageIndex")));
        this.btnOk.ImeMode =
((System.Windows.Forms.ImeMode)(resources.GetObject("btnOk.ImeMode")));
        this.btnOk.Location =
((System.Drawing.Point)(resources.GetObject("btnOk.Location")));
        this.btnOk.Name = "btnOk";
        this.btnOk.RightToLeft =
((System.Windows.Forms.RightToLeft)(resources.GetObject("btnOk.RightToLeft")));
        this.btnOk.Size = ((System.Drawing.Size)(resources.GetObject("btnOk.Size")));
        this.btnOk.TabIndex = ((int)(resources.GetObject("btnOk.TabIndex")));
        this.btnOk.Text = resources.GetString("btnOk.Text");
        this.btnOk.TextAlign =
((System.Drawing.ContentAlignment)(resources.GetObject("btnOk.TextAlign")));
        this.btnOk.Visible = ((bool)(resources.GetObject("btnOk.Visible")));
        this.btnOk.Click += new System.EventHandler(this.btnOk_Click);
        //
        // cbLogin
        //
        this.cbLogin.AccessibleDescription =
resources.GetString("cbLogin.AccessibleDescription");
        this.cbLogin.AccessibleName = resources.GetString("cbLogin.AccessibleName");
        this.cbLogin.Anchor =
((System.Windows.Forms.AnchorStyles)(resources.GetObject("cbLogin.Anchor")));
        this.cbLogin.BackgroundImage =
((System.Drawing.Image)(resources.GetObject("cbLogin.BackgroundImage")));
        this.cbLogin.Dock =
((System.Windows.Forms.DockStyle)(resources.GetObject("cbLogin.Dock")));
        this.cbLogin.Enabled = ((bool)(resources.GetObject("cbLogin.Enabled")));
        this.cbLogin.Font =
((System.Drawing.Font)(resources.GetObject("cbLogin.Font")));
        this.cbLogin.ImeMode =
((System.Windows.Forms.ImeMode)(resources.GetObject("cbLogin.ImeMode")));
        this.cbLogin.IntegralHeight =
((bool)(resources.GetObject("cbLogin.IntegralHeight")));
        this.cbLogin.ItemHeight = ((int)(resources.GetObject("cbLogin.ItemHeight")));
        this.cbLogin.Location =

```



```

((System.Drawing.Point)(resources.GetObject("cbLogin.Location")));
    this.cbLogin.MaxDropDownItems =
((int)(resources.GetObject("cbLogin.MaxDropDownItems")));
    this.cbLogin.MaxLength = ((int)(resources.GetObject("cbLogin.MaxLength")));
    this.cbLogin.Name = "cbLogin";
    this.cbLogin.RightToLeft =
((System.Windows.Forms.RightToLeft)(resources.GetObject("cbLogin.RightToLeft")));
    this.cbLogin.Size = ((System.Drawing.Size)(resources.GetObject("cbLogin.Size")));
    this.cbLogin.TabIndex = ((int)(resources.GetObject("cbLogin.TabIndex")));
    this.cbLogin.Text = resources.GetString("cbLogin.Text");
    this.cbLogin.Visible = ((bool)(resources.GetObject("cbLogin.Visible")));
    this.cbLogin.KeyUp += new
System.Windows.Forms.KeyEventHandler(this.txtBLogin_KeyUp);
    this.cbLogin.SelectedIndexChanged += new
System.EventHandler(this.cbLogin_SelectedIndexChanged);
    //
    // FLogin
    //
    this.AccessibleDescription = resources.GetString("$this.AccessibleDescription");
    this.AccessibleName = resources.GetString("$this.AccessibleName");
    this.AutoScaleBaseSize =
((System.Drawing.Size)(resources.GetObject("$this.AutoScaleBaseSize")));
    this.AutoScroll = ((bool)(resources.GetObject("$this.AutoScroll")));
    this.AutoScrollMargin =
((System.Drawing.Size)(resources.GetObject("$this.AutoScrollMargin")));
    this.AutoScrollMinSize =
((System.Drawing.Size)(resources.GetObject("$this.AutoScrollMinSize")));
    this.BackgroundImage =
((System.Drawing.Image)(resources.GetObject("$this.BackgroundImage")));
    this.ClientSize = ((System.Drawing.Size)(resources.GetObject("$this.ClientSize")));
    this.Controls.Add(this.cbLogin);
    this.Controls.Add(this.btnOk);
    this.Controls.Add(this.btnCancel);
    this.Controls.Add(this.txtBPwd);
    this.Controls.Add(this.label3);
    this.Controls.Add(this.label2);
    this.Controls.Add(this.label1);
    this.Enabled = ((bool)(resources.GetObject("$this.Enabled")));
    this.Font = ((System.Drawing.Font)(resources.GetObject("$this.Font")));
    this.FormBorderStyle = System.Windows.Forms.FormBorderStyle.FixedDialog;
    this.Icon = ((System.Drawing.Icon)(resources.GetObject("$this.Icon")));
    this.ImeMode =
((System.Windows.Forms.ImeMode)(resources.GetObject("$this.ImeMode")));
    this.Location = ((System.Drawing.Point)(resources.GetObject("$this.Location")));
    this.MaximizeBox = false;
    this.MaximumSize =
((System.Drawing.Size)(resources.GetObject("$this.MaximumSize")));
    this.MinimumSize =
((System.Drawing.Size)(resources.GetObject("$this.MinimumSize")));
    this.Name = "FLogin";
    this.RightToLeft =
((System.Windows.Forms.RightToLeft)(resources.GetObject("$this.RightToLeft")));

```

```

        this.StartPosition =
((System.Windows.Forms.FormStartPosition)(resources.GetObject("$this.StartPosition")));
        this.Text = resources.GetString("$this.Text");
        this.Load += new System.EventHandler(this.FLogin_Load);
        this.ResumeLayout(false);

