МІНІСТЕРСТВО ОСВІТИ УКРАЇНИ

ХАРКІВСЬКИЙ НАЦІОНАЛЬНИЙ

УНІВЕРСИТЕТ РАДІОЕЛЕКТРОНІКИ

ПОЯСНЮВАЛЬНА ЗАПИСКА

до КУРСОВОЇ РОБОТИ

на тему

«СИСТЕМА УПРАВЛІННЯ ВІДДІЛОМ КАДРІВ»

Виконав: студент групи ПЗПІп-18-1

Кузнецов П.С.

Керівник: Саманцов О.О.

ХАРКІВ 2019

Аркуш завдання та календарний план

ХАРКІВСЬКИЙ НАЦІОНАЛЬНИЙ УНІВЕРСИТЕТ

РАДІОЕЛЕКТРОНІКИ

Кафедра: ***Центр післядипломної освіти***

Дисципліна: ***Об’єктно-орієнтоване програмування***

Напрям: ***Програмна інженерія***

Курс \_\_\_\_\_ Група *ПЗПІп-18-1* Семестр \_\_\_\_\_\_

***ЗАВДАННЯ***

***на курсовий проект студента***

Кузнецова Петра Сергійовича

(Прізвище, Ім’я, По батькові)

1 Тема проекту: *Система управління відділом кадрів*

2 Термін здачі студентом закінченого проекту: **«12» -січня-2019 року**

3 Вихідні дані до проекту:

*Специфікація програми, методичні вказівки до виконання курсової роботи .*

*.*

4 Зміст розрахунково-пояснювальної записки:

*Вступ, специфікація програми, проектна сертифікація, інструкція користувача, висновок* .

.

.

.

.

.

5 Перелік графічного матеріалу:

*UML-діаграми, лістинги коду, приклади екранних форм .*

.

.

.

.

.

.

.

**КАЛЕНДАРНИЙ ПЛАН**

|  |  |  |
| --- | --- | --- |
| *№* | *Назва етапу* | *Термін виконання* |
|  | Видача теми, узгодження і затвердження теми | 06.10.2018 |
|  | Формулювання вимог до програми | 07.10.2018-13.10.2018 |
|  | Розробка підсистеми зберігання та пошуку даних | 14.10.2018-20.10.2018 |
|  | Аналіз предметної області | 21.10.2018-27.10.2018 |
|  | Розробка функції виведення даних | 28.10.2018-03.11.2018 |
|  | Розробка функції зберігання та завантаження даних | 03.11.2018-17.11.2018 |
|  | Тестування та доопрацювання розробленої програмної системи | 18.11.2018-24.11.2018 |
|  | Оформлення пояснювальної записки, додатків, графічного матеріалу | 25.11.2018-01.12.2018 |
|  | Захист курсового проекту | 12.01.2019 року |

Студент: Кузнецов П.С.

Керівник: Саманцов О.О.

13.10.2018 року

РЕФЕРАТ

Пояснювальна записка до курсової роботи: 20 сторінок., 12 рисунків, 1 додаток, 9 джерел посилання.

Метою роботи є розробка програми «Система управління відділом кадрів» на засадах об’єктно-орієнтованого програмування.

В курсовій роботі проведений аналіз деяких програмних продуктів зі схожим функціоналом, відзначені їх позитивні та негативні сторони.

Методи розробки базуються на використанні середовища розробки Community 2017 версії 15.9.4, Windows Forms, платформи .NET FRAMEWORK 4.7, мови програмування Visual C# 7. Для наочної візуалізації процесу проектування та створення курсового проекту шляхом створення UML-діаграм використаний Microsoft Visio 2010 SP2 MSO.

В результаті розроблена програма «Personnel Manager», яка дозволяє зберігати перелік співробітників підприємства із зазначенням їх відділу , посади, ім’я, прізвища, а також дані щодо вакантних посад цього підприємства. Є можливості зберігання даних у файловій системі, пошуку, редагування, добавлення та видалення даних у графічному інтерфейсі. При цьому програмний продукт має достатньо велику кількість можливостей для розширення функціоналу.

ЗМІСТ

Вступ………………………………………….…………………………………..6

1. Аналіз предметної області….………………………………………………..7
2. Постановка завдання……………………………………………………........9

2.1 Призначення програмного продукту ………………...…………………… 9

2.2 Функціональні вимоги до програмного продукту …..…………………...10

1. Реалізація…..…………………………………………………………………11

3.1 Огляд методів та засобів проектування програмного продукту………...11

* 1. Системні вимоги…………………………………………………………....11

3.3Концептуальна схема у вигляді діаграм…………………………………..12

* + 1. Діаграма варіантів використання……………………………………...12
    2. Діаграма класів……………………………………………………….....13
    3. Діаграма взаємодії………………………………………………………14
  1. Вибір засобів для зберігання даних……………………………………….16

4 Посібник користувача………………………………………………………..18

Висновок………………………………………………………………………...23

Перелік джерел посилання……...……………………………………………...24

Додаток А

ВСТУП

Управління персоналом набуває все значнішого значення як фактор підвищення конкурентної спроможності, довгострокового розвитку. Більшість спеціалістів формулюють сучасну концепцію управління персоналом (людськими ресурсами) достатньо широко, підкреслюючи її відмінності: за критеріями оцінки ефективності (більш повне використання потенціалу співробітників, а не мінімізація витрат), за ознакою контролю (самоконтроль, а не зовнішній нагляд), за переважною формою організації (органічна, гнучка, а не бюрократична централізована).

