

Міністерство освіти і науки України
Сумський державний університет
Навчально-науковий інститут бізнес-технологій «УАБС»
Кафедра економічної кібернетики

КВАЛІФІКАЦІЙНА МАГІСТЕРСЬКА РОБОТА

на тему «Автоматизація кадрового обліку підприємництва»

Виконав студент 2 курсу, групи ЕК.м-61а
(номер курсу) (шифр групи)

Спеціальності 051 «Економіка («Економічна
кібернетика»))»

Таран Д. І.
(прізвище, ініціали студента)

Керівник к.т.н., доцент Яценко В.В.
(посада, науковий ступінь, прізвище, ініціали)

Суми – 2018 рік

ЗМІСТ

ВСТУП	6
РОЗДІЛ 1 ДОСЛІДЖЕННЯ СТАНУ ОБ’ЄКТА АВТОМАТИЗАЦІЇ ТА ФОРМУВАННЯ ВИМОГ ДО СИСТЕМИ	8
1.2 Характеристика об’єкта автоматизації	8
1.2 Аналіз стану автоматизації об’єкта дослідження.....	12
1.3 Формування вимог до системи автоматизації.....	20
РОЗДІЛ 2 ПРОЕКТУВАННЯ АВТОМАТИЗОВАНОЇ СИСТЕМИ.....	21
2.1 Моделі бізнес-процесів задачі.....	21
2.2 Інформаційна архітектура та технології вирішення задачі.....	25
2.3 Функціональна структура задачі.....	29
2.4 Підсистеми забезпечення функціональної частини	37
РОЗДІЛ 3 РЕАЛІЗАЦІЯ ПРОТОТИПУ АВТОМАТИЗОВАНОЇ СИСТЕМИ.....	42
3.1 Структура та особливості реалізації інформаційного забезпечення	42
3.2 Структура та особливості реалізації алгоритмічного забезпечення.....	48
3.3 Контрольний приклад та інструкція з використання	52
3.4 Оцінка очікуваного ефекту від впровадження системи автоматизації	62
ВИСНОВКИ.....	69
ДОДАТКИ.....	74

ВСТУП

Одним з факторів, що визначають рівень розвитку сучасного суспільства і його інтелектуальні можливості, є оснащеність його засобами обчислювальної техніки – основи автоматизації розумової діяльності людини. Сфера використання комп'ютерів в даний час настільки широка, що немає такої області, де застосування ПК було б недоцільним. В даний час ПК широко застосовуються в багатьох галузях діяльності людини. Жодна фірма не може обійтися у своїй роботі без застосування комп'ютерів які з успіхом замінюють рутинну роботу, яка виконувалася раніше в ручну, підвищуючи ефективність своєї роботи. Особливо важлива роль комп'ютерів для розвитку науки, зростання промислового виробництва і підвищення ефективності управління.

Управління персоналом є основним ресурсом кожного підприємства та сьогодні стає завданням першорядної важливості. Створення умов для залучення і утримання кращих співробітників, своєчасна підготовка кадрових ресурсів для задоволення потреб бізнесу – це необхідні складові успіху будь-якої компанії. Але для того щоб мати можливість приймати вірні управлінські рішення, потрібно володіти актуальною, повною і достовірною інформацією про персонал, який, до того ж, є найбільш непостійним, мінливим ресурсом підприємства. Можливості його використання залежать від багатьох чинників – економічних, законодавчих, демографічних. Щоб аналізувати ситуацію, керівник повинен знати про свої кадри все.

Оскільки облік та контроль за наявністю та рухом кадрів є одним з найскладніших та трудомістких частин обліку праці та пов'язаний з обробкою великих обсягів облікової інформації в останні роки цей процес найчастіше автоматизують використанням обчислювальних можливостей комп'ютерної техніки. При автоматизації за допомогою інформаційної системи зменшуються обсяги ручної праці, зменшується час на обробку та отримання необхідної інформації, зростає продуктивність роботи працівників.

Успішне вирішення цієї задачі автоматизації багато в чому залежить від раціональної організації інформаційного забезпечення, яке дозволяє вирішувати

такі проблеми як: централізоване керування даними, інформаційна сумісність, гнучкість та ефективна актуалізація інформаційної бази.

Об'єктом дослідження у кваліфікаційній роботі є діяльність мережі аптек, пов'язана з обліком наявності та руху кадрів.

Предметом дослідження є інформаційна система автоматизації обліку кадрів на підприємстві.

Для досягнення мети кваліфікаційної роботи треба розв'язати такі завдання:

- охарактеризувати об'єкт автоматизації;
- проаналізувати стан об'єкту автоматизації;
- розробити концепцію автоматизації діяльності об'єкту дослідження;
- розробити вимоги до створюваної системи, надати функціональну характеристику;
- розробити проект автоматизованої системи;
- розробити прототип системи;
- визначити економічний ефект від впровадження системи.

РОЗДІЛ 1 ДОСЛІДЖЕННЯ СТАНУ ОБ'ЄКТА АВТОМАТИЗАЦІЇ ТА ФОРМУВАННЯ ВИМОГ ДО СИСТЕМИ

1.2 Характеристика об'єкта автоматизації

Аптека «Дарина» була заснована в 2003 році приватним підприємцем. Сьогодні це аптека, яка має 4 аптечні пункти в різних містах та селищах міського типу Сумської області. В даний час в компанії працює близько 20 співробітників.

Аптека «Дарина» та її структурні підрозділи вже понад 14 років займається забезпеченням населення та лікувально-профілактичних закладів, бюджетних установ та інших підприємств лікарськими засобами і виробами медичного призначення.

Робота аптеки повністю відповідає ліцензійним умовам чинного законодавства, санітарним нормам та вимогам законодавчих актів про захист прав споживачів. Має всі дозвільні документи, сертифікати якості та відповідності товарів.

За роки свого існування аптека «Дарина» завоювала популярність і довіру жителів північних районів Сумської області. Кваліфіковані співробітники, використовуючи протоколи обслуговування покупців, завжди уважно розібравшись з проблемою людини, нададуть відповідні рекомендації та підберуть лікарські засоби, які задовольнять лікувальним та матеріальним вимогам покупця.

Розглянувши організаційну структуру аптеки «Дарина» можна казати про її та адаптованість до сучасних бізнесових та інформаційних умов. Схема організаційної структури наведено на рисунку 1.

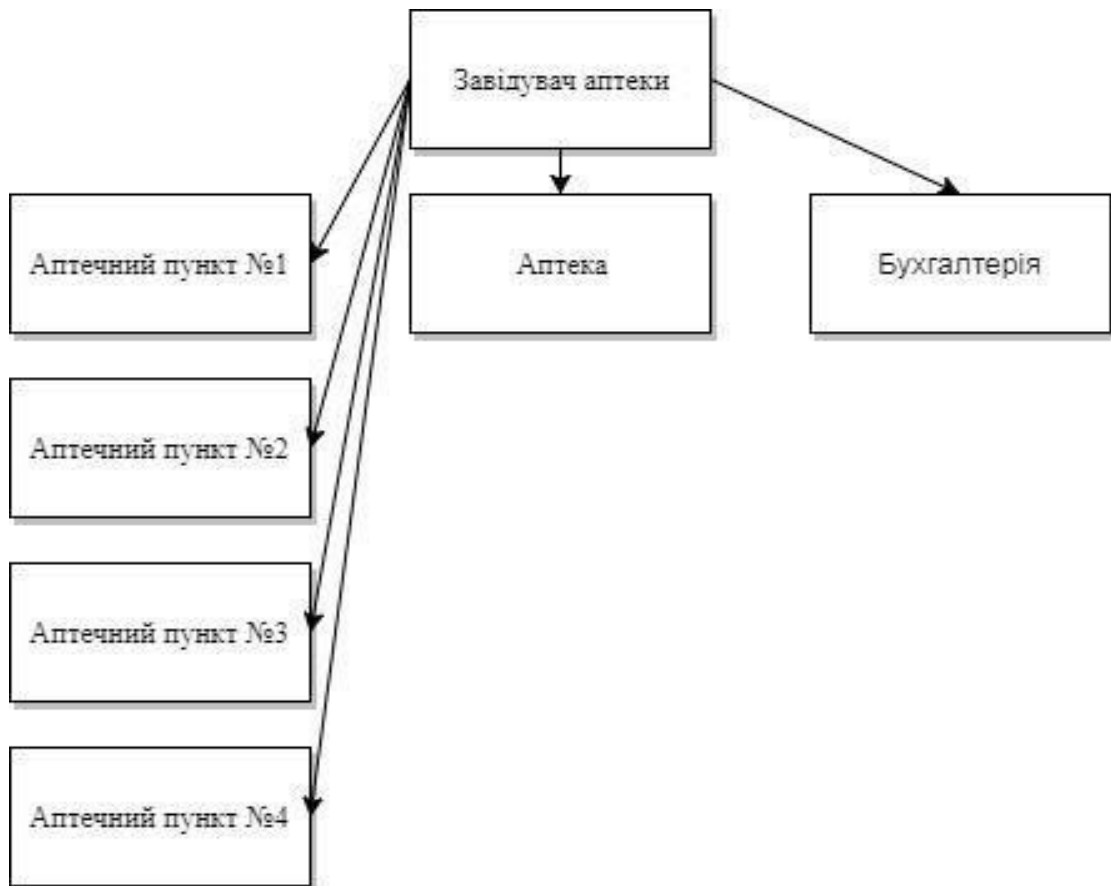


Рисунок 1.1 – Організаційна структура аптеки «Дарина»

Щодо функціональної структури, то можна виділяти такі організаційні одиниці як: аптека, бухгалтерія, 4 аптечні пункти, які розташовані в приміщеннях лікарень.

Кожний з цих відділів є ключовими для роботи підприємства. Кожний з функціональних елементів має свого керівника, який виконує координацію дій співробітників.

Керівник аптеки – завідувача, яка має вищу фармацевтичну освіту, відповідного напрямку. Їй підпорядковуються провізори аптеки, які мають відповідну вищу освіту, пройшли інтернатуру та мають гарний досвід в роботі, а також фармацевти, які мають середньо-спеціальну фармосвіту.

Аптечні пункти, очолюють завідувачі аптечними пунктами - провізори,

яким підпорядковуюються фармацевти, які працюють в аптечних пунктах. Завідуючі аптечними пунктами підпорядковуються завідуючій аптеки.

Всі провізори та фармацевти 1 раз в 5 років (відповідно до законодавства та ліцензійних умов) з метою підвищення кваліфікації співробітників, проходять курси в Харківському Національному фармацевтичному університеті, що дає можливість отримувати нові знання стосовно якості лікарських засобів, змін в чинному законодавстві та інших актуальних питаннях.

Завідуюча аптеки та головний бухгалтер звітують та підпорядковуються фізичній особі-підприємцю.

Відділ бухгалтерії – головний бухгалтер та бухгалтер. На бухгалтерію покладено обов'язки з ведення первинного, бухгалтерського, податкового та кадрового обліку (у зв'язку з відсутністю відділу кадрів). Робота бухгалтерії забезпечує постійну та своєчасну звітність в ДФС та інші фонди та служби.

Налагоджена взаємодія всіх підрозділів, відповідальне та професійне відношення співробітників, а також колективна орієнтованість на гарний фінансовий та іміджевий результат були багаторазово відмічені грамотами керівництвом району та Сумської області.

У зв'язку з відсутністю відділу кадрів ведення обліку кадрів покладено на бухгалтера. Мета обліку кадрів – одержання даних про наявність і рух працівників в цілому по підприємству і по кожному його відділу, контроль за виконанням плану по встановленій чисельності робітників та службовців, визначення витрат праці, контроль за кількістю прийнятих і звільнених працівників.

При прийнятті на роботу працівника складається особиста картка. Вона містить у собі відомості, отримані з трудової книжки з диплома, паспорта, а також дані з свідоцтва про шлюб (у разі невідповідності прізвища в паспорті та дипломі про освіту), При прийомі на роботу, відділ кадрів у відповідності з професією і кваліфікацією працівника, призначає йому посаду, відділ, в якому він буде працювати, а потім по цим даним у документі. Штатний розклад визначає заробітну плату. Посада, підрозділ, а також прізвище, ім'я та по-батькові, дата зарахування на роботу, дату проходження курсів підвищення кваліфікації,

табельний номер, який працівникові присвоюють, вказують у Наказі «Про прийняття на роботу».

Затверджений підприємцем Наказ передається у бухгалтерію.

Бухгалтер попереджує керівника підрозділу, в який призначено працівника на роботу.

Для успішної діяльності аптек із забезпечення населення та лікувально-профілактичних закладів лікарськими засобами і виробами медичного призначення необхідний постійний моніторинг основних економічних показників таких, як дохід, прибуток, витрати, торгові накладення, торгівельна націнка та інші. Для розрахунку їх абсолютних та відносних значень використовуються вихідні дані із первинних та нагромаджуючих облікових документів і матеріалів звітності. До них відносяться товарно-транспортні накладні (рахунки-фактури), рецепти, касові чеки, відомості, реєстри, журнали, звіти.

Для обліку товарів в аптеці використовується програма "PARACELS" ("ПАРАЦЕЛЬС"), для обробки прайсів від постачальників, формування заявок – програму "ROBODOK", для подання звітності в ДФС та подання податкової звітності використовується два Модулі програми М.Е.Дос - "Звітність", "Обмін документами з контрагентами", але немає жодного модуля чи програми для обліку кадрів та зарплати.

Кількість працюючих на підприємстві складає 19 чоловік. Коефіцієнт плинності кадрів складає близько 10%. Це пояснюється тим, що в колективі працюють люди віком від 20 до 67 років. Середній вік працюючого складає 44 роки. Кількість працюючих не дуже велика, але те що молоді співробітниці одружуються, та переїжджають на проживання в інші міста, або перебувають у декретній відпустці, або у відпустці по догляду за дитиною до 3-х років, а на їх місце приймаються (тимчасово) інші співробітники, а також жінки передпенсійного та пенсійного віку звільняються у зв'язку із виходом на пенсію, обумовлює великий об'єм обробки кадрової інформації та роботи.

Головними техніко-економічними показниками аптеки «Дарина» можна назвати кількість реалізованих лікарських засобів, виробів медичного призначення, БАДів та косметичних засобів

Аналізуючи економічні показники діяльності аптеки можна навести таку статистику. Річний обсяг оборотних коштів за 2016 рік складає 4033523 грн. Річний прибуток до оподаткування за 2016 рік складає 208002 грн. Аналізуючи показники можна казати про те, що аптека зайняла велику нішу на ринку продажу лікарських засобів в північних районах Сумської області [6].

1.2 Аналіз стану автоматизації об'єкта дослідження

Облік праці та заробітної плати є одним з найбільш трудомістких ділянок облікової роботи. Були вказані основні недоліки, притаманні ручному варіанту розв'язання задачі. Тому на основі вищесказаного можна зробити висновки про те, що використання засобів обчислювальної техніки дозволить значно підвищити продуктивність та ефективність розрахункових операцій та операцій пошуку даних. На даний момент існує досить багато варіантів спеціалізованого програмного забезпечення, за допомогою якого ці задачі можуть бути розв'язані за досить короткий час. Розглянемо деякі з них [8].

Основним завданням додатку "Облік кадрів" є ведення облікової роботи за персоналом в компанії за допомогою автоматизованого виконання операцій по прийому, переміщенню та звільненню працівників, а також по підготовці та обліку наказів, формування різноманітних списків і звітних документів (без обмеження їх кількості).

Додаток "Облік кадрів" дозволяє користувачам отримувати довільні звіти з будь-якої кадрової інформації, введеної в автоматизовану систему управління персоналом. Наприклад, інформація про відпустки, лікарняні, що вводиться в модулі "Облік кадрів", є вихідною для точного та оперативного розрахунку заробітної плати співробітників підприємства. Загалом дуже зручна та проста у застосування автоматизована система, але має ряд недоліків:

- додаток робочого столу, тобто відсутній багатокористувацький підхід;
- в процесі роботи спостерігається надмірність інформації;
- великі ресурси для роботи з програмою.

«Облік кадрів 3.1» – програма, що призначена для обліку кадрів на

підприємстві. В програмі представлені сучасні види звітності, та оновлена правова база, що відповідає законодавству. Вбудований VBScript дозволяє ефективно управляти даними. Серед недоліків програми можна виділити те, що відсутня можливість паралельного доступу до файлів, а також проблеми узгодження даних з кількох автоматизованих робочих місць. Також програма призначена лише для підприємств малого бізнесу і не здатна до розширення;

«Парус-Персонал» – автоматизована система управління персоналом для комплексної автоматизації бізнес-процесів HR-департаменту та відділу кадрів.

Використання системи значно підвищує ефективність управління персоналом і полегшує HRM-процеси (Human Resource Management), в т.ч. завдання рекрутингу, мотивації, підвищення кваліфікації та ін.

Функціонал системи дозволяє автоматизувати:

Рекрутинг:

- в системі анкети співробітників з можливістю подальшого їх виведення на друк;
- формувати добірку анкет згідно вказаними параметрами;
- автоматизувати процес прийому на роботу кадрів на підставі вже наявних анкет.

Кадровий облік:

- створювати штатний розклад по співробітниках з можливістю створення вкладених (підлеглих) підрозділів;
- вести в системі стандартні кадрові накази з їх автоматичною відпрацюванням (на прийом, звільнення, переміщення, відпустки тощо) і можливістю друку як в стандартному текстовому варіанті так і у вигляді деяких стандартних форм (форма розпорядження на відпустку і т.д.);
- вести табеля співробітників;
- виконувати дії, пов'язані з переміщенням співробітників, прийомом на роботу, звільненням, виведенням за штат, наданням відпустки тощо. Дані по співробітниках автоматично відображаються в історії призначень і переміщень, до якої завжди можна звернутися;

– вести облік інформації про курсову підготовку, атестації і відпустки співробітників, виконання ними посад і переміщення усередині організації, а також облік про військовозобов'язаних.