    }
#endregion
private void btnSubmit_Click(object sender, System.EventArgs e)
{
    }
private void btnCancel_Click(object sender, System.EventArgs e)
{
}
private void txtBLogin_KeyUp(object sender, System.Windows.Forms.KeyEventArgs e)
{
    if(e.KeyCode == Keys.Enter)
    {
        txtBPwd.Focus();
    }
}
private void txtBPwd_TextChanged(object sender, System.EventArgs e)
{
}
private void txtBPwd_KeyPress(object sender, System.Windows.Forms.KeyPressEventArgs e)
{
}
private void txtBPwd_KeyUp(object sender, System.Windows.Forms.KeyEventArgs e)
{
    if(e.KeyCode == Keys.Enter)
    {
        btnOk.Focus();
        btnOk_Click(sender,e);
    }
}
private void btnOk_Click(object sender, System.EventArgs e)
{
    if(asa_conn.State == 0)
        SetConnection();
    str_pwd = GetUsData(str_cmd);
    if(txtBPwd.Text.Length !=0)
    {
        string str_tmp = "";
        for(int i =0;i<str_pwd.Length;i++)
        {
            str_tmp = str_pwd[i];
            if(String.Compare(str_tmp,txtBPwd.Text,false) == 0)
            {
                flag=true;
                break;
            }
        }
    }
}

```

```

        }
    }
    if(flag)
    {
        str_cmd="select FName + ' ' + LName from empl where empl_id = (select empl_id from uauth
where Ulogin="" + cbLogin.Items[cbLogin.SelectedIndex].ToString()+""");
        string [] str_tmp = GetUsData(str_cmd);
        str_info.Insert(0,str_tmp[0].ToString());
        this.Close();
    }
    else
    {
        string message = "Try again?";
        string caption = "Invalid Password";
        MessageBoxButtons buttons = MessageBoxButtons.YesNo;
        DialogResult result;
        result = MessageBox.Show(this, message, caption, buttons,
            MessageBoxIcon.Error);

        if(result == DialogResult.No)
        {
            this.Close();
            Application.Exit();
        }
    }
}
private void btnCanc_Click(object sender, System.EventArgs e)
{
    string message = "Exit from application?";
    string caption = "Confirm";
    MessageBoxButtons buttons = MessageBoxButtons.YesNo;
    DialogResult result;
    result = MessageBox.Show(this, message, caption, buttons,
        MessageBoxIcon.Question, MessageBoxDefaultButton.Button1,
        MessageBoxOptions.RightAlign);
    if(result == DialogResult.Yes)
    {
        this.Close();
        Application.Exit();
    }
}
public string[] GetUsData(string cmd)
{
    string[] str_rslt = new string[20];
    asa_cmd = new AsaCommand();
    asa_cmd.Connection = asa_conn;
    asa_cmd.CommandText = cmd;
    AsaDataReader reader = asa_cmd.ExecuteReader();
    int i = 0;
    while(reader.Read())
    {
        str_rslt[i] = reader.GetString(0);
    }
}

```

```

        i++;
    }
    string[] rslt = new string[i];
    for(int j=0;j<i;j++)
    {
        rslt[j] = String.Copy(str_rslt[j]);
    }
    return rslt;
}
public bool SetConnection()
{
    bool flag=false;
    try
    {
        asa_conn = new AsaConnection(str_conn);
        asa_conn.Open();
        flag = true;
    }
    catch(AsaException ex )
    {
        MessageBox.Show( ex.Errors[0].Source + " : "
            + ex.Errors[0].Message + " (" +
            ex.Errors[0].NativeError.ToString() + ")",
            "Failed to connect" );
    }
    return flag;
}
private void FLogin_Load(object sender, System.EventArgs e)
{
    if(SetConnection())
    {
        str_cmd="select Ulogin from uauth";
        cbLogin.Items.Clear();
        cbLogin.Items.AddRange(GetUsData(str_cmd));
        cbLogin.SelectedIndex = 0;
    }
    asa_conn.Close();
}
private void cbLogin_SelectedIndexChanged(object sender, System.EventArgs e)
{
    str_cmd="select pwd from uauth where
Ulogin='"+cbLogin.Items[cbLogin.SelectedIndex].ToString()+"'";
}
}
}

using System;
using System.Drawing;
using System.Collections;
using System.Text;
using System.ComponentModel;
using System.Collections.Specialized;

```

```

using System.Windows.Forms;
using System.Data;
namespace Messenger
{
    /// <summary>
    /// Summary description for Form1.
    /// </summary>
    public class FMain : System.Windows.Forms.Form
    {
        private System.Windows.Forms.MainMenu mainMenu1;
        private System.Windows.Forms.MenuItem menuItem1;
        private System.Windows.Forms.MenuItem menuItem2;
        private System.Windows.Forms.MenuItem menuItem3;
        private System.Windows.Forms.MenuItem menuItem4;
        private System.Windows.Forms.MenuItem menuItem5;
        private System.Windows.Forms.MenuItem menuItem6;
        private System.Windows.Forms.MenuItem menuItem7;
        private System.Windows.Forms.MenuItem menuItem8;
        private System.Windows.Forms.MenuItem menuItem9;
        private System.Windows.Forms.MenuItem menuItem10;
        private System.Windows.Forms.MenuItem menuItem11;
        private System.Windows.Forms.MenuItem menuItem12;
        private System.Windows.Forms.MenuItem menuItem13;
        private System.Windows.Forms.MenuItem menuItem14;
        private System.Windows.Forms.MenuItem menuItem15;
        private System.Windows.Forms.MenuItem menuItem16;
        private System.Windows.Forms.MenuItem menuItem17;
        private System.Windows.Forms.MenuItem menuItem18;
        private System.Windows.Forms.MenuItem menuItem19;
        private System.Windows.Forms.MenuItem menuItem20;
        private System.Windows.Forms.MenuItem menuItem21;
        private System.Windows.Forms.MenuItem menuItem22;
        private System.Windows.Forms.MenuItem menuItem23;
        private System.Windows.Forms.MenuItem menuItem24;
        private System.Windows.Forms.MenuItem menuItem25;
        private System.Windows.Forms.MenuItem menuItem26;
        private System.Windows.Forms.MenuItem menuItem27;
        private System.Windows.Forms.MenuItem menuItem28;
        private System.Windows.Forms.MenuItem menuItem29;
        private System.Windows.Forms.MenuItem menuItem30;
        private System.Windows.Forms.MenuItem menuItem31;
        private System.Windows.Forms.MenuItem menuItem32;
        private System.Windows.Forms.MenuItem menuItem33;
        private System.Windows.Forms.TextBox txtInfo;
        /// <summary>
        private StringCollection str_AllFrom103;
        /// </summary>
        private System.ComponentModel.Container components = null;
        public FMain()
        {
            //
            // Required for Windows Form Designer support

```