При цьому відмічається наявність тенденції до підвищення ролі аналітичних функцій кадрових підрозділів, для чого їм потрібен постійний доступ до значних масивів даних, що стосуються співробітників підприємства, організації або установи, її систематизації та аналізу, в тому числі за допомогою спеціалізованих програмних засобів.

Під час виконання даного курсового проекту необхідно розробити інформаційну базу даних для підрозділу кадрів підприємства, яка допоможе будь-якому користувачеві легко знайти потрібну йому інформацію щодо стану кадрового забезпечення окремих структурних підрозділів підприємства, редагувати, добавляти, видаляти та здійснювати пошук окремих співробітників підприємства.

Ефект від використання зазначених програмних продуктів в роботі кадрових підрозділів є високим незалежно від загальної кількості співробітників. Їх запровадження суттєво знижує час, що потребується для обробки інформації та підготовки кадрових документів.

1. АНАЛІЗ ПРЕДМЕТНОЇ ОБЛАСТІ

Як вже було зазначене раніше, метою курсового проекту є розробка програмного продукту для обробки даних співробітників підприємства. На даний час ринок вказаних систем кадрового забезпечення достатньо насичений. Так, одними з найвідоміших програми для роботи кадрових підрозділів є:

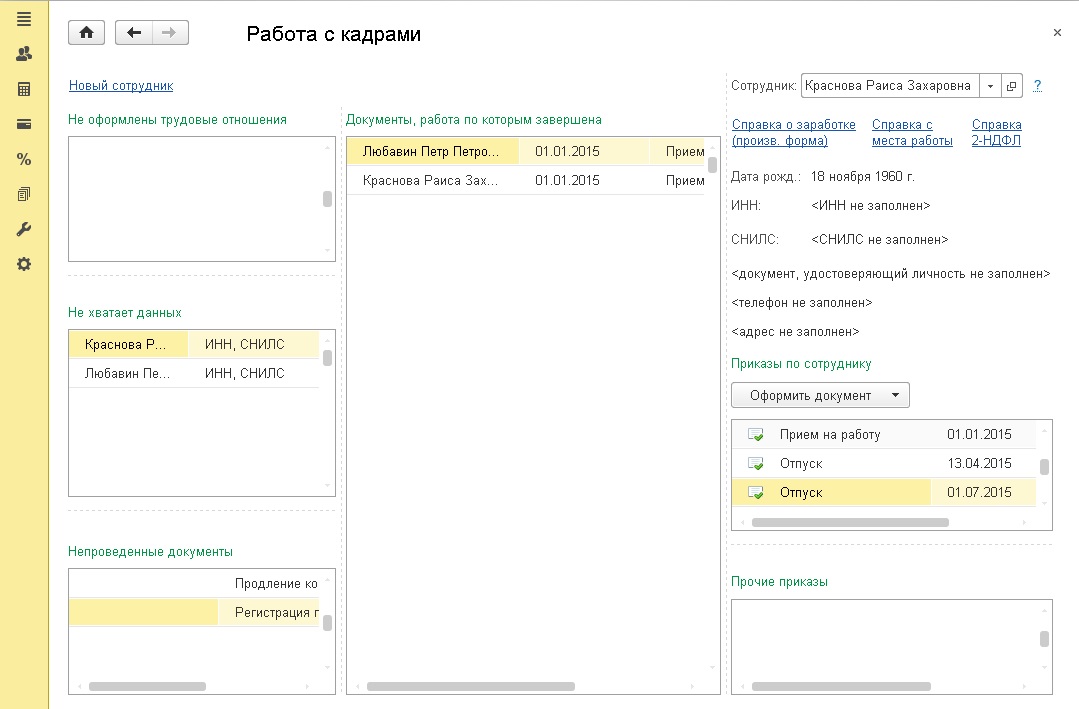
* «1С Кадры» - досить поширений програмний продукт, який має додатковий функціонал у вигляді розрахунку заробітної платні (Рисунок 1.1);

Рисунок 1.1 Зовнішній вигляд програми «1С Кадры»

* «Отдел кадров 5.7» - має додатковий функціонал щодо автоматичного формування кадрових документів, розробник ООО «К-софт»;
* «SAP R3, модуль HR.8» - програмний продукт SAP R3 який має модульну структуру, що охоплює різноманітні бізнес-процеси підприємства (Рисунок 1.2);