HRM:

– ведення обліку стягнень та заохочень;
– можливість інтеграції з системами контролю доступу в приміщення і системою нарахування заробітної плати.

Звітність:

– штатний розпис у розрізі посад і співробітників;
– звіт по руху співробітників;
– звіт за стажем роботи;
– звіт за вакантними посадами (форма 3-ПН);
– перелік дітей для надання допомоги.

У системі також передбачена можливість друку статистичних звітів: звіт за формами 1-ПВ, 3-ПВ, 6-ПВ, 10-1, 9-ДС. Також є можливість отримати відомості по військовозобов'язаним, укомплектованості і вакантним посадам на підприємстві й багато інших звітів.

Система "Парус-Персонал" дає можливість отримувати користувальницькі звіти облікового характеру на підставі відібраних співробітників по тим чи іншим умовам відбору, слід зазначити, що поля, що включаються в такий звіт, вибираються користувачем самостійно, що робить систему більш гнучкою та функціональною. Налаштовані користувачем звіти зберігаються в базі даних і при подальшому їх виклику досить вказати ім'я звіту для виведення на друк.

Шаблони для друку документів містяться у форматі Excel, тому при необхідності є можливість їх редагування.

Недоліком цієї системи є її ціна для підприємців.

«Беліса Software: Облік кадрів» – програма дозволяє автоматизувати облік кадрів на підприємствах.

Можливості програми:

– дозволяє вести всі форми обліку як при прийомі на роботу так і по поточній роботі;

- автоматичний розрахунок стажу за кожним працівником (загальний, поточний, в займаній посаді);

- враховуються всі дані по працівнику включаючи і його фотографію з можливістю роздрукувати його особисту справу разом з фото.

Форми кадрового обліку програми «Belica Software: Облік кадрів»:

- особовий листок з обліку кадрів;
- доповнення до особового листка;
- список працівників;
- список працівників і їх домашні і робочі телефони;
- список працівників з фотографією і короткою характеристикою;
- список працівників і їх домашні адреси;
- список працівників і їх дати народження.
- список військовозобов'язаних працівників.;
- список працівників та їх утворення.

Ця програма має ряд недоліків, що пов'язані з паралельною роботою кількох працівників, і виходячи з цього, надмірністю інформації, а також обмежена кількість звітів та статистики, що надано розробниками.

Програма «СКАРБ Зарплата» («Scarb Salary») призначена для ефективного та швидкого заповнення необхідної інформації про працівників, побудови дерева штатних розкладів, створення необхідних кадрових наказів, розрахунку заробітної плати і формування необхідних звітів і вихідних документів.

Програма «СКАРБ Зарплата» – мережева, реалізована на платформі SQL-сервера, призначена для малих, середніх і великих підприємств.

Програма дозволяє вести бази даних працівників, проводити розрахунок зарплати на підприємствах, які мають складну розгалужену структуру, підрозділи яких територіально віддалені один від одного.

Декілька користувачів одночасно можуть працювати з базою даних працівників, штатним розкладом, формувати кадрові накази, проводити розрахунки і формувати вихідні документи для різних структурних підрозділів чи відділень.

Програма проста у використанні, всі необхідні параметри, довідники та

основні види нарахувань і утримань налаштовані розробником.

Всі трудомісткі операції автоматизовані, при цьому є можливість ручного введення та коригування розрахункових даних.

Програма може гнучко перебудовуватися і легко адаптуватися до зміни алгоритмів розрахунку зарплати, констант (законодавчо змінених параметрів), звітів у відповідності з чинним законодавством України.

Недоліками цієї програми є великі витрати ресурсів та дороговизна програмного рішення та налаштування, супроводу в процесі експлуатації.

«Конфігурація MASTER:Зарплата і Кадри» програма покликана автоматизувати задачі управління персоналом і розрахунку заробітної плати на підприємствах малого та середнього бізнесу. Програмний продукт відповідає всім вимогам чинного законодавства України та є повністю адаптованим до українського ринку.

Конфігурація MASTER:Зарплата і Кадри включає в себе два багатofункціональних модуля:

МОДУЛЬ «ЗАРПЛАТА» передбачає створення та ведення банку даних (нормативно-довідкових та оперативних масивів інформації), що містить необхідну для розрахунку та аналізу заробітної плати інформацію та забезпечує зберігання розрахованих нарахувань та утримань.

Функціональні можливості модуля «ЗАРПЛАТА»:

1. Нарахування

– погодинної та відрядної заробітної плати, як за основною, так і сумісною посадами;

– надбавок та доплат, разових нарахувань та умовно-постійних нарахувань;

– допомоги по соціальному страхуванню (в тому числі лікарняних листів) та відпускних виплат;

– різних видів премій (за поточний місяць, за минулий місяць, за квартал, за півріччя, за рік);

– введення нарахувань за минулі та майбутні періоди з подальшим автоматичним перерахунком податків;

– автоматичне нарахування індексації.

2. Утримання

- податку з доходів фізичних осіб, профспілкового внеску та інших обов'язкових утримань в автоматичному режимі;
- постійні утримання: за виконавчими листами, кредитні внески, квартплата, комунальні послуги тощо;
- розподіл розрахованих нарахувань та утримань за шифрами виробничих витрат (ШВВ) та балансовими рахунками;
- розрахунок відрахувань з підприємства (ЄСВ);
- перерахунок податку з доходів фізичних осіб за підсумками року;
- формування проводок по нарахуванню / утриманню заробітної плати, відрахуваннях з підприємства та за виплатою заробітної плати;
- формування платіжних відомостей на виплату авансу, заробітної плати, відпусток та інших виплат (на банк та на касу).

3.Звітність

- формування квартальної звітності по формі 1-ДФ (в електронних та паперових копіях);
- підготовка баз даних для персоніфікованого обліку та подальшої обробки в АРМ Пенсійного фонду;
- формування та друк вихідних документів: розрахункових листів, платіжних відомостей, зведень, реєстрів, довідок.

МОДУЛЬ «КАДРИ» автоматизує роботу відділу кадрів, відділу праці та заробітної плати, забезпечує збір й аналіз інформації про персонал підприємств, автоматизує документи кадрового обліку.

Функціональні можливості модуля «КАДРИ»:

1. Штатний розклад:

- довідники для формування штатного розкладу;
- ведення штатного розкладу, зв'язок між даними кадрів та штатним розкладом;
- майстер масових змін штатного розкладу;
- планування штатного розкладу;
- штатний розклад та зарплата;

- звіти по штатному розкладу;
- планування потреби в працівниках.

2. Табелі обліку часу:

- ведення таблицю обліку часу;
- таблиці обліку часу відкривається на кожен розрахунковий місяць;
- можливість закривати таблиці для розрахунку авансу.

3. Кадри:

- автоматичне формування особової картки;
- призначення та переміщення працівників;
- дані про освіту працівників;
- підвищення кваліфікації;
- склад сім'ї, родичі;
- іноземні мови;
- нагороди та почесні звання;
- участь в ліквідації наслідків аварії на ЧАЕС;
- історія зміни прізвища;
- членство в профспілці;
- порушення трудової дисципліни;
- перепустки;
- відпустки;
- лікарняні;
- документи за середнім;
- періоди використання відпусток;
- пільги;
- інвалідність;
- стаж;
- спецстаж.

4. Накази:

- картотека наказів на прийом, переведення та звільнення працівників, для організації послідовного проходження інформації про працівників через відділ кадрів;

– автоматичне перенесення даних, що вводяться в накази, до відповідних масивів особистих карток і призначень / переміщень, виключаючи повторне введення.

5. Відпустки та лікарняні:

- відпустки;
- повернення з відпустки;
- лікарняні та відомості про захворюваність;
- планування та аналіз відпусток.

6. Документи по середньому:

- відрядження і документи, оплачувані за середнім.

До недоліків цієї програми також можна віднести великі для затрати на придбання модулів та їх обслуговування.

«Барекс Зарплата +» програма призначена для обліку і управління кадрами. Підсистема вирішує задачі обліку та управління кадровими ресурсами підприємства. Вона тісно пов'язана з підсистемою нарахування зарплати, має спільні з нею довідники та картотеку, схожий принцип роботи, але в той же час обидві підсистеми можуть експлуатуватися незалежно один від одного.

Підсистема функціонально складається з таких модулів:

- модуль роботи з довідниками;
- модуль роботи зі структурою підприємства;
- модуль роботи з особистими картками;
- модуль роботи з графіком відпусток;
- модуль реєстрації і розрахунку листків непрацездатності;
- модуль роботи з кадровим резервом;
- модуль роботи зі штатним розкладом;
- модуль роботи з відрядженнями;
- модуль роботи з заохоченнями та стягненнями;
- модуль підготовки і формування звітів.

Недоліками такою системи є те, що така програма більше спрямована на розрахунок заробітної плати ніж на облік кадрів на підприємстві. В такій системі відсутні більшість звітів та таблиць щодо руху кадрів на підприємстві та

відомостей про персонал. Також така програма не дозволяє паралельну роботу співробітників відділу кадрів та не забезпечує належний рівень ненадмірності інформації.

На основі проведеного обзору щодо існуючих теоретико-практичних розробок в галузі обліку кадрів можна зробити висновок про недосконалість існуючих програмних рішень, велику кількість зайвих даних, велику ціну, внаслідок чого підтверджується актуальність теми цієї роботи щодо створення багатокористувацького клієнт – серверного додатку на основі бази даних для обліку кадрів в аптеці.

1.3 Формування вимог до системи автоматизації

На основі проведеного аналізу існуючих варіантів автоматизації обліку кадрів на підприємстві та детального огляду недоліків, пов'язаних з їх функціонуванням, проведеного у попередньому підрозділі, можна сформулювати ряд загальних вимог до інформаційної системи.

При розробці інформаційної системи слід враховувати наступні вимоги:

- а) максимальна орієнтація на кінцевого користувача;
- б) простий, зрозумілий інтерфейс тільки з необхідними елементами;
- в) мала ресурсозалежність системи;
- г) незначні витрати коштів на встановлення та супровід автоматизованої системи.

Початкове налагодження системи проводиться розробником.

Розробка, що пропонується, містить автоматизоване розв'язання ряду задач: обробка вхідної інформації щодо руху кадрів на підприємстві, зберігання даних щодо всіх працівників підприємства, зберігання та зручний спосіб відображення інформації щодо працівників.

РОЗДІЛ 2 ПРОЕКТУВАННЯ АВТОМАТИЗОВАНОЇ СИСТЕМИ

2.1 Моделі бізнес-процесів задачі

Взагалі бізнес-процеси роботи аптеки можна описати за допомогою системного аналізу, а саме використавши нотацію побудови IDEF0 діаграму. Це дозволить описати саму суть процесу та логічну послідовність виконання операцій в системі.

На сьогоднішній день існують різні програмні продукти, що дозволяють моделювати бізнес-процеси, що відбуваються на підприємствах. Бізнес-процеси підприємства електронної комерції були змодельовані за допомогою програмного продукту «Allfusion Process Modeler 7.0» (BPwin 7.0). Продукт володіє простим і інтуїтивно зрозумілим інтерфейсом.

IDEF0 – методологія функціонального моделювання і графічного описання процесів. Її ціль – формалізація і опис бізнес-процесів. Особливістю IDEF0 є те, що вона акцентує увагу на ієрархічне представлення об'єктів, що значно полегшує розуміння предметної області. Ідея IDEF0 полягає в тому, що бізнес-процес відображається у вигляді прямокутника, в якій входять і виходять стрілки.

Для IDEF0 має значення сторона процесу та пов'язаною з нею стрілкою:

- ліва сторона – вхід бізнес-процесу – інформація (документ) або ТМЦ, яка буде перетворена в ході виконання процесу;
- права сторона – вихід бізнес-процесу – перетворена інформація (документ) або ТМЦ;
- верхня сторона – управління бізнес-процесу – інформація або документ, який визначає те, як повинен виконуватися бізнес-процес, як має відбуватися перетворення входу у вихід;
- нижня сторона – механізм бізнес-процесу – те, що перетворює вхід в вихід: співробітники або техніка. Вважається, що за один цикл процесу не відбувається зміни механізму[14, 26].

Контекстна діаграма прийому на роботу зображена на рисунку 2.1

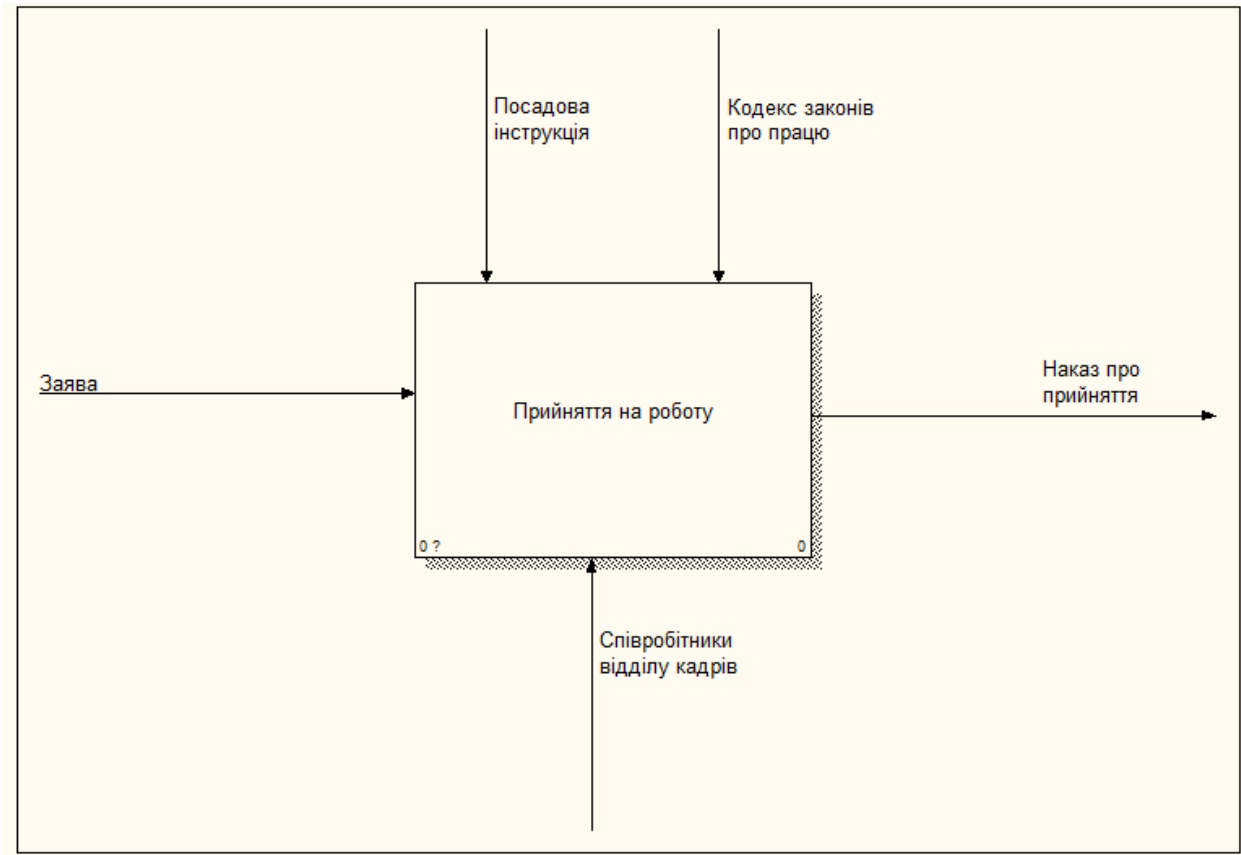


Рисунок 2.1 – Контекстна діаграма «Прийняття на роботу».

Взаємодія системи з навколишнім середовищем описується в термінах входу (на рис 2.3 – «Заявка»), виходу («Наказ про прийняття») управління («Посадова інструкція», «Кодекс законів про працю») і механізмів («Співробітник відділу кадрів»).

Загальні стрілки і діаграми, опишемо за допомогою таблиця 2.1:

Таблиця 2.1 – Опис елементів контекстної діаграми "Прийняття на роботу"

Назва стрілки	Опис	Тип
Заява	Отримання заяви потенційного співробітника	Input
Кодекс законів про працю	Нормативні та законодавчі акти відповідно до яких здійснюється діяльність	Control
Посадові інструкції	Документи, в яких визначено основні завдання, обов'язки, права і відповідальність працівника при виконанні роботи	Control
Наказ про прийняття	Узгоджений документ зарахування співробітника	Output
Співробітник відділу кадрів	Співробітник на підприємстві	Mechanism

На рисунку 2.2 зображена декомпозиція контекстної діаграми «Прийняття на роботу».