```

        //
        InitializeComponent();

        //
        // TODO: Add any constructor code after InitializeComponent call
        //
    }

    /// <summary>
    /// Clean up any resources being used.
    /// </summary>
    protected override void Dispose( bool disposing )
    {
        if( disposing )
        {
            if (components != null)
            {
                components.Dispose();
            }
        }
        base.Dispose( disposing );
    }
    #region Windows Form Designer generated code
    /// <summary>
    /// Required method for Designer support - do not modify
    /// the contents of this method with the code editor.
    /// </summary>
    private void InitializeComponent()
    {
        System.Resources.ResourceManager resources = new
        System.Resources.ResourceManager(typeof(FMain));
        this.mainMenu1 = new System.Windows.Forms.MainMenu();
        this.menuItem1 = new System.Windows.Forms.MenuItem();
        this.menuItem6 = new System.Windows.Forms.MenuItem();
        this.menuItem7 = new System.Windows.Forms.MenuItem();
        this.menuItem8 = new System.Windows.Forms.MenuItem();
        this.menuItem9 = new System.Windows.Forms.MenuItem();
        this.menuItem10 = new System.Windows.Forms.MenuItem();
        this.menuItem11 = new System.Windows.Forms.MenuItem();
        this.menuItem2 = new System.Windows.Forms.MenuItem();
        this.menuItem24 = new System.Windows.Forms.MenuItem();
        this.menuItem25 = new System.Windows.Forms.MenuItem();
        this.menuItem12 = new System.Windows.Forms.MenuItem();
        this.menuItem13 = new System.Windows.Forms.MenuItem();
        this.menuItem14 = new System.Windows.Forms.MenuItem();
        this.menuItem15 = new System.Windows.Forms.MenuItem();
        this.menuItem16 = new System.Windows.Forms.MenuItem();
        this.menuItem17 = new System.Windows.Forms.MenuItem();
        this.menuItem18 = new System.Windows.Forms.MenuItem();
        this.menuItem19 = new System.Windows.Forms.MenuItem();
        this.menuItem3 = new System.Windows.Forms.MenuItem();
        this.menuItem20 = new System.Windows.Forms.MenuItem();
    }
}

```

```

this.menuItem21 = new System.Windows.Forms.MenuItem();
this.menuItem22 = new System.Windows.Forms.MenuItem();
this.menuItem4 = new System.Windows.Forms.MenuItem();
this.menuItem23 = new System.Windows.Forms.MenuItem();
this.menuItem26 = new System.Windows.Forms.MenuItem();
this.menuItem27 = new System.Windows.Forms.MenuItem();
this.menuItem28 = new System.Windows.Forms.MenuItem();
this.menuItem29 = new System.Windows.Forms.MenuItem();
this.menuItem30 = new System.Windows.Forms.MenuItem();
this.menuItem31 = new System.Windows.Forms.MenuItem();
this.menuItem5 = new System.Windows.Forms.MenuItem();
this.menuItem32 = new System.Windows.Forms.MenuItem();
this.menuItem33 = new System.Windows.Forms.MenuItem();
this.txtInfo = new System.Windows.Forms.TextBox();
this.SuspendLayout();
//
// mainMenu1
//
this.mainMenu1.MenuItems.AddRange(new System.Windows.Forms.MenuItem[] {
this.menuItem1, this.menuItem2, this.menuItem3, this.menuItem4,
this.menuItem5});
this.mainMenu1.RightToLeft =
((System.Windows.Forms.RightToLeft)resources.GetObject("mainMenu1.RightToLeft"));
//
// menuItem1
//
this.menuItem1.Enabled = ((bool)resources.GetObject("menuItem1.Enabled"));
this.menuItem1.Index = 0;
this.menuItem1.MenuItems.AddRange(new System.Windows.Forms.MenuItem[] {
this.menuItem6, this.menuItem7, this.menuItem8, this.menuItem9,
this.menuItem10, this.menuItem11});
this.menuItem1.Shortcut =
((System.Windows.Forms.Shortcut)resources.GetObject("menuItem1.Shortcut"));
this.menuItem1.ShowShortcut =
((bool)resources.GetObject("menuItem1.ShowShortcut"));
this.menuItem1.Text = resources.GetString("menuItem1.Text");
this.menuItem1.Visible = ((bool)resources.GetObject("menuItem1.Visible"));
//
// menuItem6
//
this.menuItem6.Enabled = ((bool)resources.GetObject("menuItem6.Enabled"));
this.menuItem6.Index = 0;
this.menuItem6.Shortcut =
((System.Windows.Forms.Shortcut)resources.GetObject("menuItem6.Shortcut"));
this.menuItem6.ShowShortcut =
((bool)resources.GetObject("menuItem6.ShowShortcut"));
this.menuItem6.Text = resources.GetString("menuItem6.Text");
this.menuItem6.Visible = ((bool)resources.GetObject("menuItem6.Visible"));
this.menuItem6.Click += new System.EventHandler(this.menuItem6_Click);
//
// menuItem7
//

```