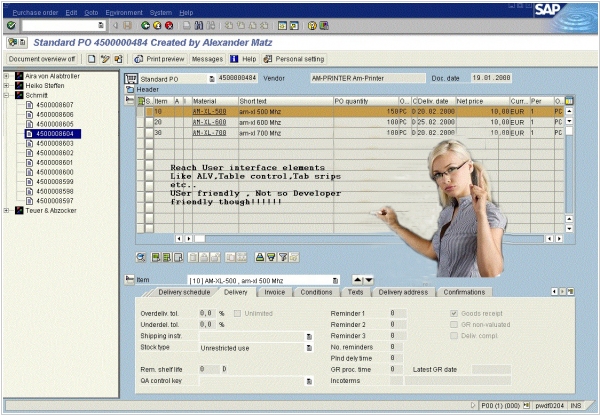


Рисунок 1.2 Головний модуль програми SAP R3

* «Кадровый учет» - багатофункціональний програмний продукт з додатковими функціями щодо автоматичного формування кадрових документів, обліку відпусток та ін., розробник компанія «PVE».

Загальним недоліком перелічених програмних продуктів є те, що їх розробники знаходяться у Російській Федерації, а отже можливі ускладнення при здійсненні технічної підтримки та оновленні програмних продуктів. Крім того, ці системи розроблені з урахуванням вимог законодавства РФ, що можуть не відповідати нормативно-правовим нормам України.

2. Постановка завдання

2.1 Призначення програмного продукту

Загальною метою створення програмного продукту є підвищення ефективності використання усіх ресурсів підприємства, установи або організації. Організація роботи із вчасного отримання інформації про результати діяльності підприємства, а також інформації про роботу окремих відділів та підрозділів, про недоліки в роботі працівників, погіршення продуктивності праці, виявлення причин та визначення заходів щодо їх усунення дасть підприємству змогу оптимально використовувати кадрові резерви, що сприятиме досягненню підприємством найбільш значних результатів.

На даному етапі розвитку економіки є особлива необхідність в організації ефективної роботи із систематизації та автоматизації інформації, що стосується управління персоналом та спрямовує наявні та потенційні компетенції персоналу на досягнення найвищих результатів.

Таким чином, база даних кадрового підрозділу є важливим інструментом для створення та регулювання загальної комплексної роботи кадрового забезпечення підприємства.

Програмний продукт, що буде створений під час виконання курсової роботи, призначений для використання співробітниками кадрових підрозділів будь-якого підприємства та забезпечувати базу для ефективного використання кадрових ресурсів підприємства.

2.2 Функціональні вимоги до програмного продукту

Проектний програмний продукт має містити інформацію щодо відділів підприємства, співробітників підприємства, керівництві кожного відділу та здійснювати наступні операції зі вказаними даними:

* зберігати дані у файловій системі;
* здійснювати пошук даних за ключовим словом або його частиною;
* надавати можливість редагування даних;
* надавати можливість додавання нових даних;
* надавати можливість видалення даних.

При цьому зазначені операції мають проводитись у графічному інтерфейсі.

3 РЕАЛІЗАЦІЯ

3.1 Огляд методів та засобів проектування програмного продукту

У даній курсовій роботі був використаний Microsoft Visio 2010 SP2 MSO – набір засобів для моделювання інформаційних систем та бізнес-проектів, проектування коду програмних продуктів на базі моделей UML. Microsoft Visio 2010 – це система управління вимогами, що підтримує повний цикл розробки програмного продукту – аналіз, дизайн архітектури, розробка програмного коду, а також забезпечує організацію одночасної розробки командою програмістів одного проекту.

Для розробки програмного коду була використана Microsoft Visual Studio Community 2017 версії 15.9.4 та інтегрований до неї Windows Forms – інтерфейс програмування додатків API (application programming interface), який відповідає за графічний інтерфейс користувача, платформа .NET FRAMEWORK 4.7. Мовою програмування був обраний Visual C# 7.

3.2 Системні вимоги

Системні вимоги до функціонування створеного програмного продукту обумовлені системними вимогами до програмного забезпечення, за допомогою якого було виготовлено програмний продукт:

* операційна система Windows 7 SP1 або вища;
* оперативна пам'ять 2 Gb RAM або більше;
* процесор 1.2 GHz, Dual-core або краще;
* відеокарта, яка підтримує мінімальне розрішення 720р (1280х720).
  1. Концептуальна схема у вигляді діаграм

3.3.1 Діаграма варіантів використання

Діаграма варіантів використання (Use Case Diagram) зображує функціональне призначення системи, або, іншими словами, те, що система має робити в процесі свого функціонування.

Діаграма варіантів використання є базовим концептуальним уявленням або моделлю в процесі її проектування та розробки.

Розробка діаграми варіантів використання переслідує наступні цілі:

* визначення загальних границь та контексту системи та предметної області, що моделюється на первинних етапах проектування системи;
* формулювання загальних вимог до функціональної поведінки системи, що моделюється;
* розробка вихідної концептуальної моделі системи для її подальшої деталізації у формі логічних та фізичних моделей;
* підготовка вихідної документації для взаємодії між розробниками системи та замовниками та користувачами.

Діаграма використання наявно демонструє сервіси, що надаються користувачеві. Вона показана на рисунку 3.1.