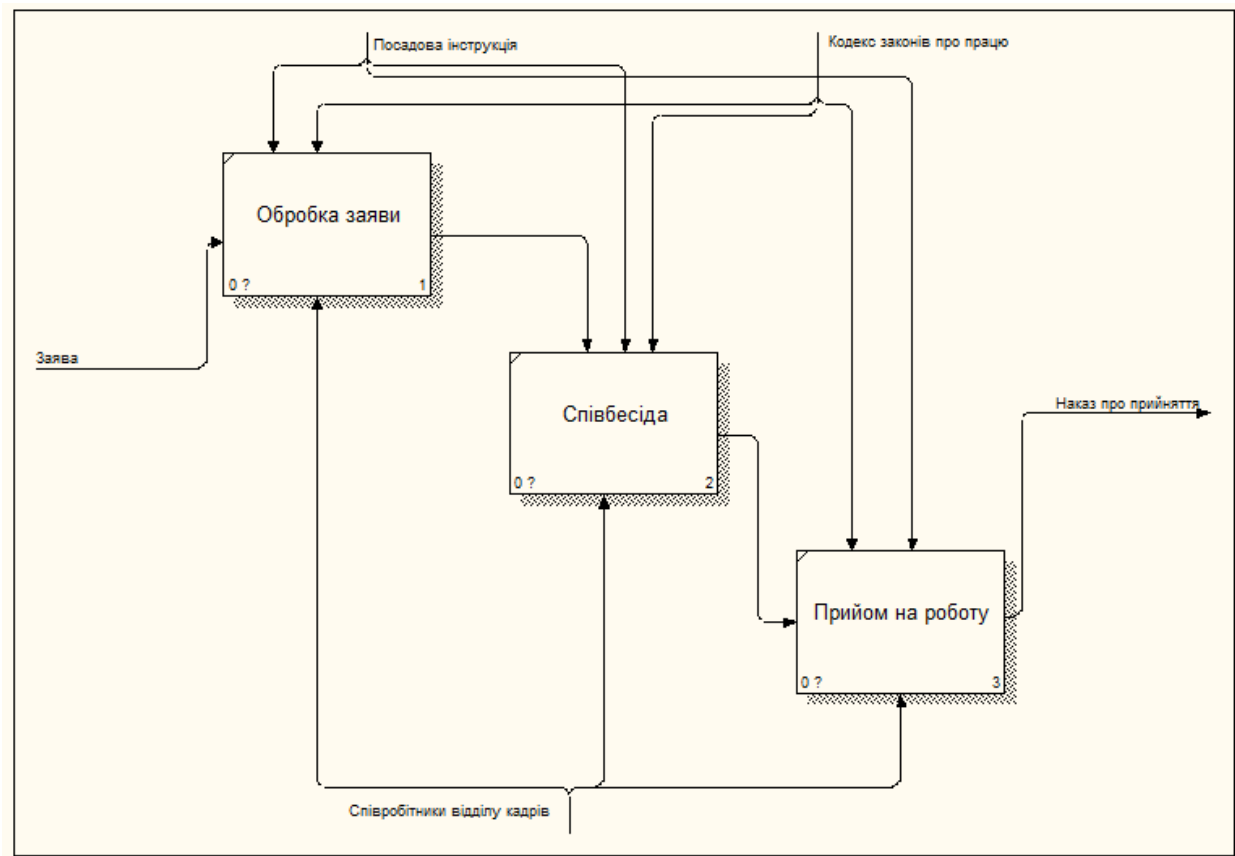


Рисунок 2.2 – Декомпозиція контекстної діаграми

Таблиця 2.2 – Опис елементів декомпозиції контекстної діаграми "Прийняття на роботу"

Функціональний блок	Опис	Статус
Обробка заяви	Процес обробки документів	WORKING
Співбесіда	Процес взаємодії співробітника з робітником відділу кадрів	WORKING
Прийом на роботу	Процес підписання документів	WORKING

Контекстна діаграма звільнення зображена на рисунку 2.3

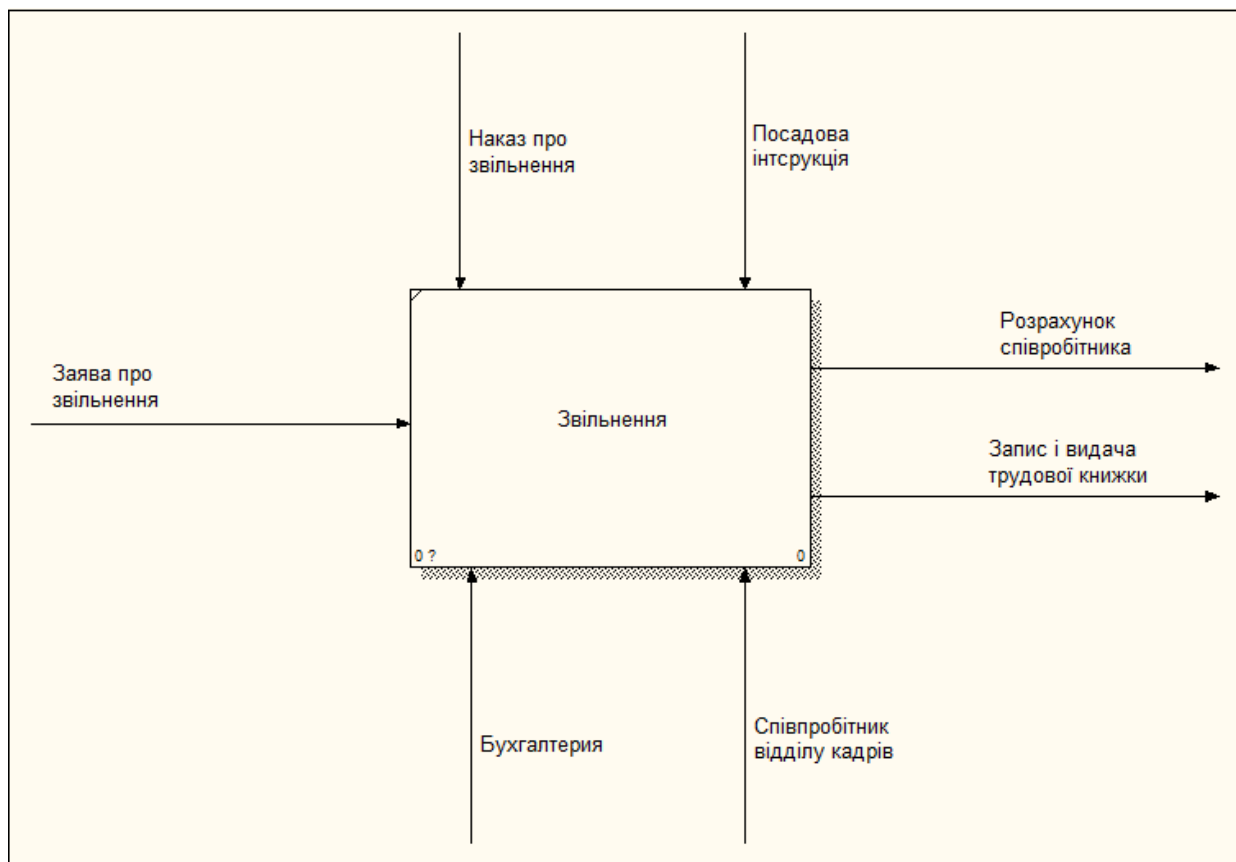


Рисунок 2.3 – Контекстна діаграма «Звільнення».

Взаємодія системи з навколишнім середовищем описується в термінах входу (на рис 2.2 – «Заява про звільнення»), виходу («Розрахунок співробітника», «Запис і видача трудової книжки») управління («Наказ про звільнення», «Посадова інструкція»), і механізмів («Співробітник кадрового відділу», «Бухгалтерія»).

Загальні стрілки і діаграми опишемо за допомогою таблиця 2.3:

Таблиця 2.3 – Опис елементів контекстної діаграми "Звільнення"

Назва стрілки	Опис	Тип
Заява про звільнення	Отримання заяви від співробітника	Input
Наказ про звільнення	Узгоджений та підписаний документ	Control
Посадові інструкції	Документи, в яких визначено основні завдання, обов'язки, права і відповідальність працівника при виконанні роботи	Control
Розрахунок співробітника	Розрахунок з співробітником	Output
Запис і видача трудової книжки	Відмітка про звільнення та повернення книжки	Output
Співробітник відділу кадрів	Співробітник на підприємстві	Mechanism
Бухгалтерія	Бухгалтер проводить розрахунок	Mechanism

На рисунку 2.4 зображена декомпозиція контекстної діаграми «Звільнення».

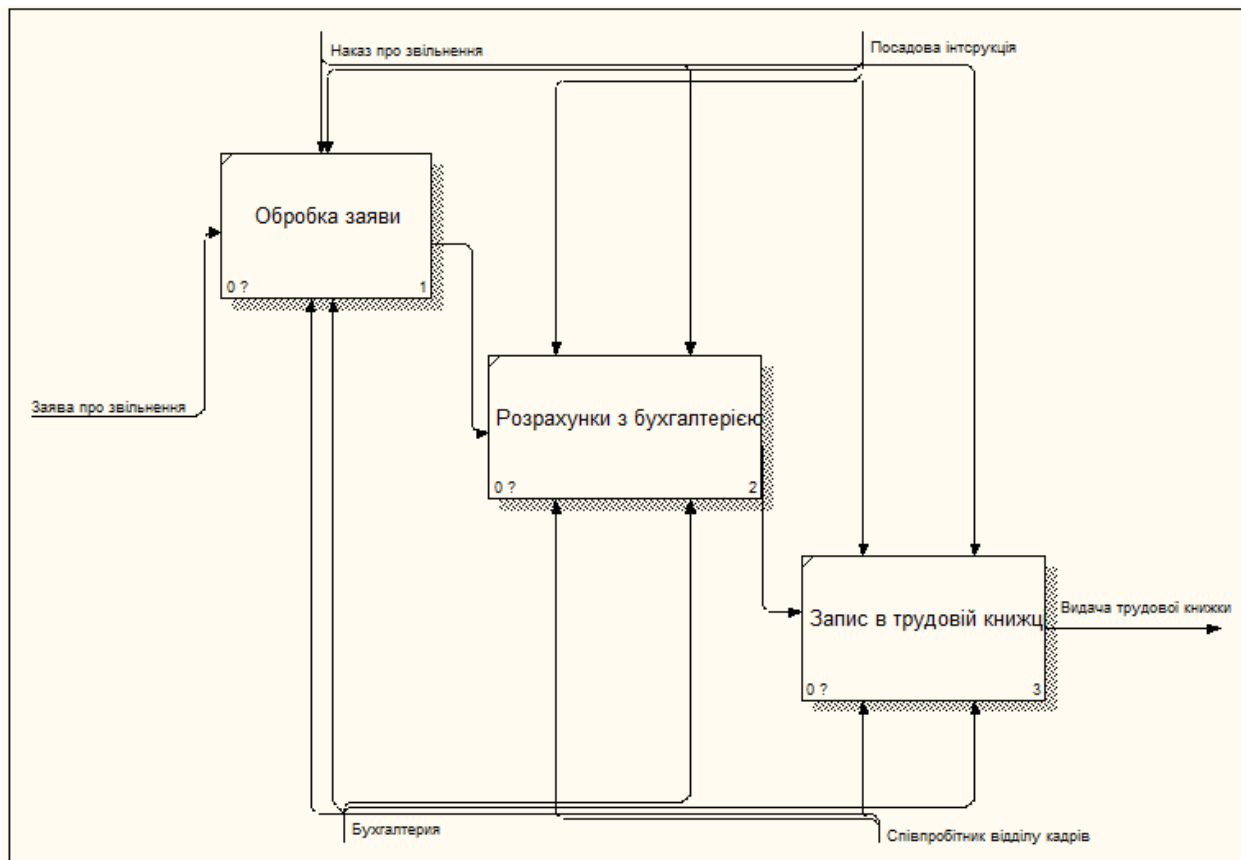


Рисунок 2.4 – Декомпозиція контекстної діаграми

Таблиця 2.4 – Опис елементів контекстної діаграми "Звільнення"

Функціональний блок	Опис	Статус
Обробка заяви	Процес взаємодії обробки заяви	WORKING
Розрахунок з бухгалтерією	Розрахунки з бухгалтерією	WORKING
Запис в трудовій книжці	Процес підписання трудової книжки	WORKING

2.2 Інформаційна архітектура та технології вирішення задачі

Розглянувши в пункті 1.2 автоматизовані програмні рішення можна виділити одну спільну рису, яка притаманна для кожного з них – всі зазначені системи працюють на основі технології «клієнт-сервер».

Перш за все, SQL Server є сервером баз даних [30, 27]. З технічної точки зору термін «клієнт-сервер» пов'язаний з двома взаємодіючими процесами. Клієнтський процес запитує у серверного процесу якусь службу, яка, в свою чергу, обробляє запит клієнта [15]. Клієнтський і серверний процеси можуть бути

запущені на різних комп'ютерах або на одному. У даному питанні важлива сама взаємодія процесів, а не їх фізичне розміщення.

На противагу настільним базам даних, таких як Microsoft Access, яка виконує всю роботу на комп'ютері клієнта, бази даних з архітектурою «клієнт-сервер» подібні бібліотекарю, який приймає запит клієнта, шукає запитану інформацію і повертає фотокопію знайдених матеріалів. Вміщені в бібліотеці реальні матеріали ніколи не виходять з поля зору бібліотекаря. У базах даних з архітектурою «клієнт-сервер» клієнт готує запит на мові SQL (невелике текстове повідомлення) і відсилає його на сервер баз даних, який читає і обробляє його. У сервері підтримується система безпеки, індексуються та зберігаються матеріали, заносяться і обробляються дані, виконуються серверні програми, і виконується доставка результатів запитів клієнта.

Вся робота з базою даних виконується на сервері. Якщо клієнт запитує який-небудь набір даних, він готується на сервері, і його копія доставляється клієнту. Реальні дані та індекси ніколи не покидають межі сервера. Коли клієнт запитує виконання операції вставки, оновлення або видалення, сервер отримує цей запит і сам обробляє його.

Клієнт-серверна модель бази даних володіє рядом переваг в порівнянні з настільною моделлю, а саме:

- підвищена достовірність даних, оскільки вони не розкидані по всій мережі та різних додатків, дані обслуговує тільки один процес;
- обмеження цілісності даних і бізнес-правила можуть підтримуватися на рівні сервера, в результаті чого вони строго дотримуються;
- підвищена безпека даних, оскільки база даних зберігає їх в межах одного сервера. Відкрити файл даних, що захищається сервером, набагато складніше, ніж файл на робочій станції;
- підвищена продуктивність і краще збалансовані робочі станції.

Оскільки велика частина роботи (обробка бази даних) виконується на сервері, а користувача. Оскільки серверний процес забезпечує швидкий доступ користувача до файлів даних, а велика частина даних кешована в пам'яті, операції з базою даних виконуються швидше, ніж в багатокористувацькому настільному

середовищі. Сервер баз даних обслуговує всіх користувачів, що працюють з додатками баз даних, таким чином, набагато простіше оцінити вартість серверу, що встановлюється.

Значною мірою скорочені мережеві потоки. У порівнянні з мережевими потоками, створюваними багатокористувацькими настільними системами, потоки в архітектурі «клієнт-сервер». Заміна перевантаженої настільної системи базою даних «клієнт-сервер» здатна скоротити мережеві потоки більше ніж на 95%.

Зниження мережевих потоків в системах «клієнт-сервер» призводить до того, що додатки добре працюють навіть в розподіленому середовищі і навіть при наявності повільних сполук. Такі маленькі мережеві потоки дозволяють зрівняти в продуктивності локальну мережу зі швидкістю 100 Мбіт/с з модемним підключенням зі швидкістю 56 Кбіт/с для клієнтських додатків, що використовують NET-технології і підключених до бази даних SQL Server.

В клієнт-серверній конфігурації бази даних кожна зі сторін грає конкретну роль. Якщо не впорядкувати ці ролі, то продуктивність і цілісність програми клієнт-серверної бази даних істотно знизяться.

Сервер баз даних відповідає за наступне:

- обробка запитів на вилучення та модифікацію даних;
- інтенсивна обробка даних;
- підтримку всіх правил і обмежень бази даних;
- підтримання безпеки бази даних.

Клієнтський процес відповідає за наступне:

- представлення даних користувачеві в зрозумілому і зручному форматі;
- забезпечення інтерфейсу користувача всілякими інструментами та даними;
- відправку запитів серверу.

Клієнт-серверні реалізації працюють найкраще, коли строго підтримується рівень абстракції даних. Це означає, що будь-який запит на доступ до даних пропускається через збережені процедури рівня абстракції даних. Недбалі клієнт-

серверні моделі, що дозволяють додаткам або звітам безпосередньо звертатися до таблиць бази даних, нежиттєздатні, оскільки база даних не буде розширюваною.

Враховуючи архітектуру роботи перспективних на сьогоднішній день автоматизованих програмних рішень, можливості технології «клієнт-сервер», система, що розробляється також працюватиме за даною технологією. При цьому база даних буде проектуватись та реалізовуватись за допомогою середовища SQL Server 2014 Management Studio [28].

З функціональної точки зору прототип системи, що проектується міститиме три програмні модулі.

1. Модуль «Звіти» – призначений для перегляду інформації про працівників, має такі пункти:

- співробітники;
- зарплата;
- премії;
- штрафи;
- відпустки;
- журнал відвідувань.

2. Модуль «Обробка даних» – призначений для додавання чи видалення даних, має такі пункти:

- співробітники (додавання чи видалення);
- нарахування премії/зарплати/штрафу;
- журнал відвідувань;
- відпустки.

3. Модуль «Пошук» – призначений для пошуку та має такі пункти:

- пошук співробітників(за ім'ям та адресою);
- пошук зарплат;
- пошук штрафів;
- пошук премій;
- журнал відвідувань.

Кожен з модулів зв'язаний зі спільною базою даних, в якій зберігається вся інформація.

2.3 Функціональна структура задачі

Згідно з сформульованими вимогами до автоматизованої системи обліку кадрів можна говорити про її функціональні частини.

На основі літературного огляду існуючих автоматизованих систем обліку кадрів, сформулюємо такі основні її функції:

- ведення БД;
- введення даних про працівників;
- фіксування переміщення кадрів в організації;
- виведення різноманітних даних по запити користувача;
- створення форм статистичної звітності і вихідних форм для внутрішнього користування.

У даній системі користувач сам здійснює введення інформації з первинних документів, які надходять у відділ кадрів на протязі місяця. Введена інформація підлягає візуальному контролю, який являє собою процес перегляду на екрані дисплею набраної інформації і звірення її з первинними документами, а також програмному контролю, який реалізується за допомогою спеціальних програм. Ці програми повинні виконувати наступні перевірки: перевіряють діапазон змін значення атрибуту, тип і наявність його у довіднику. При знаходженні помилки повинно видаватися відповідне повідомлення і користувач повинен здійснювати редагування введеної інформації. Процес вводу повинен супроводжуватися наданням допомоги вибору із довідників відповідних атрибутів, що забезпечить безпомилковий ввід. При необхідності можна знищити непотрібні записи та роздрукувати вміст масивів бази даних.

Результатом виведення інформації за запитом користувача є такі документи: «Співробітники», «Журнал відвідувань», «Відпустки», «Зарплати», «Премії», «Штрафи».