```

        this.menuItem7.Enabled = ((bool)(resources.GetObject("menuItem7.Enabled")));
        this.menuItem7.Index = 1;
        this.menuItem7.Shortcut =
((System.Windows.Forms.Shortcut)(resources.GetObject("menuItem7.Shortcut")));
        this.menuItem7.ShowShortcut =
((bool)(resources.GetObject("menuItem7.ShowShortcut")));
        this.menuItem7.Text = resources.GetString("menuItem7.Text");
        this.menuItem7.Visible = ((bool)(resources.GetObject("menuItem7.Visible")));
        //
        // menuItem8
        //
        this.menuItem8.Enabled = ((bool)(resources.GetObject("menuItem8.Enabled")));
        this.menuItem8.Index = 2;
        this.menuItem8.Shortcut =
((System.Windows.Forms.Shortcut)(resources.GetObject("menuItem8.Shortcut")));
        this.menuItem8.ShowShortcut =
((bool)(resources.GetObject("menuItem8.ShowShortcut")));
        this.menuItem8.Text = resources.GetString("menuItem8.Text");
        this.menuItem8.Visible = ((bool)(resources.GetObject("menuItem8.Visible")));
        //
        // menuItem9
        //
        this.menuItem9.Enabled = ((bool)(resources.GetObject("menuItem9.Enabled")));
        this.menuItem9.Index = 3;
        this.menuItem9.Shortcut =
((System.Windows.Forms.Shortcut)(resources.GetObject("menuItem9.Shortcut")));
        this.menuItem9.ShowShortcut =
((bool)(resources.GetObject("menuItem9.ShowShortcut")));
        this.menuItem9.Text = resources.GetString("menuItem9.Text");
        this.menuItem9.Visible = ((bool)(resources.GetObject("menuItem9.Visible")));
        //
        // menuItem10
        //
        this.menuItem10.Enabled = ((bool)(resources.GetObject("menuItem10.Enabled")));
        this.menuItem10.Index = 4;
        this.menuItem10.Shortcut =
((System.Windows.Forms.Shortcut)(resources.GetObject("menuItem10.Shortcut")));
        this.menuItem10.ShowShortcut =
((bool)(resources.GetObject("menuItem10.ShowShortcut")));
        this.menuItem10.Text = resources.GetString("menuItem10.Text");
        this.menuItem10.Visible = ((bool)(resources.GetObject("menuItem10.Visible")));
        //
        // menuItem11
        //
        this.menuItem11.Enabled = ((bool)(resources.GetObject("menuItem11.Enabled")));
        this.menuItem11.Index = 5;
        this.menuItem11.Shortcut =
((System.Windows.Forms.Shortcut)(resources.GetObject("menuItem11.Shortcut")));
        this.menuItem11.ShowShortcut =
((bool)(resources.GetObject("menuItem11.ShowShortcut")));
        this.menuItem11.Text = resources.GetString("menuItem11.Text");
        this.menuItem11.Visible = ((bool)(resources.GetObject("menuItem11.Visible")));

```



```

//
// menuItem2
//
this.menuItem2.Enabled = ((bool)(resources.GetObject("menuItem2.Enabled")));
this.menuItem2.Index = 1;
this.menuItem2.MenuItems.AddRange(new System.Windows.Forms.MenuItem[] {
    this.menuItem24, this.menuItem25, this.menuItem12, this.menuItem13, this.menuItem14,
    this.menuItem15, this.menuItem16, this.menuItem17, this.menuItem18, this.menuItem19});
this.menuItem2.Shortcut =
((System.Windows.Forms.Shortcut)(resources.GetObject("menuItem2.Shortcut")));
this.menuItem2.ShowShortcut =
((bool)(resources.GetObject("menuItem2.ShowShortcut")));
this.menuItem2.Text = resources.GetString("menuItem2.Text");
this.menuItem2.Visible = ((bool)(resources.GetObject("menuItem2.Visible")));
//
// menuItem24
//
this.menuItem24.Enabled = ((bool)(resources.GetObject("menuItem24.Enabled")));
this.menuItem24.Index = 0;
this.menuItem24.Shortcut =
((System.Windows.Forms.Shortcut)(resources.GetObject("menuItem24.Shortcut")));
this.menuItem24.ShowShortcut =
((bool)(resources.GetObject("menuItem24.ShowShortcut")));
this.menuItem24.Text = resources.GetString("menuItem24.Text");
this.menuItem24.Visible = ((bool)(resources.GetObject("menuItem24.Visible")));
//
// menuItem25
//
this.menuItem25.Enabled = ((bool)(resources.GetObject("menuItem25.Enabled")));
this.menuItem25.Index = 1;
this.menuItem25.Shortcut =
((System.Windows.Forms.Shortcut)(resources.GetObject("menuItem25.Shortcut")));
this.menuItem25.ShowShortcut =
((bool)(resources.GetObject("menuItem25.ShowShortcut")));
this.menuItem25.Text = resources.GetString("menuItem25.Text");
this.menuItem25.Visible = ((bool)(resources.GetObject("menuItem25.Visible")));
//
// menuItem12
//
this.menuItem12.Enabled = ((bool)(resources.GetObject("menuItem12.Enabled")));
this.menuItem12.Index = 2;
this.menuItem12.Shortcut =
((System.Windows.Forms.Shortcut)(resources.GetObject("menuItem12.Shortcut")));
this.menuItem12.ShowShortcut =
((bool)(resources.GetObject("menuItem12.ShowShortcut")));
this.menuItem12.Text = resources.GetString("menuItem12.Text");
this.menuItem12.Visible = ((bool)(resources.GetObject("menuItem12.Visible")));
this.menuItem12.Click += new System.EventHandler(this.menuItem12_Click);
//
// menuItem13
//
this.menuItem13.Enabled = ((bool)(resources.GetObject("menuItem13.Enabled")));

```