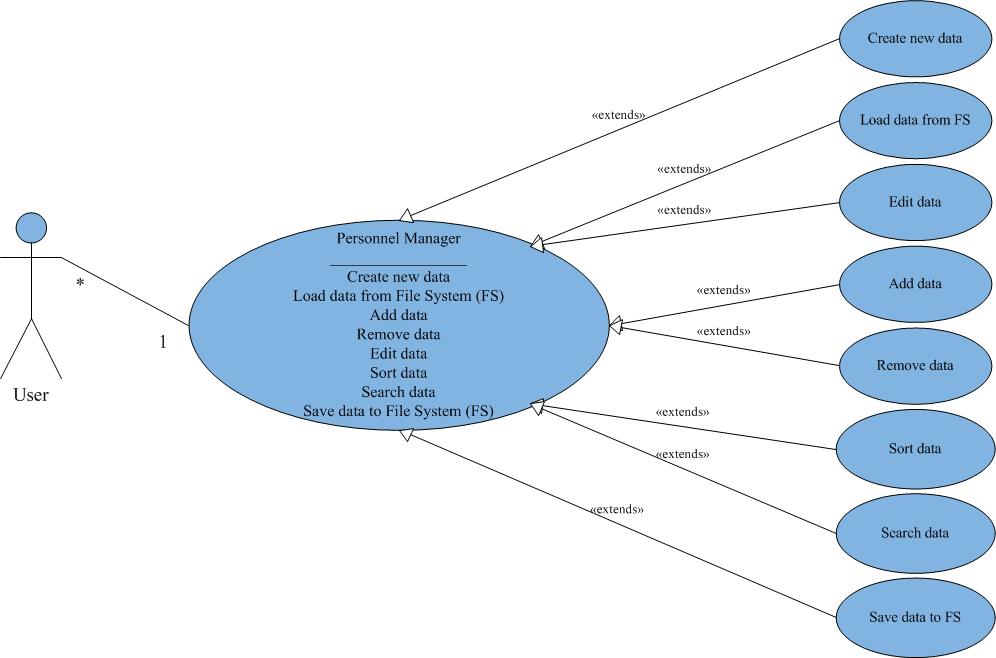


Рисунок 3.1 Діаграма використання

3.3.2 Діаграма класів

Діаграма класів (Static Structure diagram) зображує загальну структуру ієрархії класів системи, їх кооперацій, атрибутів, методів, інтерфейсів та зв’язку між ними.

Цей вид діаграм широко застосовується для конструювання шляхом прямого або зворотного проектування.

Метою створення діаграми класів є графічне представлення статичної структури декларативних елементів системи (класів, типів та ін.).

Діаграма класів наведена на рисунку 3.2.

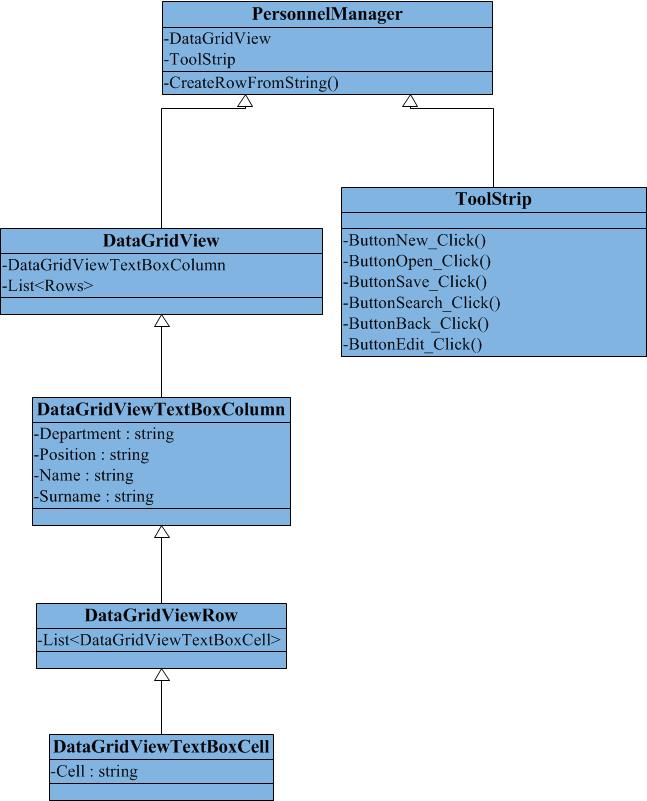


Рисунок 3.2 Діаграма класів

3.3.3 Діаграма взаємодії

Діаграма послідовності (Sequence diagram) – діаграма, на якій зображена взаємодія об’єктів, впорядкована відповідно до часу їх з’явлення. На діаграмі наявна незображена вісь часу, що дозволяє зобразити часові відносини між повідомленнями, що передаються між об’єктами.

Також на діаграмі послідовності зображуються об’єкти, що безпосередньо беруть участь у взаємодії, при цьому ніякі статичні зв’язки з іншими об’єктами не зображуються. Ключовим моментом є саме взаємодія об’єктів у часу.

Діаграма взаємодії наведена на рисунку 3.3.