Розглянемо склад програмної підсистеми забезпечення функціональної частини.

Програмне забезпечення – загальне поняття, що вказує на набір кодованих інструкцій для керування процесором комп'ютера. Процесор комп'ютера зчитує

такі кодовані інструкції та виконує їх. Виконання програмного забезпечення комп'ютером полягає у маніпулюванні інформацією та керуванні апаратними компонентами комп'ютера. Наприклад, типовим для персональних комп'ютерів є відображення інформації на екран та прийом її з клавіатури.

Програмне забезпечення задачі обліку руху кадрів складається із загальносистемного, прикладного та інструментального.

До загальносистемного програмного забезпечення відносять: Windows, Linux, Mac OS

До прикладного програмного забезпечення відносять: SQL Management Studio, FoxPro, Oracle.

До інструментального програмного забезпечення відносять: Visual Studio.

Windows є операційною системою, що забезпечує інфраструктуру, на якій можуть працювати прикладні програми, тобто вона керує і контролює комп'ютерним обладнанням для можливості виконання прикладних програм [36].

До основних переваг Windows відносяться:

- будь-яка Windows-програма не залежить від конкретних особливостей зовнішніх пристроїв і може працювати із зовнішнім пристроєм, якщо з ним працює Windows;

- до складу Windows входять всі необхідні функції для побудови користувацького графічного інтерфейсу: вікон, меню, запитів, списків;

- уніфікація інтерфейсу Windows-програм, тобто користувач, який має навички роботи з Windows, може без великих зусиль працювати і з будь-якою Windows-програмою;

- засоби одночасного використання декількох програм.

SQL Server Management Studio – це утиліта, що входить до складу Microsoft SQL Server 2014 і більш пізні версії, для конфігурування, менеджменту та адміністрування всіх компонентів Microsoft SQL Server [30]. Утиліта включає скрипт-редактор і графічну програму, яка працює з об'єктами та налаштуваннями сервера.

Головним інструментом SQL Server Management Studio є Object Explorer, який дозволяє користувачеві переглядати, витягувати, і повністю управляти

об'єктами сервера.

До інструментального програмного забезпечення можна також віднести Microsoft.Net Framework. Ця програмна технологія від компанії Microsoft, призначена для створення звичайних програм та веб-додатків.

Однією з основних ідей Microsoft.NET є сумісність служб, написаних на різних мовах. Наприклад, служба, написана на C++ для Microsoft.NET, може звернутися до методу класу з бібліотеки, написаної на Delphi. На C# можна написати клас, успадкованих від класу, написаного на Visual Basic.NET, а виключення, створене методом, написаним на C#, може бути перехоплено і оброблено в Delphi. Кожна бібліотека (збірка) в.NET має свідчення про свою версію, що дозволяє усунути можливі конфлікти між різними версіями збірок.

Склад апаратної підсистеми забезпечення функціональної частини.

До основних елементів апаратних засобів віднесемо:

- системний блок;
- клавіатура;
- пристрій відображення (монітор);
- CD-ROM;
- друкуючий пристрій;
- різноманітні засоби синхронного зв'язку та управління програмами.

Виходячи з літературного огляду існуючих систем автоматизації обліку кадрів можна виділити наступні вимоги до апаратної підсистеми забезпечення функціональної частини. Мінімальні вимоги для серверної частини: процесор Intel Core i3, 2.5 ГГц, оперативна пам'ять 1 Гб. Об'єм жорсткого диску залежить від об'єму облікової інформації, а значить і від розмірів підприємства, що підлягає автоматизації, але мінімально рекомендований об'єм жорсткого диску складає 250 Гб.

Відповідно для клієнтської частини можна говорити про наступні специфікації персональних комп'ютерів: процесор Intel Core i3, 2.3 ГГц, оперативна пам'ять 1 Гб, жорсткий диск на 250 Гб.

Сервер і клієнтські комп'ютери поєднуються в мережу, типологія та інші особливості яких обираються згідно з особливостями об'єкту автоматизації.

Розглянемо склад інших підсистем забезпечення функціональної частини.

Організаційне забезпечення – сукупність методів і засобів, що використовуються спеціалістами для підвищення ефективності управління як на стадії створення, так і на наступних стадіях життєдіяльності інформаційної системи. Воно базується на методології, що закладена основу функціонування системи, відображає її особливості, містить правові акти, що регулюють діяльність людино-машинної системи.

Інформаційне забезпечення – сукупність форм документів, класифікаторів, нормативної бази та реалізованих рішень по об'ємам, розміщенню і формам існування інформації, які застосовуються в інформаційній системі при її функціонуванні.

Інформаційне забезпечення складається з інструктивних і методичних матеріалів, системи класифікації і кодування, позамашинної і внутрішньомашинної інформаційної бази. Інформаційне забезпечення розроблюється з врахуванням вимог принципів системності ІС [9].

Основні принципи створення інформаційного забезпечення:

- цілісність інформації;
- достовірність інформації;
- контроль інформації;
- єдність та гнучкість;
- стандартизація та уніфікація;
- адаптивність;
- мінімізація вводу та виводу інформації.

Специфічною особливістю обліку кадрів порівняно з іншими функціональними видами обліку є те, що робота з людьми, пов'язана з працевлаштуванням, переведенням чи звільненням з роботи, організацією оплати праці, підготовкою та підвищенням кваліфікації, регулюванням робочого часу, створенням сприятливих умов праці, оцінюванням персоналу тощо, потребує чіткої регламентації прав і обов'язків усіх учасників трудових відносин. Це досягається прийняттям відповідних законодавчих актів державними органами влади та розробленням нормативно-правових і науково-методичних документів

(правил, положень, норм, інструкцій, рекомендацій, характеристик тощо) [17].

У зв'язку з тим, що кількість працівників стає все більше, виникає проблема структурованого зберігання та обробки даних. На вирішення саме цього питання й спрямована розробка реляційної бази даних.

Відмова від традиційного підходу, а саме збереження інформації в окремих файлах та їх безпосередньої обробки, пояснюється тим, що такий варіант управління даними призводить до надлишковості – ситуації, коли одні й ті самі дані зберігаються в різних файлах. До того ж окремі додатки можуть користуватися подібною інформацією, але необхідним буде її збереження в файлах різних форматів. Надлишковість даних є негативним явищем з причини того, що вона викликає:

- неоднозначність інформації – виникає ситуація, коли один і той самий елемент у різних файлах називається по-різному;
- неузгодженість, яка полягає в тому, що кожен додаток трактує окремий елемент по-своєму;
- марний труд – створення записів з даними для підтримки окремого програмного додатку у той час, коли більша частина цих даних уже існує, є марним витрачанням часу, сил і грошей.

Перелічені проблеми можна вирішити, об'єднавши всі дані в єдину систему – реляційну базу даних.

Вибір із чотирьох основних типів баз даних (ієрархічної, мережевої, реляційної та об'єктно-орієнтованої) саме реляційної структури не є випадковим. Ієрархічна модель бази даних найчастіше застосовується й підходить для тих систем, які ґрунтуються на ієрархічній моделі, що не є характерним для предмету дослідження. До того ж частіше за все вказаний тип організації даних не виключає можливість повторної появи даних, тобто надлишковість залишається високою.

Використовуючи мережевий тип бази даних, можна шляхом глибокого аналізу запобігти надлишковості, і файлові системи стануть інтегрованими, але це достатньо складно.

Об'єктно – орієнтовані системи управління базами даних (СУБД) не набули

достатнього розповсюдження, оскільки визначення цієї моделі організації даних є недостатньо чітким, що призвело до різних способів реалізації об'єктно-орієнтованих СУБД.

Щодо реляційного типу баз даних, то на концептуальному рівні дані представляються у вигляді таблиць. Рядки ідентичні записам спеціалізованих файлів, а стовпчики відображають зміст запису. Зв'язок між таблицями

здійснюється на основі значень стовпчиків. Реляційна структура забезпечує більш просту систему розробки, яку легше створити й прочитати. Ще одним доводом використання саме реляційного типу бази даних для проектування бази даних обліку кадрів є те, що програми для маніпулювання реляційними структурами пишуться достатньо легко, до того ж більшість виробників систем управління базами даних використовують саме реляційну модель.

Однією з основних задач, що стоять перед системами управління базами даних, є необхідність надання користувачам зручних методів доступу до даних. Ця задача може бути розв'язана шляхом використання мов, спеціально орієнтованих на роботу з даними. Однією з таких мов є структурована мова запитів SQL (Structured Query Language). Мова SQL являє собою набір команд, за допомогою яких користувачі й програмні додатки можуть отримувати доступ до даних, що зберігаються в системі управління базою даних (СУБД), і виконувати з ними різні операції [25].

Мова SQL містить команди, які дозволяють розв'язати наступні задачі:

- вибірка даних;
- вставка, видалення і зміна рядків таблиці;
- створення, заміна, видалення об'єктів;
- контроль доступу до бази даних і її об'єктів;
- гарантування цілісності і узгодженості бази даних.

На сьогоднішній день мова SQL є загальноприйнятим стандартом при роботі з реляційними системами управління базами даних. Мова SQL була офіційно затверджена в якості промислового стандарту організацій зі стандартизації ANSI (American National Standard Institute) та ISO (International Standards Organizations).

При виборі СУБД дуже важливо обрати ту базу даних, яка в найбільшій мірі відповідає вимогам до інформаційної системи. У першу чергу при виборі СУБД необхідно враховувати наступні фактори:

- максимальна кількість користувачів, які одночасно звертаються до бази;
- характеристики клієнтського персонального комп'ютера;
- апаратні компоненти сервера;
- серверна операційна система;
- рівень кваліфікації персоналу.

На сьогоднішній день відомою є велика кількість різноманітних серверів баз даних SQL. Одними з найбільш успішних серверних СУБД є Oracle, IBM DB2, Microsoft SQL Server.

Найважливішими характеристиками СУБД Microsoft SQL Server є наступні:

- простота адміністрування;
- можливість підключення до Web;
- швидкодія й функціональні можливості механізму сервера СУБД;
- наявність засобів віддаленого доступу.

Дана система має чудові засоби тиражування, які дозволяють синхронізувати дані персонального комп'ютера з інформацією бази даних і навпаки. Зрештою дана СУБД являє собою сучасну повно функціональну базу даних, яка ідеально підходить для малих та середніх організацій.

Отже, можна зазначити, на сьогоднішній день для розробки бази даних обліку кадрів найкращим є варіант вибору Microsoft SQL Server, оскільки саме ця система призначена для автоматизації роботи на малих і середніх підприємствах, характеризується невисокою складністю й реалізується на платформі Windows.

Необхідно чітковизначити коло задач, для розв'язання яких створюється база даних. Цього можна досягти з'ясувавши відомості, необхідні для розробки ефективної структури даних.

Зазвичай цей процес ділять на три етапи:

- а) визначення цілей створення системи;
- б) визначення обсягів і типів даних;
- в) визначення способів використання даних.

Щодо першого етапу, а саме визначення цілей системи, то слід зауважити, що в перспективі на підприємстві буде розширятися штат, що призведе до різкого збільшення обсягів інформації, яка буде оброблятися. Таким чином, з'являється нагальна потреба у створенні бази даних для зберігання і оброблення інформації щодо обліку кадрів, що дозволить знайти ефективне рішення проблеми, що склалася.

База даних відділу кадрів має бути реляційною базою даних, що накопичує інформацію про співробітників підприємства та їх рух в межах та за межами підприємства. Отже, метою даного проекту є створення реляційної бази даних, яка б забезпечила спрощення операцій збору, обробки та аналізу інформації відділу кадрів для підвищення ефективності його діяльності.

Другим етапом створення ефективної структури бази даних є визначення обсягів і типів даних, від яких безпосередньо залежить продуктивність бази даних, тобто об'єм інформації впливає на розмір бази даних, а типи даних виступають у ролі факторів, що визначають обмеження, які накладаються на структуру даних.

При розробці даного проекту враховані такі елементи збору інформації, як:

- прізвище, ім'я та паспортні дані особистості;
- адреса проживання працівників;
- посада працівника;
- особисті дані співробітників;
- перелік посад.

Щодо тенденції зростання обсягів інформації, то дані, що стосуються працівників зростатимуть невисокими темпами. Що ж до даних, що містять інформацію про потенційних співробітників, вакансії підприємства та оброблені працівниками заявки, то кількість записів у цих таблицях зростатиме щоденно високими темпами, що призводитиме до значного збільшення об'єму бази даних.

Наступним етапом створення ефективної структури бази даних є визначення способів використання бази даних, тобто виявлення тих, хто буде звертатися до даних і характер задач, які користувач виконуватиме з ними.

Отже, остаточною метою створення даного проекту є створення

масштабованої, доступної й надійної бази даних, що спрощує процеси введення, обробки й аналізу інформації, необхідної для щоденного ефективного функціонування відділу кадрів.

2.4 Підсистеми забезпечення функціональної частини

Забезпечення кадрового обліку складається з підсистем технічного, математичного, програмного, інформаційного, лінгвістичного, правового, організаційно-методичного та ергономічного забезпечення.

Розглянемо більш детально кожен з вищезазначених підсистем:

1. Підсистема технічного забезпечення включає наступні елементи:

- комплекс технічних засобів, що використовується для вводу, обробки, підготовки, виводу, збереження, використання та передачі даних;
- комплект технічної документації;
- персонал, що обслуговує технічні засоби.

2. Підсистема математичного забезпечення – сукупність математичних методів, моделей і алгоритмів, які застосовуються для вирішення проблем математичного моделювання процесів, прийняття рішень, прогнозування тощо.

3. Підсистема програмного забезпечення – сукупність програм постійного користування, що створюють бібліотеки програм, а саме системні програми, пакети прикладних програм, системи управління базами даних (СУБД).

4. Підсистема інформаційного забезпечення – сукупність даних та правил їх отримання, організації структури та змісту інформаційних сукупностей, зберігання та оновлення, що загалом створюють базу даних. Інформаційне забезпечення містить у собі не лише засоби і методи ведення усієї інформаційної бази системи, а й інформаційні ресурси як предмет праці та інформацію як продукт праці. Отже, до інформаційного забезпечення належать також методи класифікації і кодування інформації, способи організації нормативно-довідкової інформації.

5. Підсистема лінгвістичного забезпечення - сукупність мовних засобів для формалізації мови опису інформації та інших елементів інформаційної системи.

6. Підсистема правового забезпечення - сукупність законодавчих актів, правових норм і нормативів, що пов'язані з функціонуванням інформаційної системи.

7. Підсистема організаційно-методичного забезпечення – сукупність правил, документів, інструкцій та положень, які забезпечують створення системи та взаємодію її складових частин у процесі функціонування. Таким чином до організаційного забезпечення належить сукупність документів, що описують технологію функціонування системи, методи вибору і застосування користувачами технологічних прийомів для одержання конкретних результатів при функціонуванні.

8. Підсистема ергономічного забезпечення – сукупність вимог, що спрямовані на узгодження психологічних та фізіологічних особливостей людини з технічними характеристиками засобів обчислювальної техніки та параметрами робочого місця, середовища тощо. Загалом робоче місце має відповідати нормативним значенням рівня освітленості, шуму, випромінення, температури, вологості

До програмного забезпечення автоматизованої системи управління, що розробляється можна вказати такі вимоги:

- платформа Windows 7 і вище;
- наявність встановленого на клієнтському комп'ютері .NET Framework версії 3.5 і вище;
- наявність встановленого SQL Server 2014 і вище;
- наявність встановленого Microsoft Visual C++.

Апаратне забезпечення – набір апаратних засобів, що необхідні для нормального функціонування системи. Враховуючи це при розробці системи проводять ряд досліджень, під час яких визначають потреби у потужностях пристроїв.

Враховуючи те, що система, яка проектується, працює як на сервері, так і на клієнтському комп'ютері, вимоги до апаратного забезпечення розділимо на дві частини: вимоги до апаратного забезпечення серверу та вимоги до апаратного забезпечення клієнтського комп'ютера.

Апаратне забезпечення серверу повинно відповідати наступним вимогам (табл. 2.5).

Таблиця 2.5 – Вимоги до апаратного забезпечення серверу

Компонент	Характеристика
Операційна система	Windows 7/ Windows 8.1
Процесор	Не нижче: Intel Core i3
Швидкодія процесора	Не менше: 2 Гц
Оперативна пам'ять	Не нижче: 1 ГБ
Фізична пам'ять	Не менше 2 ГБ вільного місця

Апаратне забезпечення клієнтського комп'ютера повинно відповідати наступним вимогам (табл. 2.6):

Таблиця 2.6 - Вимоги до апаратного забезпечення клієнтського комп'ютера.

Компонент	Характеристика
Операційна система	Windows 7/Windows 8-8.1
Процесор	Не нижче: Intel Core i3
Швидкодія процесора	Не менше: 2 Гц
Оперативна пам'ять	Не нижче: 1 ГБ
Фізична пам'ять	Не менше 500 МБ вільного місця

Лінгвістичне (мовне) забезпечення – це засоби інтерфейсу між користувачем та програмними засобами інформаційної системи. Лінгвістичне забезпечення поділяють на: командний інтерфейс – коли в якості взаємодії між користувачем і системою використовуються команди та відповіді; графічний інтерфейс користувача – коли засобом взаємодії є візуальні об'єкти (меню, панелі інструментів, діалогові вікна, тощо)

Лінгвістичне забезпечення включає в себе:

- інформаційні мови для опису структурних одиниць баз даних інформаційної системи;
- мови маніпулювання і обміну даними в базі даних системи;
- мовні засоби інформаційно-пошукових систем;

- мовні засоби системи автоматизованого проектування;
- діалогові мови.

Відповідно, в якості лінгвістичного забезпечення автоматизованого програмного рішення, що проектується, можна представити:

- мова Transact-SQL на базі реляційної СУБД Microsoft SQL Server 2014;
- мова C# середовища розробки Visual Studio 2017 як мова організації внутрішньої взаємодії між елементами управління;
- мова MSIL (Microsoft Intermediate Language) каркасу .NET Framework, що є мовою-посередником між середовищем розробки Visual Studio та платформи Framework.