```

        this.menuItem13.Index = 3;
        this.menuItem13.Shortcut =
((System.Windows.Forms.Shortcut)(resources.GetObject("menuItem13.Shortcut")));
        this.menuItem13.ShowShortcut =
((bool)(resources.GetObject("menuItem13.ShowShortcut")));
        this.menuItem13.Text = resources.GetString("menuItem13.Text");
        this.menuItem13.Visible = ((bool)(resources.GetObject("menuItem13.Visible")));
        //
        // menuItem14
        //
        this.menuItem14.Enabled = ((bool)(resources.GetObject("menuItem14.Enabled")));
        this.menuItem14.Index = 4;
        this.menuItem14.Shortcut =
((System.Windows.Forms.Shortcut)(resources.GetObject("menuItem14.Shortcut")));
        this.menuItem14.ShowShortcut =
((bool)(resources.GetObject("menuItem14.ShowShortcut")));
        this.menuItem14.Text = resources.GetString("menuItem14.Text");
        this.menuItem14.Visible = ((bool)(resources.GetObject("menuItem14.Visible")));
        //
        // menuItem15
        //
        this.menuItem15.Enabled = ((bool)(resources.GetObject("menuItem15.Enabled")));
        this.menuItem15.Index = 5;
        this.menuItem15.Shortcut =
((System.Windows.Forms.Shortcut)(resources.GetObject("menuItem15.Shortcut")));
        this.menuItem15.ShowShortcut =
((bool)(resources.GetObject("menuItem15.ShowShortcut")));
        this.menuItem15.Text = resources.GetString("menuItem15.Text");
        this.menuItem15.Visible = ((bool)(resources.GetObject("menuItem15.Visible")));
        //
        // menuItem16
        //
        this.menuItem16.Enabled = ((bool)(resources.GetObject("menuItem16.Enabled")));
        this.menuItem16.Index = 6;
        this.menuItem16.Shortcut =
((System.Windows.Forms.Shortcut)(resources.GetObject("menuItem16.Shortcut")));
        this.menuItem16.ShowShortcut =
((bool)(resources.GetObject("menuItem16.ShowShortcut")));
        this.menuItem16.Text = resources.GetString("menuItem16.Text");
        this.menuItem16.Visible = ((bool)(resources.GetObject("menuItem16.Visible")));
        //
        // menuItem17
        //
        this.menuItem17.Enabled = ((bool)(resources.GetObject("menuItem17.Enabled")));
        this.menuItem17.Index = 7;
        this.menuItem17.Shortcut =
((System.Windows.Forms.Shortcut)(resources.GetObject("menuItem17.Shortcut")));
        this.menuItem17.ShowShortcut =
((bool)(resources.GetObject("menuItem17.ShowShortcut")));
        this.menuItem17.Text = resources.GetString("menuItem17.Text");
        this.menuItem17.Visible = ((bool)(resources.GetObject("menuItem17.Visible")));
        //

```

```

// menuItem18
//
this.menuItem18.Enabled = ((bool)(resources.GetObject("menuItem18.Enabled")));
this.menuItem18.Index = 8;
this.menuItem18.Shortcut =
((System.Windows.Forms.Shortcut)(resources.GetObject("menuItem18.Shortcut")));
this.menuItem18.ShowShortcut =
((bool)(resources.GetObject("menuItem18.ShowShortcut")));
this.menuItem18.Text = resources.GetString("menuItem18.Text");
this.menuItem18.Visible = ((bool)(resources.GetObject("menuItem18.Visible")));
//
// menuItem19
//
this.menuItem19.Enabled = ((bool)(resources.GetObject("menuItem19.Enabled")));
this.menuItem19.Index = 9;
this.menuItem19.Shortcut =
((System.Windows.Forms.Shortcut)(resources.GetObject("menuItem19.Shortcut")));
this.menuItem19.ShowShortcut =
((bool)(resources.GetObject("menuItem19.ShowShortcut")));
this.menuItem19.Text = resources.GetString("menuItem19.Text");
this.menuItem19.Visible = ((bool)(resources.GetObject("menuItem19.Visible")));
//
// menuItem3
//
this.menuItem3.Enabled = ((bool)(resources.GetObject("menuItem3.Enabled")));
this.menuItem3.Index = 2;
this.menuItem3.MenuItems.AddRange(new System.Windows.Forms.MenuItem[] {
    this.menuItem20, this.menuItem21, this.menuItem22});
this.menuItem3.Shortcut =
((System.Windows.Forms.Shortcut)(resources.GetObject("menuItem3.Shortcut")));
this.menuItem3.ShowShortcut =
((bool)(resources.GetObject("menuItem3.ShowShortcut")));
this.menuItem3.Text = resources.GetString("menuItem3.Text");
this.menuItem3.Visible = ((bool)(resources.GetObject("menuItem3.Visible")));
//
// menuItem20
//
this.menuItem20.Enabled = ((bool)(resources.GetObject("menuItem20.Enabled")));
this.menuItem20.Index = 0;
this.menuItem20.Shortcut =
((System.Windows.Forms.Shortcut)(resources.GetObject("menuItem20.Shortcut")));
this.menuItem20.ShowShortcut =
((bool)(resources.GetObject("menuItem20.ShowShortcut")));
this.menuItem20.Text = resources.GetString("menuItem20.Text");
this.menuItem20.Visible = ((bool)(resources.GetObject("menuItem20.Visible")));
//
// menuItem21
//
this.menuItem21.Enabled = ((bool)(resources.GetObject("menuItem21.Enabled")));
this.menuItem21.Index = 1;
this.menuItem21.Shortcut =
((System.Windows.Forms.Shortcut)(resources.GetObject("menuItem21.Shortcut")));

```

```

        this.menuItem21.ShowShortcut =
((bool)(resources.GetObject("menuItem21.ShowShortcut")));
        this.menuItem21.Text = resources.GetString("menuItem21.Text");
        this.menuItem21.Visible = ((bool)(resources.GetObject("menuItem21.Visible")));
        //
        // menuItem22
        //
        this.menuItem22.Enabled = ((bool)(resources.GetObject("menuItem22.Enabled")));
        this.menuItem22.Index = 2;
        this.menuItem22.Shortcut =
((System.Windows.Forms.Shortcut)(resources.GetObject("menuItem22.Shortcut")));
        this.menuItem22.ShowShortcut =
((bool)(resources.GetObject("menuItem22.ShowShortcut")));
        this.menuItem22.Text = resources.GetString("menuItem22.Text");
        this.menuItem22.Visible = ((bool)(resources.GetObject("menuItem22.Visible")));
        //
        // menuItem4
        //
        this.menuItem4.Enabled = ((bool)(resources.GetObject("menuItem4.Enabled")));
        this.menuItem4.Index = 3;
        this.menuItem4.MenuItems.AddRange(new System.Windows.Forms.MenuItem[] {
            this.menuItem23 });
        this.menuItem4.Shortcut =
((System.Windows.Forms.Shortcut)(resources.GetObject("menuItem4.Shortcut")));
        this.menuItem4.ShowShortcut =
((bool)(resources.GetObject("menuItem4.ShowShortcut")));
        this.menuItem4.Text = resources.GetString("menuItem4.Text");
        this.menuItem4.Visible = ((bool)(resources.GetObject("menuItem4.Visible")));
        //
        // menuItem23
        //
        this.menuItem23.Enabled = ((bool)(resources.GetObject("menuItem23.Enabled")));
        this.menuItem23.Index = 0;
        this.menuItem23.MenuItems.AddRange(new System.Windows.Forms.MenuItem[] {
            this.menuItem26, this.menuItem27, this.menuItem28, this.menuItem29,
this.menuItem30, this.menuItem31 });
        this.menuItem23.Shortcut =
((System.Windows.Forms.Shortcut)(resources.GetObject("menuItem23.Shortcut")));
        this.menuItem23.ShowShortcut =
((bool)(resources.GetObject("menuItem23.ShowShortcut")));
        this.menuItem23.Text = resources.GetString("menuItem23.Text");
        this.menuItem23.Visible = ((bool)(resources.GetObject("menuItem23.Visible")));
        //
        // menuItem26
        //
        this.menuItem26.Enabled = ((bool)(resources.GetObject("menuItem26.Enabled")));
        this.menuItem26.Index = 0;
        this.menuItem26.Shortcut =
((System.Windows.Forms.Shortcut)(resources.GetObject("menuItem26.Shortcut")));
        this.menuItem26.ShowShortcut =
((bool)(resources.GetObject("menuItem26.ShowShortcut")));
        this.menuItem26.Text = resources.GetString("menuItem26.Text");

```