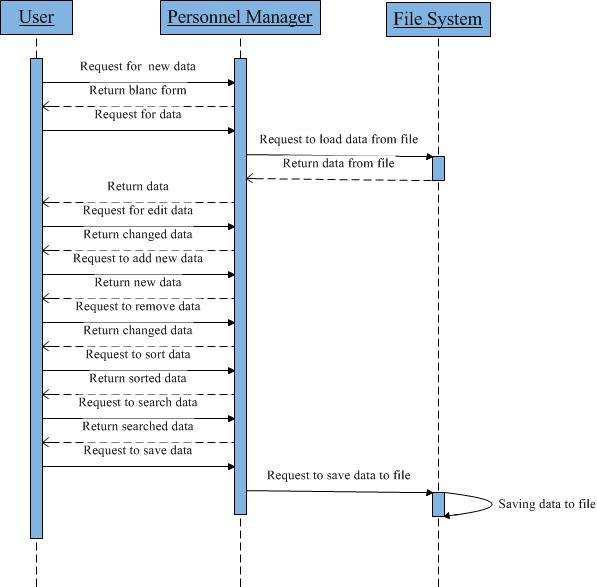


Рисунок 3.3 Діаграма взаємодії

Таким чином, під час обробки даних співробітників підприємства ці дані знаходяться у пом’яті самого продукту, тобто не змінюються у начальному файлі до тих пір, доки вони не будуть туди записані за командою користувача. Такий алгоритм обробки є запобіжним заходом від випадкового пошкодження даних. Уразі випадкової втрати будь-якої інформації у будь-який час є можливість загрузити первинні дані з файлу.

* 1. Вибір засобів для зберігання даних

У якості засобу для зберігання даних, що оброблюються програмним продуктом, обрана файлова система. Таким чином усі дані перед початком роботи загружаються з текстового файлу, а після роботи зберігаються у текстовому файлі (будь-якому файлі з розширенням .txt). При цьому запис даних та їх зчитування здійснюється рядками, з метою розподілу між окремими словами застосовується символ ‘,’. Зразок вмісту файлу на рисунку 3.4.

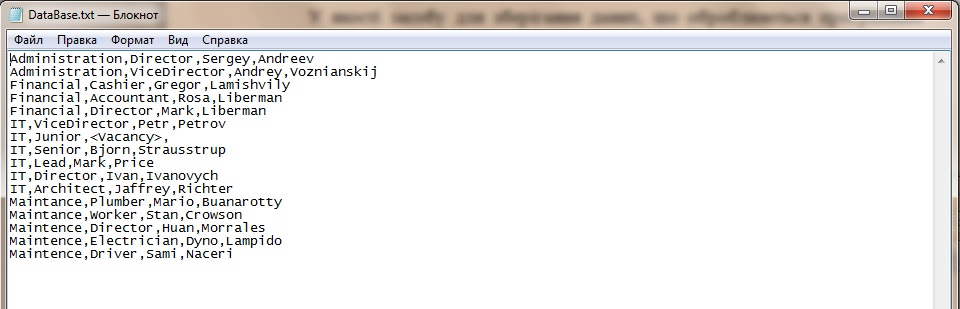


Рисунок 3.4 Вміст файлу DataBase.txt

Така система була обрана у зв’язку з її надійністю (простота створення резервних файлів) та незалежністю від наявності та швидкості зв’язку з дискретною базою даних. Крім того, в умовах розвитку обсягів накопичувачів даних, зникає проблема з розміром бази даних. В подальшому, для захисту персональних даних від несанкційованого копіювання, можливо введення криптографічних алгоритмів при запису інформації до файлу.

4 ПОСІБНИК КОРИСТУВАЧА

Програмний продукт для обробки даних відділом кадрів розроблений за допомогою Microsoft Visual C# та Windows Forms, а отже має графічний інтерфейс, зображений на рисунку 4.1.