Технічне забезпечення – це комплекс технічних засобів (КТС), що використовуються в інформаційній системі. До КТС належить: комп'ютерна техніка; периферійне обладнання; засоби оргтехніки; технічні лінії зв'язку

Робота інформаційної системи реалізується за допомогою використання технічних засобів, які забезпечують збирання, реєстрацію, передачу, зберігання, обробку інформації, а також процеси взаємодії людини і комп'ютера для вирішення задач управління об'єктом.

Відповідно, в якості технічного забезпечення автоматизованого програмного рішення, що проектується, можна представити:

- настільний або переносний комп'ютер з відповідними вимогами;
- клавіатура, мишка.

Інформаційне забезпечення містить сукупність інформації, що зберігається, використовується та обробляється в інформаційній системі. До складу інформаційного забезпечення належить поза машинна та машинна інформаційні бази. Позамашинна інформаційна база складає: систему первісних, довідкових, нормативних та вихідних документів; систему класифікації та кодування; технічну документацію. Машинна інформаційна база складає: комп'ютерну базу даних, електронні документи, екранну допомогу.

Інформаційне забезпечення в досить впливово визначає інтелектуальність системи, так як містить всю інформацію, що використовується, володіє нею і здійснює обмін інформацією всередині і зовні інформаційної системи. До

інформаційного забезпечення в системах управління висувуються значні вимоги. Інформація повинна бути своєчасною, постійно оновлюваною, достовірною та доступною, представляти в зручній для користувача формі, повною. Неповність інформації досить часто є причиною прийняття несвоєчасних і нерациональних управлінських рішень. Основний ефект інформаційної системи в напрямку удосконалення управління досягається за рахунок автоматизації інформаційних процесів і покращення управлінських рішень.

Інформаційне забезпечення містить в собі не лише дані, але і систему доступу до них, яка реалізується програмними засобами. В цьому проявляється тісний взаємозв'язок інформаційного і програмного забезпечення, яке надає ці засоби.

РОЗДІЛ 3 РЕАЛІЗАЦІЯ ПРОТОТИПУ АВТОМАТИЗОВАНОЇ СИСТЕМИ

3.1 Структура та особливості реалізації інформаційного забезпечення

Згідно зі здійсненим аналізом інформаційних процесів на об'єкті автоматизації проведемо семантичне моделювання, яке повинно відобразити основні зв'язки між об'єктами системи, що підлягає автоматизації.

Семантичне моделювання займається побудовою моделі, яка визначає змістовне значення даних, тобто семантика бази даних займається зв'язками між множиною даних і елементами реального світу, які вони представляють.

Отже, після виявлення функціональних вимог до бази даних обліку кадрів слід розглянути питання відповідності об'єктів реального світу їх віртуальним аналогам, виявити зв'язки, які існують між цими віртуальними об'єктами (сутностями), а також визначити ті якості, за допомогою яких буде охарактеризований кожен об'єкт. Сутність – це деяка модель об'єкту реального світу, наприклад, людина. Кожній сутності у відповідність ставляться властивості або атрибути, які можуть включати в себе, в прикладі з людиною ім'я, статус, адресу проживання, посаду тощо.

Виявивши об'єкти, які представляють найбільший інтерес для даної бази даних, можна переходити до побудови моделі «сутність–зв'язок». Діаграма «сутність–зв'язок» є однією з ключових концепцій, які використовуються в процесі розробки бази даних. За допомогою цієї діаграми структура даних може бути чітко спроектована перед тим, як модель буде реалізована у вигляді бази даних. Діаграма «сутність–зв'язок» (див. рисунок 3.1) включає в себе логічні (віртуальні) об'єкти, які були використані для представлення в базі даних об'єктів реального світу, і взаємозв'язки між цими об'єктами. У випадку проектування бази даних для обліку кадрів зазначена діаграма визначає взаємовідносини між працівниками.

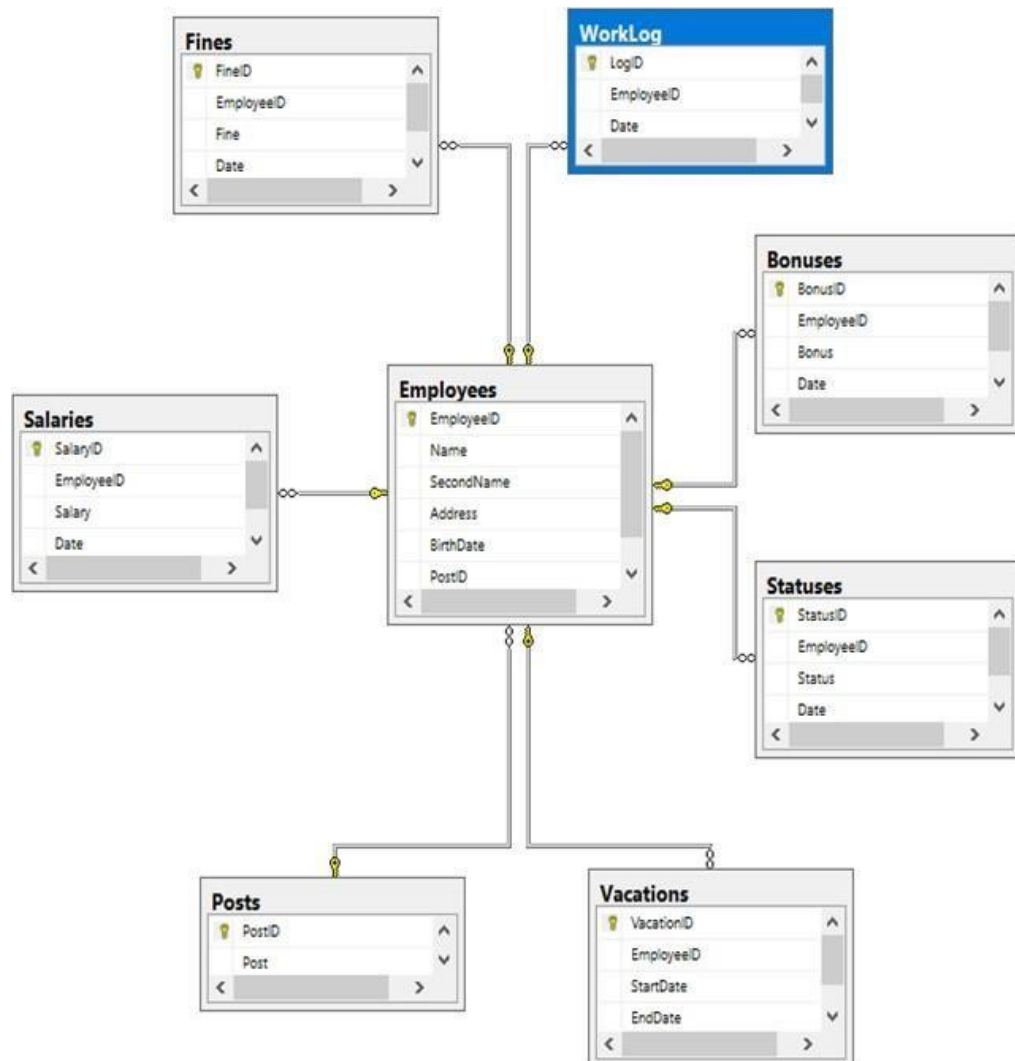


Рисунок 3.1 – Діаграма «Сутність-зв'язок»

Таким чином, для проектування бази даних відділу кадрів виділяються наступні сутності:

- «Співробітник»(Employee);
- «Посада»(Post);
- «Журнал відвідувань»(WorkLog);
- «Відпустки»(Vacations);
- «Зарплати»(Salaries);
- «Премії»(Bonuses);
- «Штрафи»(Fines);
- «Статус»(Status).

Сутності, окрім атрибутів, характеризуються ще й зв'язками між собою.

Відношення між сутностями – це віртуальний зв'язок, як правило, заснований на

взаємовідносинах об'єктів реального світу. Наприклад, оскільки кожен працівник і здобувач є людьми, то між ними можна створити деякий віртуальний зв'язок. Відношення приймають одну із трьох форм, які називають кардинальністю, а саме: «один до одного», «один до багатьох», «багато до багатьох».

Після створення діаграми «сутність–зв'язок» перетворимо її у множину таблиць реляційної бази даних. Приведемо структури утворених таблиць (таблиці 3.1 – 3.8)

Таблиця 3.1 – Структура таблиці Співробітник (Employee)

№ п/п	Атрибут	Тип даних	Опис
1	EmployeeID	Integer	Унікальний індекс робітника
2	SecondName	VARCHAR(30)	Прізвище
3	Name	VARCHAR(30)	Ім'я
4	BirthDay	DATE	Дата народження

Таблиця 3.2 – Структура таблиці Посада (Post)

№ п/п	Атрибут	Тип даних	Опис
1	PostID	Integer	Унікальний індекс посади
2	Post	VARCHAR(30)	Назва посади

Таблиця 3.3 – Структура таблиці Відпустки (Vacations)

№ п/п	Атрибут	Тип даних	Опис
1	VacationID	Integer	Унікальний індекс відпустки
2	EmployeeID	VARCHAR(25)	Посилання на співробітника, якого стосується цей запис
3	StartDate	DATE	Дата початку відпустки
4	EndDate	DATE	Дата кінця відпустки

Таблиця 3.4 – Структура таблиці Розклад

№ п/п	Атрибут	Тип даних	Опис
1	StatusID	Integer	Унікальний індекс статусу
2	EmployeeID	Integer	Посилання на співробітника, якого стосується цей запис.
3	Date	DATE	Дата зміни статусу
4	Status	Bit	Логічне поле(1 – працює, 0- ні)

Таблиця 3.5 – Структура таблиці Зарплати (Salaries)

№ п/п	Атрибут	Тип даних	Опис
1	SalaryID	Integer	Унікальний індекс зарплати
2	EmployeeID	Integer	Посилання на співробітника, якого стосується цей запис.
3	Date	DATE	Дата виплати
4	Salary	MONEY	Розмір виплати в грошових одиницях

Таблиця 3.6 – Структура таблиці Штрафи (Fines)

№ п/п	Атрибут	Тип даних	Опис
1	FineID	Integer	Унікальний індекс штрафу
2	EmployeeID	Integer	Посилання на співробітника, якого стосується цей запис.
3	Date	DATE	Дата штрафу
4	Fine	MONEY	Розмір штрафу в грошових одиницях

Таблиця 3.7 – Структура таблиці Премії (Bonuses)

№ п/п	Атрибут	Тип даних	Опис
1	BonusID	Integer	Унікальний індекс премії
2	EmployeeID	Integer	Посилання на співробітника, якого стосується цей запис.

3	Date	DATE	Дата виплати
4	Bonus	MONEY	Розмір шпремії в грошових одиницях

Таблиця 3.8 – Структура таблиці Журнал відвідування (WorkLog)

№ п/п	Атрибут	Тип даних	Опис
1	LogID	Integer	Унікальний індекс запису в журналі
2	EmployeeID	Integer	Посилання на співробітника, якого стосується цей запис.
3	Date	DATE	Дата запису

Для перетворення діаграми «сутність–зв’язок» в реляційну базу даних, необхідно виконати такі дії:

- для кожної сильної сутності створити базову таблицю.
- ключовий атрибут сутності стає її первісним ключем;
- для кожної слабкої сутності створити таблицю, що складається з усіх простих атрибутів даної сутності і додаткових стовпців первісних ключів сутностей, від яких залежить її існування.

Якщо дві сутності беруть участь у зв’язку — один до багатьох (1- M), таблиця, що представляє сутність з кардинальністю M, повинна мати стовпець зовнішнього ключа, що представляє цей зв’язок;

- якщо дві сутності беруть участь у зв’язку — один до одного (1-1), в таблицю, що представляє одну з цих сутностей, необхідно включити стовпець зовнішнього ключа;

– якщо дві сутності беруть участь у зв’язку — багато до багатьох (M- M), необхідно створити таблицю зовнішніх ключів двох таблиць, що представляють сутності учасники;

– якщо сутність має багатозначний атрибут, то необхідно створити окрему таблицю. Один стовпець даної таблиці буде зовнішнім ключем до таблиці, що представляє сутність, а інший буде представляти багатозначний атрибут. Первісний ключ у даному випадку – комбінація цих стовпців;

- якщо у зв’язку беруть участь більше двох сутностей, необхідно створити

таблицю, що складається із зовнішніх стовпців всіх таблиць, що представляють сутності учасники.

Нормалізація таблиць – це формальний апарат обмежень на формування таблиць, що описує поділ таблиць на дві чи більше частин і забезпечує застосування кращих методів додавання, зміни і видалення даних; процес представлення даних в вигляді простих двомірних таблиць, який дозволяє вилучити дублювання цих даних і забезпечити узгодженість збережених в базі даних. Ціллю нормалізації являється отримання такого проекту бази даних, в якому будь-яка частина інформації зберігається лише в одному місці.

Таблиця вважається нормалізованою на певному рівні, якщо вона задовольняє вимоги, відповідні формі нормалізації.

Кожна наступна форма має ознаки попередньої, доповнені новою власною ознакою.

Так, перша нормальна форма вимагає, щоб домени усіх атрибутів містили прості, неподільні значення, а значення атрибутів у кортежі повинні бути одними із значень його домену.

Для визначення другої нормальної форми необхідно розглянути деякі поняття. Нехай у таблиці існує деяка множина атрибутів X . Деякий атрибут (або множина атрибутів) A вважається функціонально залежним від X тоді і тільки тоді, коли кожній комбінації значень X відповідає тільки одне значення A . Множина атрибутів складає суперключ, коли кожна комбінація його значень зустрічається тільки один раз [28]. Суперключ, що не має зайвості, називається потенційним, причому зайвість – це спроможність ключа містити атрибути, видалення яких не призводить до втрати його унікальної ідентифікації кортежу. Усі атрибути потенційного ключа називаються ключовими, а всі інші – не ключовими. Один з потенційних ключів необхідно зробити первинним. Атрибут A знаходиться у повній функціональній залежності від атрибутів X і Y , якщо він функціонально залежить від цих атрибутів і не існує функціональної залежності цього атрибута від жодної підмножини атрибутів. Інакше мова йде про часткову функціональну залежність.

Отже, база даних знаходиться у другій нормальній формі, якщо вона

знаходиться у першій нормальній формі і якщо вона не містить неключових атрибутів, що знаходяться в частковій функціональній залежності від первинного ключа.

Щоб дати визначення третій нормальній формі, необхідно навести поняття транзитивної залежності. Функціональна залежність між неключовими атрибутами називається транзитивною залежністю.

Таким чином, база даних знаходиться у третій нормальній формі, якщо вона знаходиться у другій нормальній формі і в ній відсутні транзитивні залежності.

Реалізація такої структури даних та інформаційних потоків у реляційну базу даних як було сказано в попередніх розділах була проведена за допомогою мови програмування Transact SQL у SQL Server Management Studio. Програмний код створення бази даних та базових таблиць наведено у додатку Б.

Для реалізації об'єктів вихідної інформації було використано представлення. Представлення дозволяє визначити спосіб подачі інформації, що зберігається в базі даних кінцевому користувачеві. Використовуючи представлення, можна сховати від користувача код Transact-SQL, необхідний для витягу з бази даних складних структур інформації.

Завдяки схожому з таблицею поводженню, представлення дозволяє виконати стосовно нього характерні для таблиць функції. Це має на увазі можливість використання таких операторів керування даними, як INSERT, UPDATE і DELETE, хоча тут існують деякі специфічні обмеження.

Представлення мають кілька істотних переваг:

- можливість приречення структури даних для перегляду кінцевим користувачем;
- можливість обмеження прав на перегляд представлення; можливість використання представлення для створення звітності.

3.2 Структура та особливості реалізації алгоритмічного забезпечення

Реалізація алгоритмічного забезпечення була основана на взаємодії клієнтського додатку з базою даних обліку кадрів. Оскільки головними

елементами обліку кадрів є накази про зарахування на роботу, звільнення, фізичний перевід співробітника, додавання здобувачів та їх видалення, то важливо маніпулювати та обробляти вхідні дані. Така концепція реалізовувалась на основі створення збережених процедур та тригерів в базі даних обліку кадрів. Збережена процедура – це об'єкт MS SQL Server 2014, представлений набором відкомпільованих операторів мови запитів Transact-SQL. Подібно представленню, збережена процедура не містить ніякої інформації з бази даних, замість цього вона містить посилання на базові таблиці, у яких зберігаються всі потрібні дані.

Збережена процедура, подібно представленню, дозволяє проводити оновлення інформації одночасно тільки в одній базовій таблиці, але оскільки збережена процедура – це об'єкт, що містить набір операторів Transact-SQL, можна оновлювати кожну таблицю за допомогою усього лише однієї збереженої процедури.

Поряд із можливістю передачі в збережену процедуру значень змінних, існує можливість витягу значень зі збереженої процедури.

Збережені процедури можуть містити керуючі оператори Transact-SQL, включаючи оголошення і присвоєння значень змінним, оператори IF, CASE та GOTO.

Оскільки збережена процедура являє собою набір операторів Transact-SQL, в одній процедурі можуть одночасно бути присутні такі оператори керування даними, як SELECT, INSERT, UPDATE і DELETE.

Оскільки всі оператори збереженої процедури попередньо компілюються ще до її виконання, SQL Server 2014 має засоби, що дозволяють підвищити швидкість і ефективність процесу виконання збереженої процедури.

Слід пам'ятати, що процедури, що зберігаються, так само як і представлення, таблиці і індекси, є об'єктами бази даних. Таким чином, при використанні процедури, що зберігається, спочатку необхідно створити відповідний об'єкт, а потім виконати його (використовуючи операторів управління даними Transact-SQL), вказавши при цьому всі необхідні значення параметрів.

Створення збережених процедур до бази даних наведено у додатку В.