```

        this.menuItem26.Visible = ((bool)(resources.GetObject("menuItem26.Visible")));
        //
        // menuItem27
        //
        this.menuItem27.Enabled = ((bool)(resources.GetObject("menuItem27.Enabled")));
        this.menuItem27.Index = 1;
        this.menuItem27.Shortcut =
((System.Windows.Forms.Shortcut)(resources.GetObject("menuItem27.Shortcut")));
        this.menuItem27.ShowShortcut =
((bool)(resources.GetObject("menuItem27.ShowShortcut")));
        this.menuItem27.Text = resources.GetString("menuItem27.Text");
        this.menuItem27.Visible = ((bool)(resources.GetObject("menuItem27.Visible")));
        //
        // menuItem28
        //
        this.menuItem28.Enabled = ((bool)(resources.GetObject("menuItem28.Enabled")));
        this.menuItem28.Index = 2;
        this.menuItem28.Shortcut =
((System.Windows.Forms.Shortcut)(resources.GetObject("menuItem28.Shortcut")));
        this.menuItem28.ShowShortcut =
((bool)(resources.GetObject("menuItem28.ShowShortcut")));
        this.menuItem28.Text = resources.GetString("menuItem28.Text");
        this.menuItem28.Visible = ((bool)(resources.GetObject("menuItem28.Visible")));
        //
        // menuItem29
        //
        this.menuItem29.Enabled = ((bool)(resources.GetObject("menuItem29.Enabled")));
        this.menuItem29.Index = 3;
        this.menuItem29.Shortcut =
((System.Windows.Forms.Shortcut)(resources.GetObject("menuItem29.Shortcut")));
        this.menuItem29.ShowShortcut =
((bool)(resources.GetObject("menuItem29.ShowShortcut")));
        this.menuItem29.Text = resources.GetString("menuItem29.Text");
        this.menuItem29.Visible = ((bool)(resources.GetObject("menuItem29.Visible")));
        //
        // menuItem30
        //
        this.menuItem30.Enabled = ((bool)(resources.GetObject("menuItem30.Enabled")));
        this.menuItem30.Index = 4;
        this.menuItem30.Shortcut =
((System.Windows.Forms.Shortcut)(resources.GetObject("menuItem30.Shortcut")));
        this.menuItem30.ShowShortcut =
((bool)(resources.GetObject("menuItem30.ShowShortcut")));
        this.menuItem30.Text = resources.GetString("menuItem30.Text");
        this.menuItem30.Visible = ((bool)(resources.GetObject("menuItem30.Visible")));
        //
        // menuItem31
        //
        this.menuItem31.Enabled = ((bool)(resources.GetObject("menuItem31.Enabled")));
        this.menuItem31.Index = 5;
        this.menuItem31.Shortcut =
((System.Windows.Forms.Shortcut)(resources.GetObject("menuItem31.Shortcut")));

```

```

        this.menuItem31.ShowShortcut =
((bool)(resources.GetObject("menuItem31.ShowShortcut")));
        this.menuItem31.Text = resources.GetString("menuItem31.Text");
        this.menuItem31.Visible = ((bool)(resources.GetObject("menuItem31.Visible")));
        //
        // menuItem5
        //
        this.menuItem5.Enabled = ((bool)(resources.GetObject("menuItem5.Enabled")));
        this.menuItem5.Index = 4;
        this.menuItem5.MenuItems.AddRange(new System.Windows.Forms.MenuItem[] {
            this.menuItem32, this.menuItem33});
        this.menuItem5.Shortcut =
((System.Windows.Forms.Shortcut)(resources.GetObject("menuItem5.Shortcut")));
        this.menuItem5.ShowShortcut =
((bool)(resources.GetObject("menuItem5.ShowShortcut")));
        this.menuItem5.Text = resources.GetString("menuItem5.Text");
        this.menuItem5.Visible = ((bool)(resources.GetObject("menuItem5.Visible")));
        //
        // menuItem32
        //
        this.menuItem32.Enabled = ((bool)(resources.GetObject("menuItem32.Enabled")));
        this.menuItem32.Index = 0;
        this.menuItem32.Shortcut =
((System.Windows.Forms.Shortcut)(resources.GetObject("menuItem32.Shortcut")));
        this.menuItem32.ShowShortcut =
((bool)(resources.GetObject("menuItem32.ShowShortcut")));
        this.menuItem32.Text = resources.GetString("menuItem32.Text");
        this.menuItem32.Visible = ((bool)(resources.GetObject("menuItem32.Visible")));
        //
        // menuItem33
        //
        this.menuItem33.Enabled = ((bool)(resources.GetObject("menuItem33.Enabled")));
        this.menuItem33.Index = 1;
        this.menuItem33.Shortcut =
((System.Windows.Forms.Shortcut)(resources.GetObject("menuItem33.Shortcut")));
        this.menuItem33.ShowShortcut =
((bool)(resources.GetObject("menuItem33.ShowShortcut")));
        this.menuItem33.Text = resources.GetString("menuItem33.Text");
        this.menuItem33.Visible = ((bool)(resources.GetObject("menuItem33.Visible")));
        //
        // txtInfo
        //
        this.txtInfo.AccessibleDescription =
resources.GetString("txtInfo.AccessibleDescription");
        this.txtInfo.AccessibleName = resources.GetString("txtInfo.AccessibleName");
        this.txtInfo.Anchor =
((System.Windows.Forms.AnchorStyles)(resources.GetObject("txtInfo.Anchor")));
        this.txtInfo.AutoSize = ((bool)(resources.GetObject("txtInfo.AutoSize")));
        this.txtInfo.BackColor = System.Drawing.SystemColors.InactiveBorder;
        this.txtInfo.BackgroundImage =
((System.Drawing.Image)(resources.GetObject("txtInfo.BackgroundImage")));
        this.txtInfo.BorderStyle = System.Windows.Forms.BorderStyle.None;

```