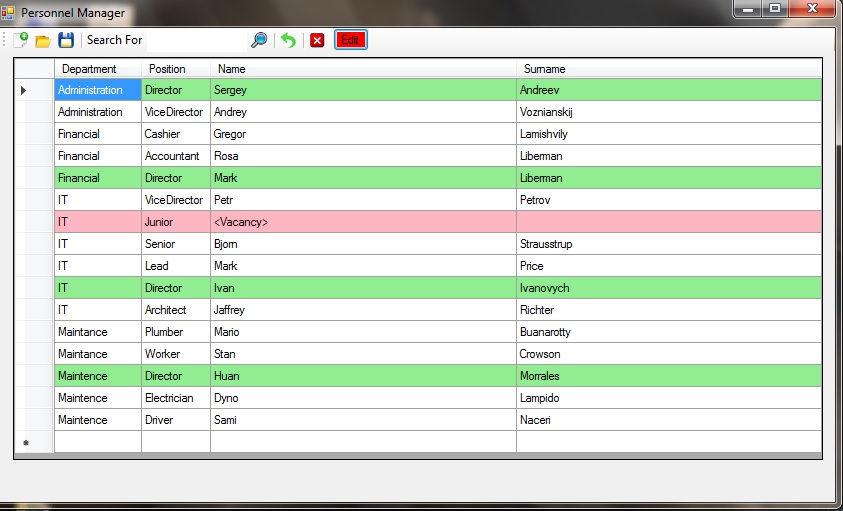


Рисунок 4.1 Зовнішній вид Personnel Manager

На вказаному інтерфейсі відображені усі основні дані: таблиця з даними щодо співробітників підприємства із зазначенням їх посади (Position та Department), ім’я та фамілії. При цьому, як видно на рисунку 4, посади начальників відділів (Director of Department) виділені зеленим кольором, а вакантні посади – рожевим. Дані можна сортувати за даними, зазначеними у шапці таблиці (Department, Position, Name та Surname). Для цього потрібно обрати, за якою категорією це буде зроблено. Сортування відбувається в алфавітному порядку або в зворотньому напрямку. Так, на зображенні дані відсортовані в алфавітному порядку за категорією Department.

Розглянемо керуючи кнопки на панелі управління (рисунок 4.2).



Рисунок 4.2. Керуючі кнопки панелі управління

 - Кнопка створення нової таблиці даних, при її натисканні буде відкрите наступне пусте вікно(рисунок 4.3)

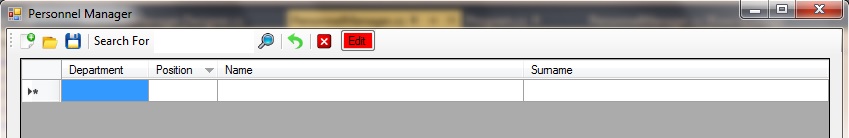


Рисунок 4.3. Відкрите нове вікно для введення нових даних

При цьому необхідно звернути увагу на колір останнього елементу панелі управління – кнопку Edit. Ця кнопка здійснює перемикання режиму редагування даних таблиці: у разі, якщо кнопка має червоний колір, дані таблиці знаходяться в стані ReadOnly, тобто редагування заборонено. Якщо натиснути на цю кнопку, вона змінює статус даних та дозволяє їх редагувати. При цьому кнопка стає зеленого кольору.

 - Кнопка загрузки даних з файлу, при її натисканні буде відкрите діалогове вікно вибору файлу для загрузки даних (рисунок 4.4)

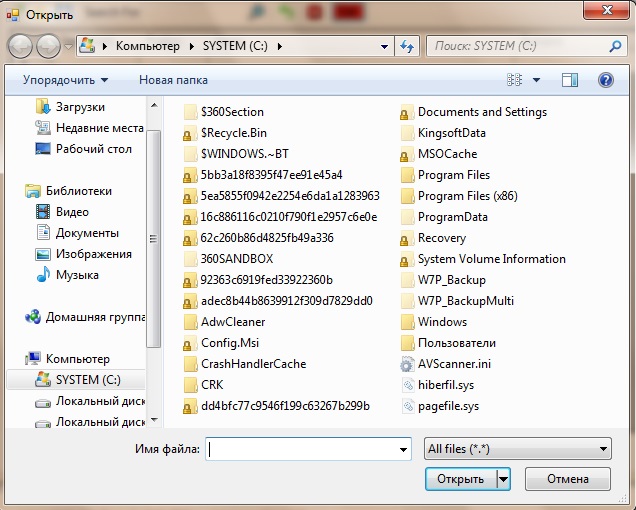


Рисунок 4.4. Діалогове вікно загрузки файлу

У даному вікні обирається файл, у якому знаходяться дані, що були збережені раніше (файл з розширенням .txt), після чого до бази будуть внесені відповідні дані (рисунок 4.1).

- Кнопка збереження даних до файлу, при її натисканні буде відкрите діалогове вікно вибору файлу для збереження даних (рисунок 4.5)



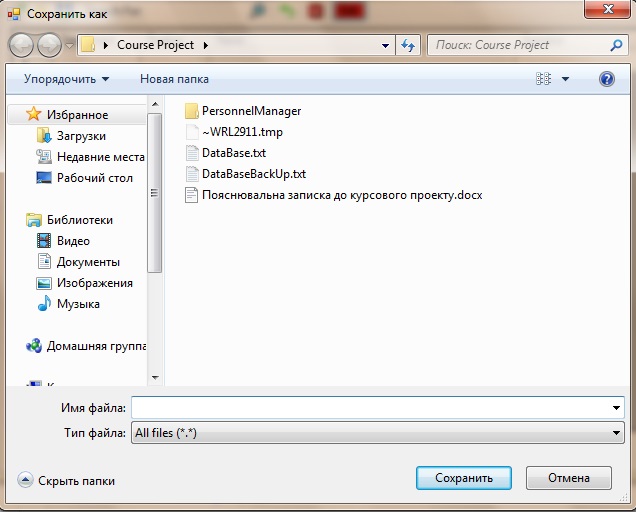


Рисунок 4.5. Діалогове вікно збереження файлу

Наступний елемент панелі управління – це блок пошуку даних:



У діалоговому вікні, що знаходиться відразу після слів «Search For» вводиться параметр(слово цілком або його частина), в даному випадку “Mar”, за яким буде здійснений пошук у даних (при цьому слід пам’ятати, що пошук чутливий до реєстру параметру), та натиснути на кнопку пошуку, що знаходиться справа від діалогового вікна. Результати виконання пошуку зображений на рисунку 4.6.