При всій своїй ефективності і універсальності використання збережена процедура, позбавлена однієї дуже важливої властивості:

Здатності реагувати на події. Це зокрема, позначає неможливість автоматичного виконання процедури, що зберігається, при внесенні до таблиці нового рядка інформації. Здатністю реагувати на події володіють спеціальні об'єкти бази даних MS SQL Server 2014, так звані – тригери.

Тригер можна розглядати як особливий тип збереженої процедури, який здатний реагувати на події. Так, наприклад, при внесенні нової інформації у таблицю бази даних, можна за допомогою тригера відразу перевірити її на правильність і крім того виконати ще якісь додаткові дії. Те саме можна робити і при видаленні інформації з таблиць.

Основна перевага тригера над збереженою процедурою – здатність тригера реагувати на події.

Приклад створення тригера у проекті наведено у додатку Г.

Загальний алгоритм діяльності автоматизованої системи обліку кадрів на підприємстві наведено на рисунку 3.1.

Такий алгоритм реалізований на основі створення клієнтського додатку, який є посередником між користувачем автоматизованої системи та базою даних з її об'єктами.

Згідно з прийнятими рішеннями у попередніх розділах, створення клієнтського додатку проводилося за допомогою Microsoft Visual Studio 2017 на мові програмування C# [1]. Це об'єктно-орієнтована мова високого рівня, що забезпечує гнучкі методи маніпулювання даними та створення високо функціональних Windows-додатків. Реалізація алгоритмічного забезпечення в Microsoft Visual Studio 2017 здійснювалася за допомогою численних елементів управління, що представляються цією платформою, серед яких можна назвати: DataGridView, button, textbox, combobox та інші [10, 35].

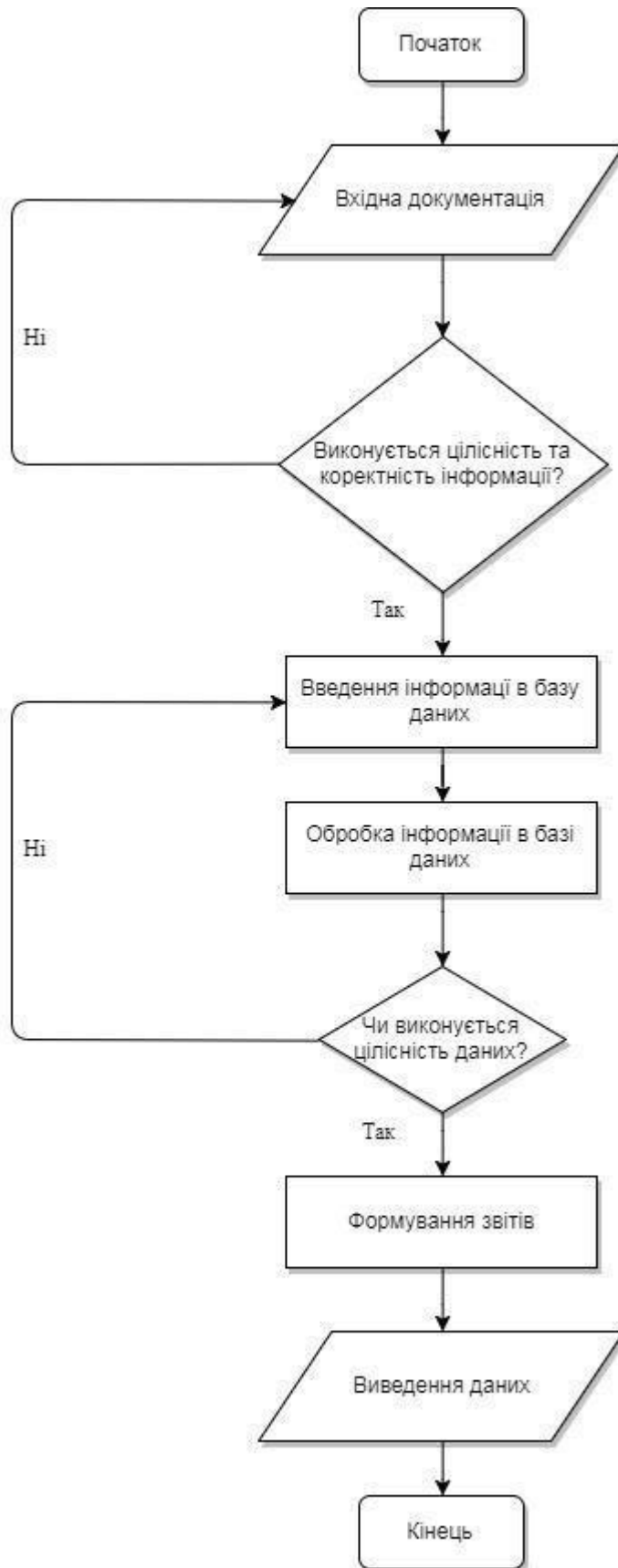


Рисунок 3.2 – Загальний алгоритм роботи автоматизованої системи

3.3 Контрольний приклад та інструкція з використання

Концептуальною засадою розробки інтерфейсу користувача є визначення того, що клієнтський додаток не повинен бути ресурсоємким, складним у використанні.

Для розробки інтерфейсу користувача одним з найбільш поширених підходів, які використовують при розробці клієнтських програм для взаємодії з реляційними базами даних під ОС Windows, є використання інструментарію Visual Studio корпорації Microsoft. Для рішення цієї задачі створимо багатодокументний інтерфейс користувача – MDI (Multiple Document Interface). Windows-програма типу MDI може підтримувати роботу водночас з декількома документами або формами, що спрощує розуміння та роботу з програмою. В MDI-програмах одна з форм об'являється як головна, яка вміщує елементи управління для виклику дочірніх форм. Функціональність програми реалізується в дочірніх формах. Головною формою у проекті є Form1.

Більшість Windows-програм мають головне меню, яке представляє собою ієрархічну структуру виконуючих функцій та команд. Для створення головного меню використовується елемент управління MenuStrip, розміщений на Панелі елементів. У головному меню розміщуються наступні пункти головного меню: «Звіти», «Обробка даних», «Пошук» та «Про програму». В пункті «Звіти» будуть відображатись такі пункти як: «Список співробітників»; пункт «Журнал відвідувань»; пункт «Зарплати»; пункт «Премії»; пункт «Штрафи» та «Відпустки».

Інтерфейс користувача має вигляд, який представлено на рисунку 3.3

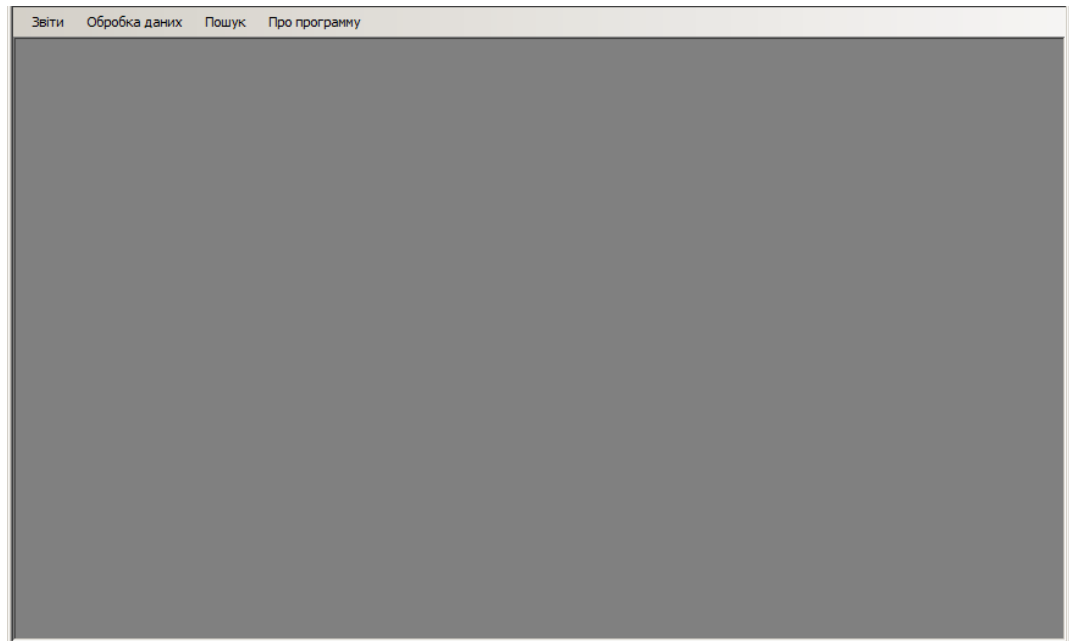


Рисунок 3.3 – Головне вікно клієнтської програми автоматизованої системи

За допомогою технології ADO.NET було реалізовано прив'язку клієнтського додатку до бази даних обліку кадрів [23].

Наприклад, для відображення представлення «Співробітники» в клієнтській програмі було створено дочірню форму, на якій розміщено елемент управління DataGridView. Далі за допомогою графічного інтерфейсу Visual Studio та технології ADO.NET було здійснено підключення цього елементу управління до створеної бази даних, а саме представлення «Співробітники». Аналогічно було підключено всі інші представлення та довідкові таблиці в клієнтській програмі автоматизованої системи обліку кадрів. Для того, щоб форми могли взаємодіяти з базою даних через певний драйвер ADO.NET необхідно задіяти Адміністратор джерел даних ADO.NET та додати нове джерело даних.

На рисунках 3.4 – 3.6 представлено таблиці які знаходяться в розділі «Звіти».

На рисунку 3.4 представлено таблицю «Відпустки»

	Name	SecondName	VacationID	StartDate	EndDate
▶	Дмитро	Таран	4	01.08.2018	31.08.2018
*					

Рисунок 3.4 – Вікно таблиці «Відпустки»

	Name	SecondName	Salary
▶	Дмитро	Таран	3500,0000
*			

Рисунок 3.5 – Вікно таблиці «Зарплати».

На рисунку 3.6 наведено таблицю співробітників підприємства.

	Name	SecondName	Address	BirthDate	Post
▶	Дмитро	Таран	пров. Глухівськи...	11.08.1994	Провізор
*					

Рисунок 3.6 – Вікно «Список співробітників»

Для зберігання та оброблення інформації щодо співробітників реалізовано форми для додавання та видалення інформації про співробітників на підприємстві в розділі «Обробка Інформації»

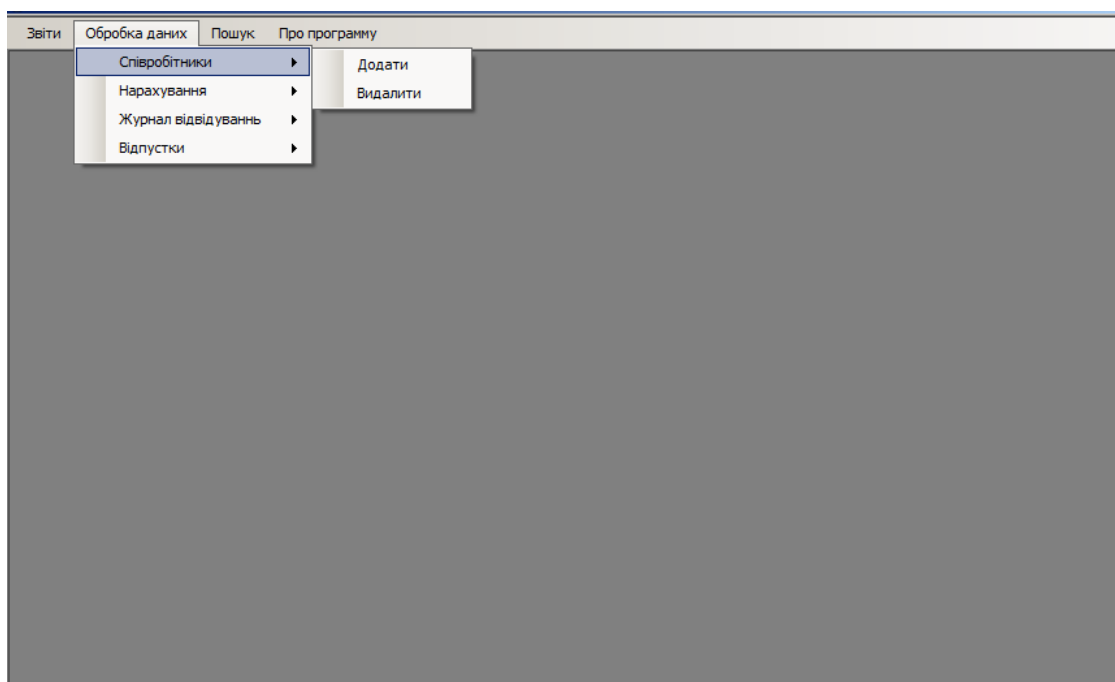


Рисунок 3.7 – Вікно «Обробка даних»

Для реалізації головної задачі, тобто додавання співробітників та видалення співробітників представлено на рисунку 3.8.

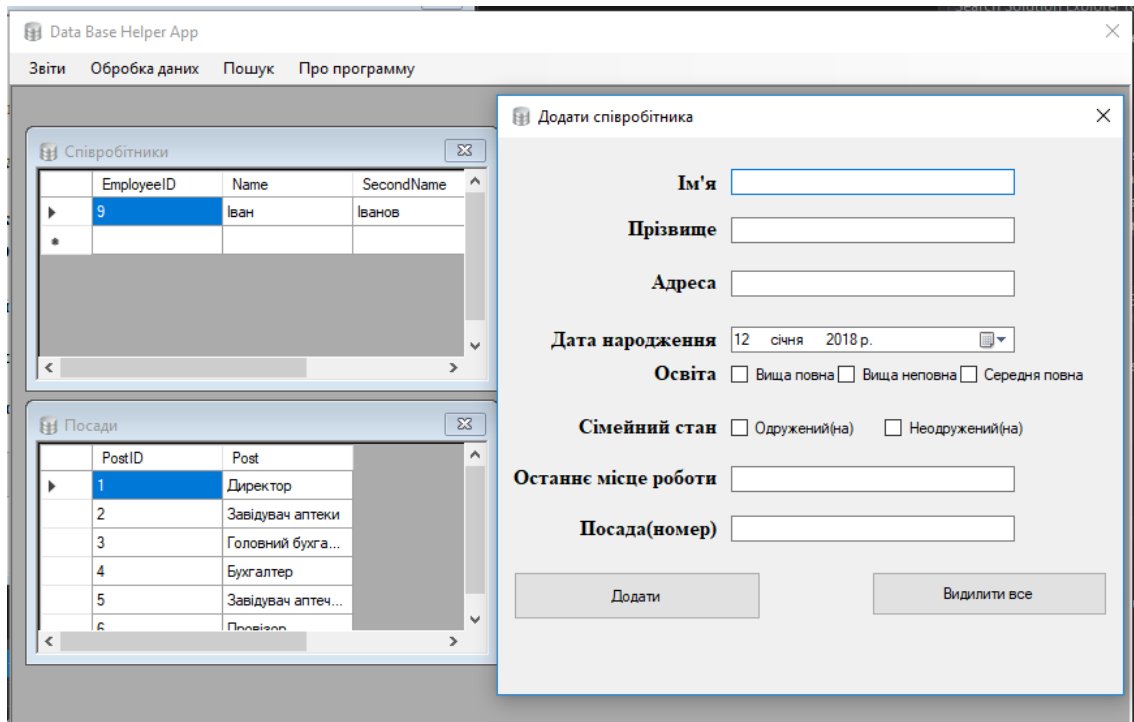


Рисунок 3.8 Додавання співробітника

Для зручності співробітнику присвоюється власний ID та для посади використовується такий сам принцип, посада вписується цифрою відповідно до списку посад, щоб не було орфографічної помилки та для зручності пошуку даних про співробітника.

Для видаленням співробітника достатньо ввести його ID. Вікно видалення зображено на рисунку 3.9.

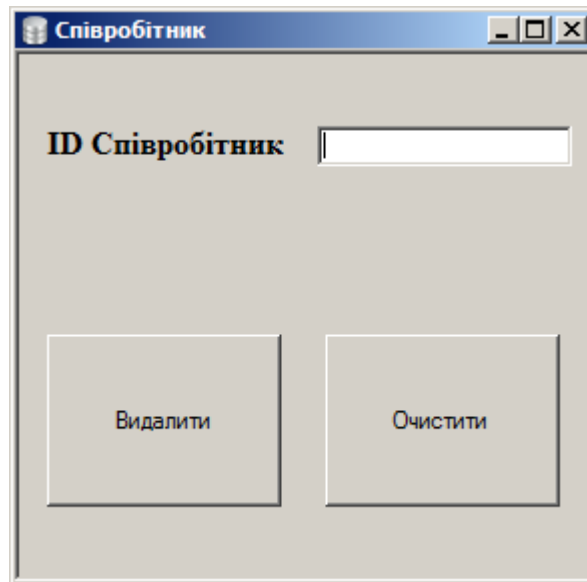


Рисунок 3.9 – Вікно «Видалення співробітника»

Для нарахування зарплати треба ввести ID співробітника та кількість нарахованої зарплати. Вікно додавання даних про зарплату відображено на рисунку 3.10.