```

        this.txtInfo.Dock =
((System.Windows.Forms.DockStyle)(resources.GetObject("txtInfo.Dock")));
        this.txtInfo.Enabled = ((bool)(resources.GetObject("txtInfo.Enabled")));
        this.txtInfo.Font = ((System.Drawing.Font)(resources.GetObject("txtInfo.Font")));
        this.txtInfo.ImeMode =
((System.Windows.Forms.ImeMode)(resources.GetObject("txtInfo.ImeMode")));
        this.txtInfo.Location =
((System.Drawing.Point)(resources.GetObject("txtInfo.Location")));
        this.txtInfo.MaxLength = ((int)(resources.GetObject("txtInfo.MaxLength")));
        this.txtInfo.Multiline = ((bool)(resources.GetObject("txtInfo.Multiline")));
        this.txtInfo.Name = "txtInfo";
        this.txtInfo.PasswordChar = ((char)(resources.GetObject("txtInfo.PasswordChar")));
        this.txtInfo.RightToLeft =
((System.Windows.Forms.RightToLeft)(resources.GetObject("txtInfo.RightToLeft")));
        this.txtInfo.ScrollBars =
((System.Windows.Forms.ScrollBars)(resources.GetObject("txtInfo.ScrollBars")));
        this.txtInfo.Size = ((System.Drawing.Size)(resources.GetObject("txtInfo.Size")));
        this.txtInfo.TabIndex = ((int)(resources.GetObject("txtInfo.TabIndex")));
        this.txtInfo.Text = resources.GetString("txtInfo.Text");
        this.txtInfo.TextAlign =
((System.Windows.Forms.HorizontalAlignment)(resources.GetObject("txtInfo.TextAlign")));
        this.txtInfo.Visible = ((bool)(resources.GetObject("txtInfo.Visible")));
        this.txtInfo.WordWrap = ((bool)(resources.GetObject("txtInfo.WordWrap")));
        //
        // FMain
        //
        this.AccessibleDescription = resources.GetString("$this.AccessibleDescription");
        this.AccessibleName = resources.GetString("$this.AccessibleName");
        this.AutoScaleBaseSize =
((System.Drawing.Size)(resources.GetObject("$this.AutoScaleBaseSize")));
        this.AutoScroll = ((bool)(resources.GetObject("$this.AutoScroll")));
        this.AutoScrollMargin =
((System.Drawing.Size)(resources.GetObject("$this.AutoScrollMargin")));
        this.AutoScrollMinSize =
((System.Drawing.Size)(resources.GetObject("$this.AutoScrollMinSize")));
        this.BackgroundImage =
((System.Drawing.Image)(resources.GetObject("$this.BackgroundImage")));
        this.ClientSize = ((System.Drawing.Size)(resources.GetObject("$this.ClientSize")));
        this.Controls.Add(this.txtInfo);
        this.Enabled = ((bool)(resources.GetObject("$this.Enabled")));
        this.Font = ((System.Drawing.Font)(resources.GetObject("$this.Font")));
        this.FormBorderStyle = System.Windows.Forms.FormBorderStyle.Fixed3D;
        this.Icon = ((System.Drawing.Icon)(resources.GetObject("$this.Icon")));
        this.ImeMode =
((System.Windows.Forms.ImeMode)(resources.GetObject("$this.ImeMode")));
        this.Location = ((System.Drawing.Point)(resources.GetObject("$this.Location")));
        this.MaximumSize =
((System.Drawing.Size)(resources.GetObject("$this.MaximumSize")));
        this.Menu = this.mainMenu1;
        this.MinimizeBox = false;
        this.MinimumSize =
((System.Drawing.Size)(resources.GetObject("$this.MinimumSize")));

```

```

        this.Name = "FMain";
        this.RightToLeft =
((System.Windows.Forms.RightToLeft)(resources.GetObject("$this.RightToLeft")));
        this.StartPosition =
((System.Windows.Forms.FormStartPosition)(resources.GetObject("$this.StartPosition")));
        this.Text = resources.GetString("$this.Text");
        this.Load += new System.EventHandler(this.FMain_Load);
        this.ResumeLayout(false);
    }
#endregion
/// <summary>
/// The main entry point for the application.
/// </summary>
[STAThread]
static void Main()
{
    Application.Run(new FMain());
}
private void menuItem6_Click(object sender, System.EventArgs e)
{
}
private void FMain_Load(object sender, System.EventArgs e)
{
    StringBuilder str_info = new StringBuilder();
    FLogin fLogin = new FLogin(str_info);
    fLogin.ShowDialog();
    fLogin.Dispose();
    txtInfo.Text = str_info.ToString();
}
private void menuItem12_Click(object sender, System.EventArgs e)
{
    string str_tmp = new string(' ',10000);
    str_tmp = str_tmp.Trim();
    CTS mt103 = new CTS();
    if(mt103.SetConnection())
    {
        ShowSearchOp();
        str_AllFrom103 = mt103.Get_103("20060227%");
        for(int i = 0;i<str_AllFrom103.Count;i++)
        {
            string str = new string(' ',str_AllFrom103[i].Length);
            str = str.Trim();
            str = String.Copy(str_AllFrom103[i]);
            str_tmp = str.Substring(str.IndexOf("{ 1:}")-19,18);
            str_tmp = mt103.Get_103_all(str_tmp);
            str = str+str_tmp;
            str_AllFrom103[i] = str;
        }
        DataSet ds;
        ds = mt103.FillDs(0,str_AllFrom103);
        // dgMain.DataSource = ds.Tables[0];

```