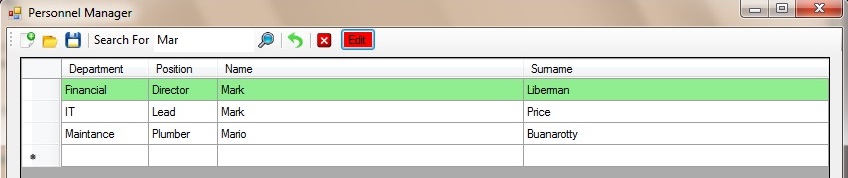


Рисунок 4.6. Результати пошуку за обраним параметром

Для повернення до робочої таблиці необхідно натиснути кнопку 

Додавання нових даних до таблиці здійснюється введенням даних у пустий рядок унизу таблиці. При цьому у разі перевищення кількості рядків розміру екрану таблиці В її правій частині автоматично з’явиться полоса прокрутки (або можна збільшити розмір вікна програми).

 - Кнопка видалення даних. В неї є два режими: вона може видаляти цілком рядок, якщо він виокремлений цілком, або будь яку виділену ячейку таблиці.

 - Кнопка редагування даних. Як вже вказувалось вище, ця кнопка здійснює перемикання режиму редагування даних таблиці: у разі, якщо кнопка має червоний колір, дані таблиці знаходяться в стані ReadOnly, тобто редагування заборонено. Якщо натиснути на цю кнопку, вона змінює статус даних та дозволяє їх редагувати. При цьому кнопка стає зеленого кольору.

ВИСНОВОК

Розроблений в ході виконання курсової роботи програмний продукт «СИСТЕМА УПРАВЛІННЯ ВІДДІЛОМ КАДРІВ» («Personnel Manager») є досить актуальним на даний час та має значний потенціал до подальшого розвитку функціональності. Він спрощує роботу користувача та дозволяє зберігати у файловій системі дані.

Під час виконання даної курсової роботи були вирішені завдання, поставлені на початку роботи. Був створений програмний продукт, до якого були включені функції зберігання даних у файловій системі, пошуку, редагування, добавлення та видалення даних у графічному інтерфейсі.

В подальшому функціонал цього продукту можливо розширити за рахунок введення:

* додаткових персональних даних (дати народження, дати та номеру наказу про прийняття на роботу, відомостей про освіту, спеціальність та ін.);
* введення авторизації у програмі з відповідним розподілом повноважень (адміністратор, користувач системи – співробітник кадрового підрозділу);
* криптографічний захист особистих даних працівників підприємства;
* введення додаткової бази даних кандидатів на вакантні посади;
* виведення даних на друк та ін.

ПЕРЕЛІК ДЖЕРЕЛ ПОСИЛАННЯ

1. Герберт Шилдт. Полное руководство С# 4.0. Пер. с англ. – Москва. Санкт-Петербург. Киев: Вильямс, 2011. – 1056 стр.
2. Джефри Рихтер. CLR via C# Программирование на платформе Microsoft .Net FRAMEWORK 2.0 на языке C# Мастер-класс. Пер. с англ. - Москва. Санкт-Петербург. Нижний Новгород. Ростов-на-Дону. Екатеринбург. Самара. Киев. Харьков. Минск: Русская редакция: Питер, 2007. – 636 стр.
3. Марк Дж. Прайс. C# 7 и .NET Core Кросс-платформенная разработка для профессионалов. 3-е издание. Пер. с англ. - Санкт-Петербург. Москва. Екатеринбург. Воронеж. Нижний Новгород. Ростов-на-Дону. Самара. Минск: Питер, 2018. – 636 стр.
4. Microsoft Developer Network (MSDN). URL: <https://docs.microsoft.com/ru-ru/visualstudio/> (дата звернення: впродовж написання коду курсового проекту).
5. METANIT.COM Сайт о программировании.

URL: [https://metanit.com/sharp/](https://metanit.com/sharp/tutorial/) (дата звернення: 25.11.2018).

1. «1С Кадры».URL: <http://v8.1c.ru/hrm/kadrovyj_uchet/> (дата звернення: 23.10.2018 року)
2. «Отдел кадров 5.7».URL: [www.ksoft.ru](http://www.ksoft.ru) (дата звернення: 23.10.2018 року)
3. «SAP R3». URL: <http://www.clouderp.ru/tags/sap_r3_erp/> (дата звернення: 23.10.2018 року)
4. «Кадровый учет». URL: <http://www.softbis.narod.ru> (дата звернення: 23.10.2018 року)

ДОДАТОК А

Фрагменти коду програми

У програмі використовуються 6 класів (рисунок 2):

1. PersonnelManager
2. ToolStrip
3. DataGridView
4. DataGridViewTextBoxColumn
5. DataGridViewRow
6. DataGridViewTextBoxCell

При цьому явно оголошується лише перший клас:

**public partial class**

**PersonnelManager : Form**

**{**

**}**

Решта 5 класів надаються бібліотекою System.Windows.Forms.Усі методи та функції, що використовуються у програмі, також надаються цією бібліотекою.