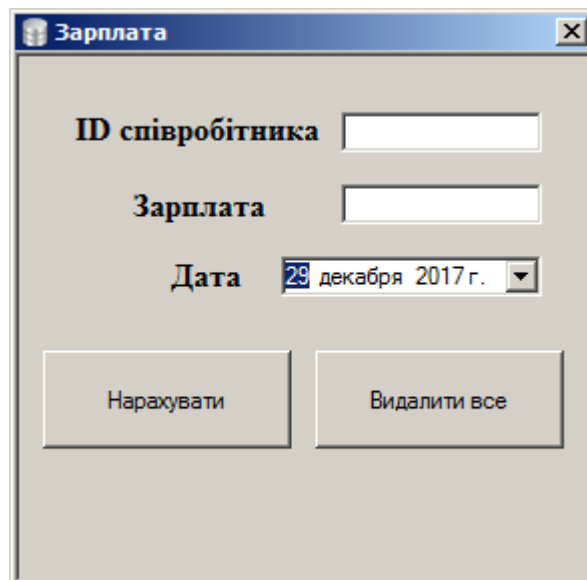
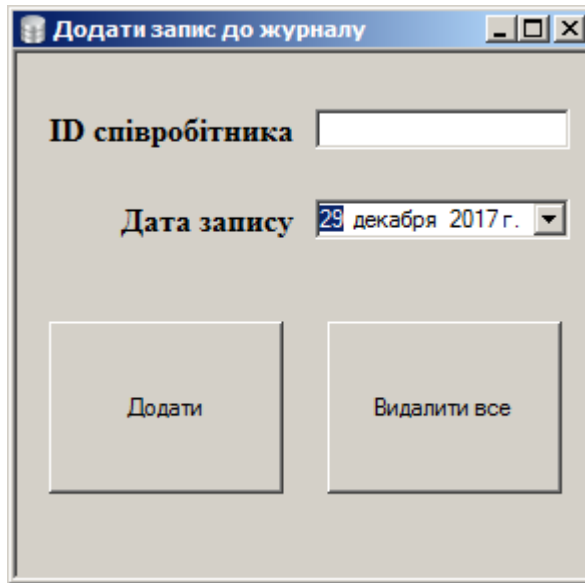


Рисунок 3.10 – Вікно «Нарахування зарплати»

Таким же чином здійснюється нарахування премії та штрафу. Для запису відвідування треба ввести ID співробітника та дату його перебування на роботі. Вікно додавання запису до журналу відображено на

рисунку 3.11.



Додати запис до журналу

ID співробітника

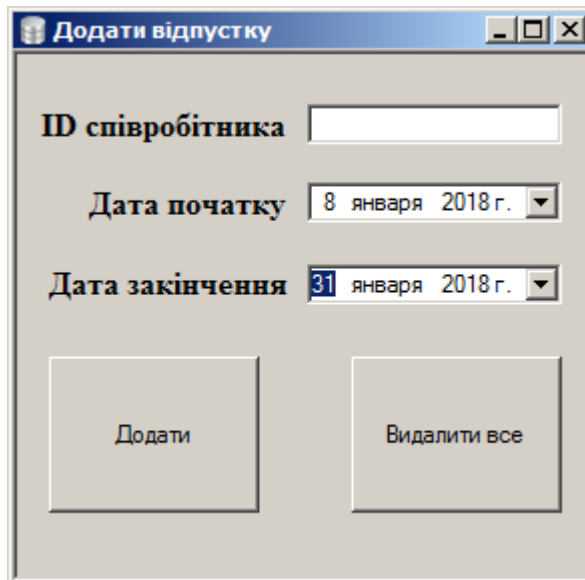
Дата запису 29 грудня 2017 г. ▼

Додати

Видалити все

Рисунок 3.11 – Вікно «Запис до журналу»

Для додавання даних про відпустку, треба ввести ID співробітника та період відпустки. Вікно додавання відпустки відображено на рисунку 3.11.



Додати відпустку

ID співробітника

Дата початку 8 січня 2018 г. ▼

Дата закінчення 31 січня 2018 г. ▼

Додати

Видалити все

Рисунок 3.12 – Вікно «Додавання відпусток»

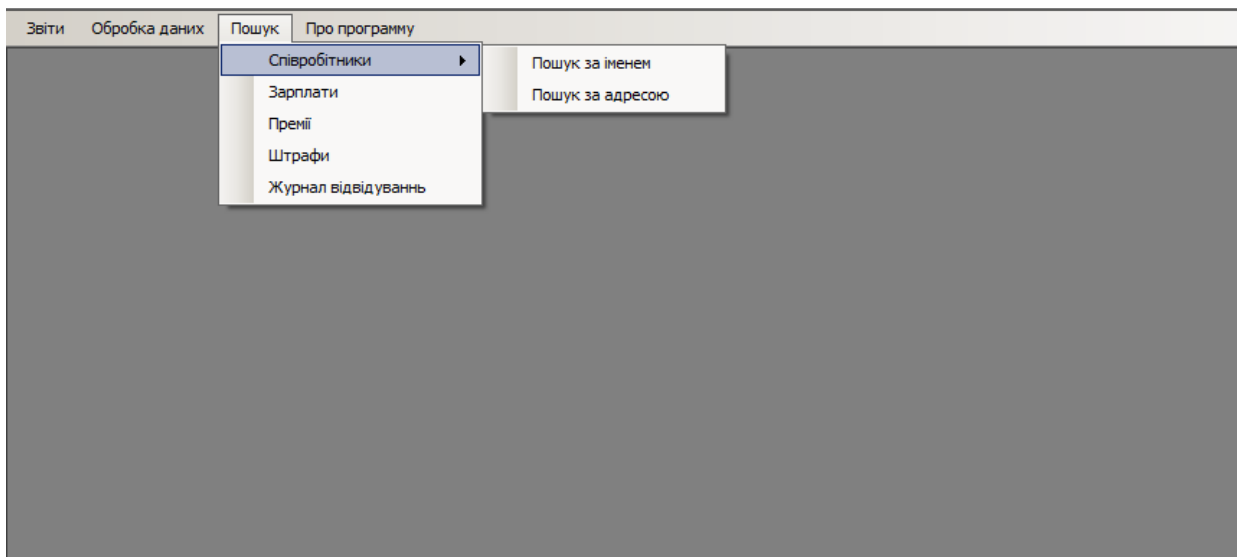


Рисунок 3.13 – Вікно «Пошук»

В розділі «Пошук» знаходяться пункти:

- пошук співробітника за ім'ям;
- пошук співробітників за адресою;
- пошук зарплати;
- пошук відвідувань.

Для пошуку співробітника за ім'ям треба ввести ім'я та прізвище співробітника. Вікно відображено на рисунку 3.14.

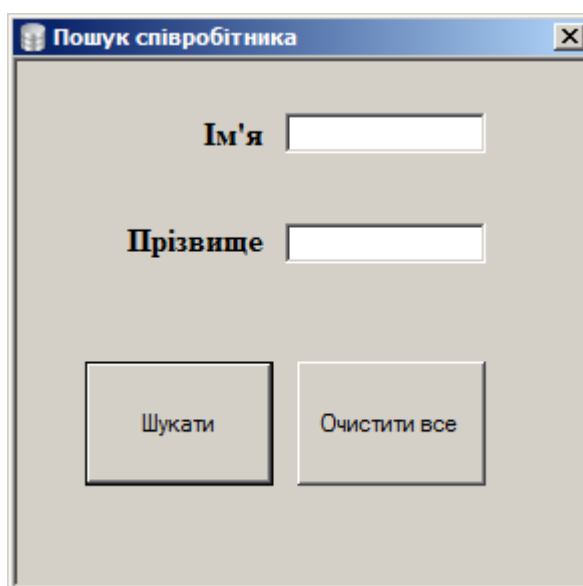


Рисунок 3.14 – Вікно «Пошук за ім'ям»

Для пошуку співробітника за адресою достатньо ввести адресу потрібної людини. Вікно пошуку за адресою відображене на рисунку 3.15.

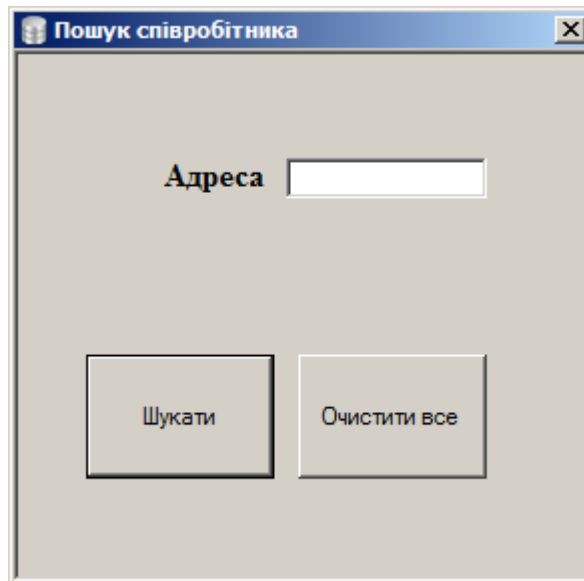


Рисунок 3.15 – Вікно «Пошук за адресою»

Для перегляду нарахованої зарплати співробітника потрібно вибрати дату нарахування. Вікно пошуку зарплати відображено на рисунку 3.16.

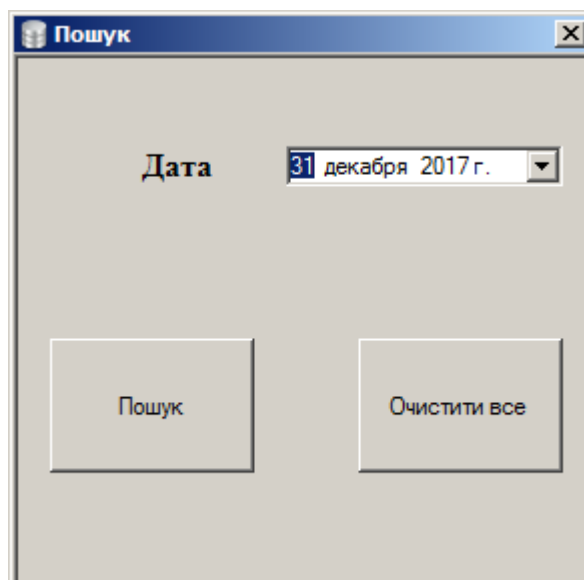


Рисунок 3.16 – Вікно «Пошук зарплати»

Для пошуку даних відвідування, достатньо ввести ID співробітника.

Вікно пошуку відображене на рисунку 3.17.

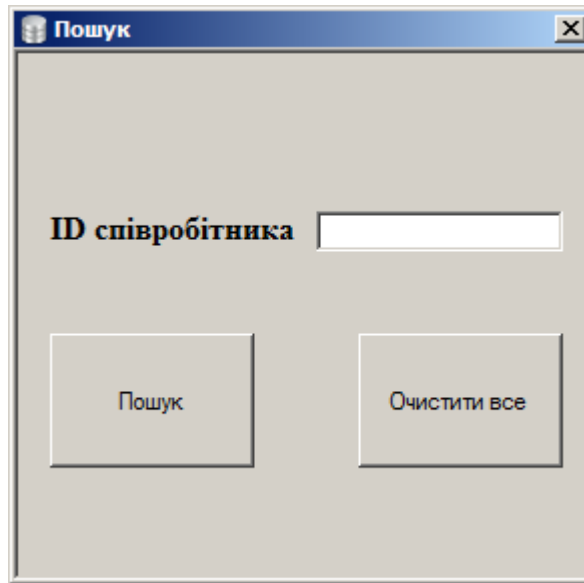


Рисунок 3.17 – Вікно «Пошук відвідувань»

Після введення ID співробітника виводиться таблиця відвідувань.

Одним з поширених елементів Windows-додатків є вікно «Про програму», в якому наводиться довідкова інформації про розробника програми. На рисунку 3.18 наведено інформаційне вікно проекту автоматизованої системи обліку кадрів.

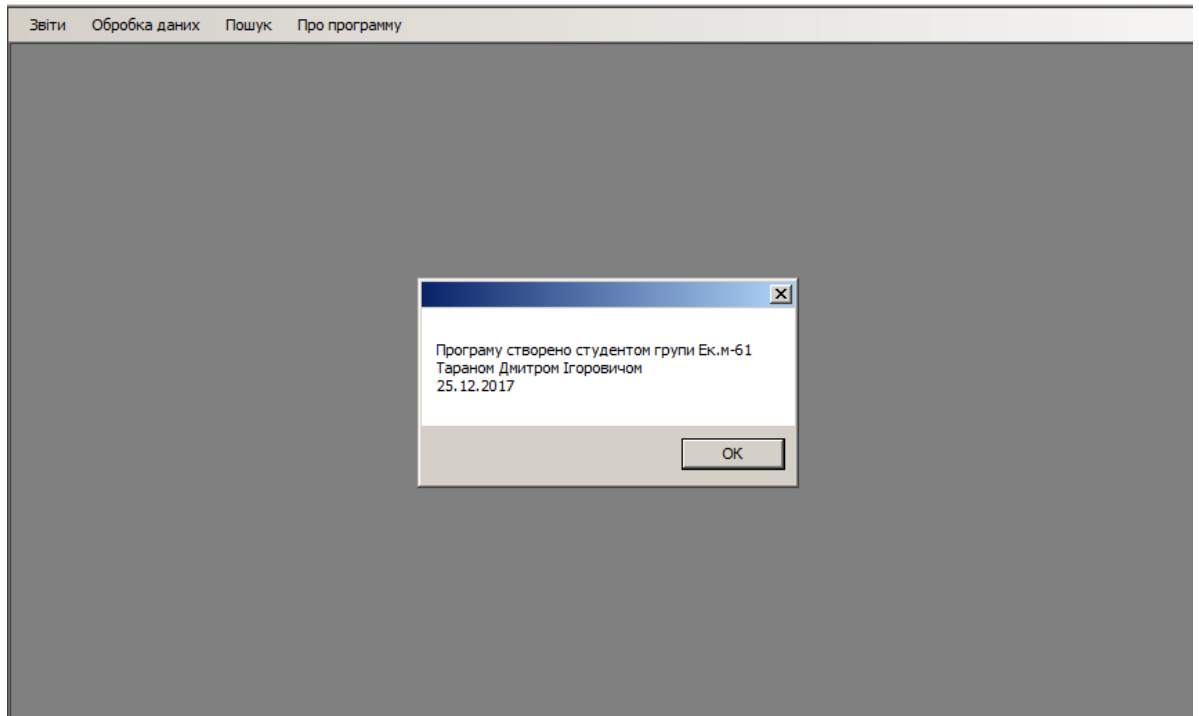


Рисунок 3.18 – Вікно «Про програму»

У додатку Д наведено код головної форми клієнтського додатку автоматизованої системи.

3.4 Оцінка очікуваного ефекту від впровадження системи автоматизації

Основними очікуваними ефектами від впровадження автоматизованої системи обліку кадрів можна вважати наступні:

а) полегшення роботи працівника відділу кадрів на ділянці обліку наявності та руху кадрів. Працівник відділу кадрів звільнюється від рутинної праці і може більше часу виділити для інших своїх функцій;

б) зменшення ймовірності помилок при вводі;

в) зручний інтерфейс, що дає можливість працювати користувачу, який мало знайомий з персональним комп'ютером і спрощує його дії до мінімуму.

Розглянемо кожний з пунктів детальніше.

Зменшення часу на обробку первинних документів та формування звітів призведе до:

- збільшення часу для співробітників відділу кадрів на виконання інших функцій на підприємстві, що призводить до підвищення ефективності праці;

- надає можливість скоріше реагувати на зміни у обліковому складі працівників та приймати швидкі управлінські рішення, що може призвести до зменшення ризику втрати ймовірної вигоди;

- зменшення помилок при вводі вхідної інформації також безпосередньо впливає на правильність та швидкодію управлінських рішень.

Простий інтерфейс програми дозволяє користуватися системою спеціалістам з поверховими знаннями персонального комп'ютера, а також економити кошти на навчання або перекваліфікацію співробітників відділу кадрів.

Результатом роботи системи є списки, що мають на меті допомогти у прийнятті ефективних управлінських рішень керівництву підприємства.

Основними очікуваними ефектами від впровадження автоматизованої системи управління комерційною клінікою можна вважати наступні:

- зменшення трудовитрат на ручну обробку інформації;
- запобігання ймовірності виникнення помилок при ручному записі;
- підвищення продуктивності праці бухгалтера-кадровика аптеки за рахунок автоматизованої обробки інформації;
- підвищення швидкості пошуку інформації про робітника.

Ефективність та продуктивність від провадження автоматизованої системи визначають порівнянням результатів її роботи і затрат всіх видів ресурсів, необхідних для її створення і розвитку.

Розрахуємо витрати на розробку та впровадження прототипу автоматизованої системи. Для розробки прототипу автоматизованої системи використовується праця одного програміста. Якщо притяти заробітну плату програміста в середньому 10000 грн. У зв'язку з відсутністю відповідного листа Мінсоцполітики (на момент написання дипломної роботи) при 40-

годинному робочому тижні норма робочого часу на 2018 рік становить 1993 годин на рік. Тоді кількість робочих годин за місяць становить:

$$T_r = 1993/12=166,08 \text{ (годин)}$$

Трудомісткість розробки прототипу автоматизованої системи приймемо рівною 0,5 місяця, тобто $166 \times 0,5=83$ години.

Розрахуємо середню заробітну плату програміста за годину:

$$= \text{_____} \quad (3.1)$$

де M_k – місячна заробітна плата розробника, грн.;

F_k^{mic} – місячний час роботи розробника, годин.

Тоді середня заробітна плата для програміста за годину складе:

$$r_k = 10000/166= 60,24 \text{ (грн./год.)}$$

Отже, витрати на розробку програмного продукту складають:

$$Z_p = 60,24 \times 83 \times 1,22 = 6099,90 \text{ (грн.)},$$

де 1,22 – коефіцієнт нарахування ЄСВ на заробітну плату.

Витрати комп'ютерного часу розраховуються за формулою:

$$Z_{кч} = C \times o \quad (3.2)$$

де C_k – вартість комп'ютерної години, грн.;

F_o – витрати комп'ютерного часу на розробку програми, год.

Вартість комп'ютерної години обчислюється по формулі:

$$= + \varepsilon + \tau \quad (3.3)$$

де C_A – амортизаційні відрахування, грн.; C_ϕ – енерговитрати, грн.;
 C_{TO} – витрати на техобслуговування, грн.