```

        }
    }
    void ShowSearchOp()
    {
        SearchOpt fOptDlg= new SearchOpt();
        fOptDlg.ShowDialog();
        fOptDlg.Dispose();
    }

}
}

using System;
using System.Data;
using System.Collections;
using System.Text;
using System.Collections.Specialized;
using System.Data.Common;
using System.Windows.Forms;
using iAnywhere.Data.AsaClient;
using System.Diagnostics;
namespace Messenger
{
    /// <summary>
    ///
    /// </summary>
    public class CTS
    {
        private string str_conn;
        private string str_cmd;
        private string[] str_rslt;
        //Anywhere data types
        private AsaCommand asa_cmd;
        private AsaConnection asa_conn;
        private AsaDataAdapter asa_adapter;
        //Common data types
        private DataSet ds;
        private DataTable dt;
        public CTS()
        {
            //
            str_conn = "Data Source=TS;UID=TS_DBA;PWD=TS_DBA";
            //str_cmd = "select * from sys.systable where creator=1";
            str_rslt = new string[500];
            //
        }
        public bool SetConnection()
        {
            bool flag=false;
            try
            {
                asa_conn = new AsaConnection(str_conn);

```

```

        asa_conn.Open();
        flag = true;
    }
    catch(AsaException ex )
    {
        MessageBox.Show( ex.Errors[0].Source + " : "
            + ex.Errors[0].Message + " (" +
            ex.Errors[0].NativeError.ToString() + ")",
            "Failed to connect" );
    }
    return flag;
}
public StringCollection Get_103(string str_dbkey)
{
    StringCollection str_rslt;
    str_rslt = new StringCollection();
    asa_conn = new AsaConnection(str_conn);
    try
    {
        asa_conn.Open();
        asa_cmd = new AsaCommand("read_103",asa_conn);
        asa_cmd.CommandType = CommandType.StoredProcedure;
        asa_cmd.Parameters.Add("@dbky",AsaDbType.Char,18);
        asa_cmd.Parameters[0].Value = str_dbkey;
        asa_adapter = new AsaDataAdapter(asa_cmd);
        AsaDataReader rdr = asa_cmd.ExecuteReader();
        ds = new DataSet();
        ds.EnforceConstraints = false;
        int i = asa_adapter.Fill(ds);
        ds.EnforceConstraints = true;
        while(rdr.Read())
        {
            str_rslt.Add(rdr[0] + "#" + rdr[1] + "#" + rdr[2] + "#" + rdr[3] + "#" + rdr[4] + "#" + rdr[5] + "#" +
rdr[6]);
        }
    }
    catch(AsaException asaex)
    {
        LogException(asaex);
    }
    catch(Exception ex)
    {
        throw ex;
    }
    finally
    {
        asa_conn.Close();
    }
    return str_rslt;
}
public string Get_103_all(string str_dbkey)
{

```

```

string str_rslt = new string(' ',1500);
str_rslt = str_rslt.Trim();
asa_conn = new AsaConnection(str_conn);
try
{
    asa_conn.Open();
    asa_cmd = new AsaCommand("rmte",asa_conn);
    asa_cmd.CommandType = CommandType.StoredProcedure;
    asa_cmd.Parameters.Add("@dbky",AsaDbType.Char,18);
    asa_cmd.Parameters[0].Value = str_dbkey;
    if((str_rslt = (string)asa_cmd.ExecuteScalar()) == null )
        str_rslt = " ";
}
catch(AsaException asaex)
{
    LogException(asaex);
}
catch(Exception ex)
{
    throw ex;
}
finally
{
    asa_conn.Close();
}
str_rslt = str_rslt.Trim();
return str_rslt;
}
public bool CloseCon()
{
    asa_conn.Close();
    return true;
}
private void LogException(AsaException exc)
{
    EventLog el = new EventLog();
    el.Source = "ComGenerator";
    string strMessage;
strMessage = "Exception Number : " + exc.Errors.Count + exc.Message + " has ocurred";
    el.WriteEntry(strMessage);
    foreach(AsaError asaerr in exc.Errors)
    {
        strMessage = "Message" + asaerr.Message+
            "Source " + asaerr.Source;
        el.WriteEntry(strMessage);
    }
}
public DataSet FillIDs(int indTable,StringCollection strings)
{
    DataSet ds;
    ds = new DataSet();
}

```

```

DataTable tbl;
DataColumn col;
ForeignKeyConstraint fk;

//Add Table mt_103_rcvd in User ds
tbl = ds.Tables.Add("mt103");
col = tbl.Columns.Add("f_20",typeof(string));
col.MaxLength = 16;
col = tbl.Columns.Add("vald",typeof(string));
col.MaxLength = 8;
col = tbl.Columns.Add("curc",typeof(string));
col.MaxLength = 3;
col = tbl.Columns.Add("amnt",typeof(string));
col.MaxLength = 17;
col = tbl.Columns.Add("sndr",typeof(string));
col.MaxLength = 12;
col = tbl.Columns.Add("ms_dbky",typeof(string));
col.MaxLength = 18;
col = tbl.Columns.Add("text",typeof(string));
col.MaxLength = 32767;
col.AllowDBNull = false;
tbl.PrimaryKey = new DataColumn[] {tbl.Columns[0]};
object[] aValues;
for(int i=0;i<strings.Count;i++)
{
    string str = new string(' ',1000);
    str = str.Trim();
    str = strings[i];
    aValues = str.Split('#');
    ds.Tables[indTable].LoadDataRow(aValues,false);
}
return ds;
}
}
}

```

ДОДАТОК Б

ВІДГУК КЕРІВНИКА ЕКОНОМІЧНОГО РОЗДІЛУ

ПЕРЕЛІК ФАЙЛІВ НА ДИСКУ

Ім'я файлу	Опис
Пояснювальні документи	
Диплом_ doc	Пояснювальна записка до кваліфікаційної роботи. Документ Word.
Диплом_.pdf	Пояснювальна записка до кваліфікаційної роботи в форматі PDF
Програма	
Program.rar	Архів. Містить коди програми і откомпільовану програму
Презентація	
Презентація_.ppt	Презентація кваліфікаційної роботи