У якості засобу зберігання даних, що оброблюються програмним продуктом, був обраний текстовий файл. При цьому дані кожного рядку таблиці записувалися рядками, у якості розподільного знаку між даними різних колонок використовувався символ ‘,’ Для запису до файлу даних та їх загрузки були написані відповідні функції:

загрузки з файлу:

**private void ToolStripButtonOpen\_Click(object sender, EventArgs e)**

**{**

**var filePath = string.Empty;**

**// Showing FileOpenDialog**

**using (OpenFileDialog openFileDialog = new OpenFileDialog())**

**{**

**openFileDialog.InitialDirectory = "c:\\";**

**openFileDialog.Filter = "txt files (\*.txt)|\*.txt|All files**

**(\*.\*)|\*.\*";**

**openFileDialog.FilterIndex = 2;**

**openFileDialog.RestoreDirectory = true;**

**if (openFileDialog.ShowDialog() == DialogResult.OK)**

**{**

**// Get the path of specified file**

**filePath = openFileDialog.FileName;**

**// Clearing DataGridView**

**dataGridViewData.Rows.Clear();**

**// Read the contents of the file into a stream**

**Stream fileStream = openFileDialog.OpenFile();**

**using (StreamReader reader = new StreamReader(fileStream))**

**{**

**// Filling DataGridView string by string**

**while (!reader.EndOfStream)**

**{**

**// Exception handling**

**try**

**{**

**CreateRowFromString(reader.ReadLine());**

**}**

**catch (Exception ex)**

**{**

**MessageBox.Show(ex.ToString());**

**}**

**}**

**// Removing StreamReader**

**reader.Close();**

**}**

**}**

**}**

**// Making DataGridView Accessible for Edit.**

**dataGridViewData.Enabled = true;**

**}**

запису до файлу:

**private void ToolStripButtonSave\_Click(object sender, EventArgs e)**

**{**

**using (SaveFileDialog saveFileDialog = new SaveFileDialog())**

**{**

**saveFileDialog.Filter = "txt files (\*.txt)|\*.txt|All files (\*.\*)|\*.\*";**

**saveFileDialog.FilterIndex = 2;**

**saveFileDialog.RestoreDirectory = true;**

**if (saveFileDialog.ShowDialog() == DialogResult.OK)**

**{**

**// Read the contents of the file into a stream**

**Stream fileStream;**

**if ((fileStream = saveFileDialog.OpenFile()) != null)**

**{**

**// StreamWriter writing**

**using (StreamWriter writer = new StreamWriter(fileStream))**

**{**

**foreach (DataGridViewRow row in dataGridViewData.Rows)**

**{**

**// If there is no data in the first column - then don't use this line**

**if (row.Cells[0].Value == null)**

**continue;**

**string tmp\_str = "";**

**// Fill line cell by cell to one string.**

**for (int i = 0; i < row.Cells.Count; ++i)**

**{**

**tmp\_str += row.Cells[i].Value;**

**if (i != row.Cells.Count - 1)**

**tmp\_str += ",";**

**}**

**// Write string to file.**

**writer.WriteLine(tmp\_str);**

**}**

**// Removing StreamWriter**

**writer.Close();**

**}**

**fileStream.Close();**

**}**

**}**

**}**

**}**

З метою запису даних, завантажених з файлу, до рядків таблиці, була написана наступна функція:

**private void CreateRowFromString(string data)**

**{**

**// Creating new row based on first row of the DataGridView.**

**DataGridViewRow row = (DataGridViewRow)dataGridViewData.Rows[0].Clone();**

**row.DefaultCellStyle.BackColor = Color.White;**

**// Split parsed string into words.**

**string[] words = data.Split(new Char[] { ',' });**

**if (words == null)**

**{**

**throw new ArgumentNullException("No correct data for Table");**

**}**

**if (DataColumnsCount < words.Count())**

**{**

**throw new ArgumentOutOfRangeException("More fields than supported by table");**

**}**

**// First two words are used as Department and Position fields.**

**row.Cells[0].Value = words[0];**

**row.Cells[1].Value = words[1];**

**// Making Director rows LightGreen for easier search.**

**if (words[1] == "Director")**

**{**

**row.DefaultCellStyle.BackColor = Color.LightGreen;**

**}**

**// If Position is empty - set Name <Vacancy>.**

**// Making Vacancy rows LightPink for easier search.**

**if ((words[2].Length != 0) && (words[2] != "<Vacancy>"))**

**{**

**row.Cells[2].Value = words[2];**

**row.Cells[3].Value = words[3];**

**}**

**else**

**{**

**row.Cells[2].Value = "<Vacancy>";**

**row.Cells[3].Value = "";**

**row.DefaultCellStyle.BackColor = Color.LightPink;**

**}**