Розрахуємо амортизаційні витрати за наступною формулою:

$$C_A = C_i * N_A / F_{год} \quad (3.4)$$

де $C_i = 18000$ грн. – балансова вартість устаткування, яке використовувалося для створення програмного продукту, грн.;

$N_A 50\%$ – річна норма амортизації устаткування, долі;

$F_{год} = 1993$ – річний фонд часу роботи устаткування, година

Згідно з новим п. 43 підп. 4 р. ХХ ПКУ (платники податку на прибуток можуть застосовувати прискорену амортизацію для четвертої групи (машини й обладнання) ОЗ. Так, під час розрахунку амортизації за прямолінійним методом щодо ОЗ четвертої групи можливо використовувати починаючи з 01.01.2017 року мінімально допустимий строк амортизації, який дорівнює 2 роки :1 рік -50%, 2 рік-50% від вартості

$$C_A = 18000 * 50\% / 1993 = 4,52 \text{ (грн.)}$$

Енерговитрати розрахуємо за формулою:

$$C_E = P_E \times C_{кВт} \quad (3.5)$$

Комп'ютер, призначений для розробки програмного забезпечення, в середньому витрачає 400 Вт за годину, тоді $P_E = 0,04$ кВт/год.. Вартість 1

кВт/год. для непромислових підприємств першого класу становлять 2,04грн.

Тоді енерговитрати складуть:

$$C_E = 0,04 \times 2,04 = 0,08 \text{ (грн./год.)}$$

Витрати на технічне обслуговування розраховуємо за формулою:

$$C_{TO} = r_{TO} \times \lambda \quad (3.6)$$

де r_{TO} 20,00– годинна заробітна плата працівника обслуговуючого устаткування, грн;

λ – періодичність обслуговування.

Оскільки штатного технічного персоналу, з обслуговування обладнання, аптека не має, то використовує стороннього спеціаліста. Його ставка за 1 годину роботи дорівнює 20 грн.

Періодичність обслуговування розраховуємо за формулою:

$$\lambda = N_{TO} / F_{mec} \quad (3.7)$$

де N_{TO} – кількість обслуговувань устаткування в місяць.

Місячний фонд часу роботи устаткування 166 годин, кількість

устаткування – 1 комп'ютер. Тоді періодичність обслуговування буде складати:

$$\lambda = 1/166 = 0,006$$

Витрати на техобслуговування складуть:

$$C_{TO} = 20 \times 0,006 = 0,12 \text{ (грн.)},$$

тоді вартість комп'ютерної години:

$$C_k = 4,52 + 0,08 + 0,12 = 4,72 \text{ (грн.)},$$

витрати комп'ютерного часу складуть:

$$Z_{кч} = 4,72 \times 166 = 783,52 \text{ (грн.)}$$

Вартість ліцензійного програмного забезпечення складає для MS SQL 2017 Standard (931\$)[18], що по курсу НБУ станом на 05.01.2018р (28,0133 грн) становитиме 26080 грн. Розрахуємо сукупні витрати на створення автоматизованого програмного рішення для обліку кадрів аптеки:

$$V_{заг} = 783,52 + 6099,90 + 26080 = 32963,42 \text{ (грн.)}$$

Розрахуємо вигоду від впровадження автоматизованої системи, що розробляється.

В результаті впровадження системи у працівника вивільнився додатковий час, тобто система дозволяє зекономити в середньому 30-60 хв. В день. До впровадження системи час, затрачений на обробку кадрової документації, становив 60-120 хв. Після впровадження системи економія часу становить приблизно 50%. Оклад бухгалтера становить 6000грн. Річний фонд оплати праці бухгалтера-кадровика аптеки складає 72000грн. Час робот коштує 36,13 грн.(72000/1993 години). У 2018р кількість робочих днів дорівнює 250. Річна економія розраховується за формулою:

$$E_p = V_{дв} - V_{пв}, \tag{3.8}$$

де E_p – сума річної економії, грн.;

$V_{дв}$ – річні витрати на обробку інформації до впровадження системи, грн.;

$V_{пв}$ – річні витрати на обробку інформації після впровадження системи, грн.

Тоді річна економія ставить:

$$E_p = ((36,13 \times 2) \times 250) - (36,13 \times 250) = 9032,5 \text{ (грн.)}$$

Розрахуємо термін окупності за формулою:

$$T_o = \frac{B_{\text{заг}}}{E_p}$$

(3.9)

Термін окупності автоматизованого програмного рішення становить:

$$T_o = \frac{32063,42}{9032,50} = 3,65 \text{ (роки)}$$

Таким чином, можна казати про економічну обґрунтованість вкладання коштів та зусиль у розробку програмного автоматизованого рішення для обліку кадрів аптеки. При цьому кошти, витрачені на розробку та впровадження такої системи, окупиться приблизно за 3,6 роки.

ВИСНОВКИ

Облік наявності та руху кадрів – відповідальний сектор робіт, який потребує точності, своєчасності та достовірності, й крім того, достатньо трудомісткий процес. Автоматизація цих процесів дозволить значно підвищити продуктивність праці робітника відділу кадрів.

У роботі відповідно до мети дослідження – розробка прототипу автоматизованої системи кадрового обліку підприємства – отримані такі результати:

- надана характеристика об’єкту автоматизації;
- проаналізовано поточний стан об’єкту автоматизації;
- визначені бізнес-процеси підприємства;
- розроблено концепцію автоматизації діяльності об’єкту дослідження;
- проаналізовані існуючі інформаційні системи для автоматизації діяльності кадрового відділу;
- сформульовані вимоги до створюваної системи;
- розроблено проект автоматизованої системи обліку кадрів;
- розроблено прототип інформаційної системи кадрового обліку підприємства;

Практичним результатом розв’язання задач дипломної роботи є прототип автоматизованої системи кадрового обліку, що має базовий функціонал та зручний інтерфейс користувача. Розроблений прототип системи може бути розширено додатковими модулями та впроваджено на малих та середніх підприємствах.

В дослідженні використано такий інструментарій програмування: MS SQL Server, Transact-SQL, Microsoft Visual Studio, C#.

Розвитком побудованої системи может бути розширення її функціоналу підключенням додаткових модулів з законодавства, реєстрації і розрахунку листків непрацездатності; роботи з кадровим резервом.

СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ

1. Введение в язык C# и .NET Framework [Электронный ресурс]. – Режим доступа до ресурсу: <http://msdn.microsoft.com/ru-ru/library/z1zx9t92%28VS.90%29.aspx>.
2. Винокуров М.А. Автоматизация кадрового учета // М.А. Винокурова Методы менеджмента качества. – М.: ИНФРА-М. – 2009. – №3.– С. 23.
3. Волохов В. Сучасні інформаційні технології кадрового менеджменту на державній службі [Електронний ресурс] / В. Волохов, Ю. Щепотко // Вісник державної служби України. – 2008. – № 3. – Режим доступа до ресурсу: <http://www.guds.gov.ua/control/uk/publish/article>.
4. Глинских А. Современные автоматизированные системы управления персоналом/ А. Глинских: [Электронный ресурс]. – Режим доступа до ресурсу: <http://www.mnogosmenka.ru/drugoe/personal.htm>.
5. Дорожная карта по системам управления персоналом (HRMS) [Электронный ресурс]. – Режим доступа до ресурсу: http://www.bsiconsulting.ru/advantages/hrms_boss_kadrovik/hrms_roadmap.
6. Економічний аналіз господарської діяльності підприємства [Електронний ресурс]. – Режим доступа до ресурсу: http://pidruchniki.com/18000102/ekonomika/ekonomichniy_analiz_gospodarskoyi_diyalnosti_pidpriyemstv.
7. Информационная система [Электронный ресурс]. – Режим доступа до ресурсу: http://ru.wikipedia.org/wiki/Информационная_система.
8. Иванов И.И. Огляд програм для автоматизації HRM / И.И. Иванов Управління персоналом. – 2009. – № 11. – С.23.
9. Інформаційна система [Електронний ресурс]. – Режим доступа до ресурсу: https://uk.wikipedia.org/wiki/Інформаційна_система.

10. Класс GridView [Електронний ресурс]. – Режим доступу до ресурсу: [https://msdn.microsoft.com/ru-ru/library/system.web.ui.webcontrols.gridview\(v=vs.110\).aspx](https://msdn.microsoft.com/ru-ru/library/system.web.ui.webcontrols.gridview(v=vs.110).aspx).
11. Клименко О. Інформаційні системи і технології в обліку: Навчальний посібник / О. Клименко; М-во освіти і науки України, Полтавський ун-т споживчої кооперації України. – К.: Центр учбової літератури, 2008. - 319 с.
12. Литвин, І.С. Інформаційні технології в економіці / І.С. Литвин. – Тернопіль: навч. посібник "Економічна думка", 2011. – 296 с. – ISBN 5-7763-0459.
13. Методические основы создания информационных систем и информационных технологий в управлении организацией [Електронний ресурс]. – Режим доступу: <http://miemp-mi-gor.narod.ru/utcheba/itu/glava/002.htm>
14. Нотация IDEF0 [Електронний ресурс] // Business Studio Wiki. – Режим доступу до ресурсу: <http://www.businessstudio.ru/wiki/docs/v4/doku.php/ru/csdesign/bpmodeling/idef0>.
15. Особенности клиент-сервер [Електронний ресурс]. – Режим доступу до ресурсу: http://citforum.ck.ua/programming/application/builder_cs3.shtml.
16. Прайс-лист продуктів Microsoft [Електронний ресурс]. – Режим доступу до ресурсу: <https://www.microsoft.com/ru-ru/sql-server/sql-server-2017-pricing>.
17. Ресурсне забезпечення менеджменту персоналу [Електронний ресурс]. – Режим доступу до ресурсу: http://lubbook.org/book_336_glava_17_Rozd%D1%96l_3._Resursne_zabezpeche.html.
18. Розробка ІС [Електронний ресурс]. – Режим доступу: <https://xreferat.com/33/6794-3-razrabotka-informacionnoiy-sistemy-sluzhba-zanyatosti.html>.
19. Суперключ [Електронний ресурс]. – Режим доступу до ресурсу: <https://ru.wikipedia.org/wiki/Суперключ>.

20. Юрченко І.В. Інформатика та програмування. Частина 1. Навчальний посібник / І.В. Юрченко – Чернівці: Книги–ХХІ, 2011.– 203 с.
21. Юрченко І.В. Інформатика та програмування. Частина 2 / І.В. Юрченко, В.С. Сікора. – Чернівці: Видавець Яворський С.Н., 2015. – 210 с.
22. .NET Framework (.NET) [Електронний ресурс].– Режим доступу до ресурсу: <https://www.techopedia.com/definition/3734/net-framework-net>.
23. ADO.NET [Електронний ресурс]. – Режим доступу до ресурсу: <https://uk.wikipedia.org/wiki/ADO.NET>.
24. Automation and the end of accounting [Електронний ресурс]. – Режим доступу до ресурсу: <https://www.computerworld.com/article/3195070/financial-it/automation-and-the-end-of-accounting.html>.
25. CREATE PROCEDURE (Transact-SQL) [Електронний ресурс]. – Режим доступу до ресурсу: <https://docs.microsoft.com/en-us/sql/t-sql/statements/create-procedure-transact-sql>.
26. IDEF0 [Електронний ресурс]. – Режим доступу до ресурсу: <https://en.wikipedia.org/wiki/IDEF0>.
27. Install SQL Server 2014 [Електронний ресурс]. – Режим доступу до ресурсу: [https://msdn.microsoft.com/en-us/library/bb500395\(v=sql.120\).aspx](https://msdn.microsoft.com/en-us/library/bb500395(v=sql.120).aspx).
28. Introduction to the C# Language and the .NET Framework [Електронний ресурс]. – Режим доступу до ресурсу: <https://docs.microsoft.com/en-us/dotnet/csharp/getting-started/introduction-to-the-csharp-language-and-the-net-framework>.
29. Introduction to the C# Language and the .NET Framework [Електронний ресурс]. – Режим доступу до ресурсу: <https://docs.microsoft.com/en-us/dotnet/csharp/getting-started/introduction-to-the-csharp-language-and-the-net-framework>.
30. Microsoft SQL Server [Електронний ресурс] // The TechTarget network. – Режим доступу до ресурсу: <http://searchsqlserver.techtarget.com/definition/SQL-Server>.

31. Microsoft Visual Studio [Электронный ресурс]. – Режим доступа до ресурсу: https://ru.wikipedia.org/wiki/Microsoft_Visual_Studio
32. Overview of the .NET Framework [Электронный ресурс]. – Режим доступа до ресурсу: <https://docs.microsoft.com/en-us/dotnet/framework/get-started/overview>.
33. Skonnard, A. XML in .NET: .NET Framework XML Classes and C# Offer Simple, Scalable Data Manipulation [Электронный ресурс]. – Режим доступа до ресурсу: <http://msdn.microsoft.com/en-us/magazine/cc302158.aspx>.
34. SQL CREATE TABLE Statement [Электронный ресурс]. – Режим доступа до ресурсу: https://www.w3schools.com/sql/sql_create_table.asp.
35. Visual Studio 2017 [Электронный ресурс]. – Режим доступа до ресурсу: <https://www.microsoft.com/ru-ru/SoftMicrosoft/vs2017>.
36. Windows [Электронный ресурс]. – Режим доступа до ресурсу: <https://ru.wikipedia.org/wiki/Windows>.

ДОДАТКИ

Додаток А

SUMMARY

Taran D.I. Automation of personnel accounting of the enterprise. – Masters-level Qualification Thesis. Sumy State University, Sumy, 2017.

In the course of the project, the implementation of the business process, the visibility of the business processes, the initialization of the system for the automation of the personnel recruitment, the formulation of the work before the installation, the project is the prototype of the automated system, the economical effect is identified in the system.

Keywords: automation, personnel accounting, Microsoft Visual Studio, motion frames, Microsoft SQL Server.

АНОТАЦІЯ

Таран Д.І. Автоматизація кадрового обліку підприємства. – Кваліфікаційна магістерська робота. Сумський державний університет, Суми, 2017 р.

В роботі досліджено діяльність підприємства з обліку кадрів, визначені бізнес-процеси, проаналізовані існуючі інформаційні системи для автоматизації діяльності кадрового відділу, сформульовані вимоги до створюваної системи, розроблено проект та прототип автоматизованої системи, визначено економічний ефект від впровадження системи.

Ключові слова: автоматизація, кадровий облік, Microsoft Visual Studio, рух кадрів, Microsoft SQL Server.

Додаток Б

Програмний код створення бази даних та базових таблиць

```

--Создание базы данных-- CREATE
DATABASE AccountingStaff ON
(
    NAME = 'dbAccountingStaff_Data',
    FILENAME = 'D:\Work\DiplomaWork\DataBase\AccountingStaff_data.mdf',
    SIZE = 50 MB,
    MAXSIZE = 2000 MB,
    FILEGROWTH = 10%
)
LOG ON
(
    NAME = 'dbAccountingStaff_Log',
    FILENAME = 'D:\Work\DiplomaWork\DataBase\AccountingStaff_log.ldf',
    SIZE = 50 MB,
    MAXSIZE = 200 MB,
    FILEGROWTH = 10%
)
GO

--Подключаемся к нашей базе--
USE AccountingStaff
GO

--Создаем нужные таблицы--

--Должность--
CREATE TABLE Posts
(
    PostID INT IDENTITY(1,1),
    Post NVARCHAR(30) NOT NULL,

    CONSTRAINT PK_Posts PRIMARY KEY(PostID)
)
GO

--Сотрудник--
CREATE TABLE Employees
(
    EmployeeID INT IDENTITY(1,1),
    Name NVARCHAR(30) NOT NULL,
    SecondName NVARCHAR(30) NOT NULL,
    [Address] NVARCHAR(30) NOT NULL,
    BirthDate DATE NOT NULL,
    PostID INT NOT NULL,

    CONSTRAINT PK_Employees PRIMARY KEY(EmployeeID),
    CONSTRAINT FK_Employees_Posts FOREIGN KEY(PostID) REFERENCES Posts(PostID)
)
GO

--Статус сотрудника--
CREATE TABLE Statuses
(
    StatusID INT IDENTITY(1,1),
    EmployeeID INT NOT NULL,
    [Status] BIT NOT NULL,
    [Date] DATE DEFAULT GETDATE()
)

```



```

        CONSTRAINT PK_Statuses PRIMARY KEY(StatusID),
        CONSTRAINT FK_Statuses_Employees FOREIGN KEY(EmployeeID) REFERENCES
Employees(EmployeeID)
    )
GO

```

--Зарплаты--

```

CREATE TABLE Salaries
(
    SalaryID INT IDENTITY(1,1),
    EmployeeID INT NOT NULL,
    Salary MONEY NOT NULL,
    [Date] DATE DEFAULT GETDATE(),

    CONSTRAINT PK_Salaries PRIMARY KEY(SalaryID),
    CONSTRAINT FK_Salaries_Employees FOREIGN KEY(EmployeeID) REFERENCES
Employees(EmployeeID),
    CONSTRAINT CH_Salaries_Salary CHECK(Salary > 0)
)
GO

```

--Штрафы--

```

CREATE TABLE Fines
(
    FineID INT IDENTITY(1,1),
    EmployeeID INT NOT NULL,
    Fine MONEY NOT NULL,
    [Date] DATE DEFAULT GETDATE(),

    CONSTRAINT PK_Fines PRIMARY KEY(FineID),
    CONSTRAINT FK_Fines_Employees FOREIGN KEY(EmployeeID) REFERENCES
Employees(EmployeeID),
    CONSTRAINT CH_Fines_Fine CHECK(Fine > 0)
)
GO

```

--Премии--

```

CREATE TABLE Bonuses
(
    BonusID INT IDENTITY(1,1),
    EmployeeID INT NOT NULL,
    Bonus MONEY NOT NULL,
    [Date] DATE DEFAULT GETDATE(),

    CONSTRAINT PK_Bonuses PRIMARY KEY(BonusID),
    CONSTRAINT FK_Bonuses_Employees FOREIGN KEY(EmployeeID) REFERENCES
Employees(EmployeeID),
    CONSTRAINT CH_Bonuses_Bonus CHECK(Bonus > 0)
)
GO

```

--Отпуска--

```

CREATE TABLE Vacations
(
    VacationID INT IDENTITY(1,1),
    EmployeeID INT NOT NULL, StartDate
DATE DEFAULT GETDATE(), EndDate
DATE NOT NULL,

    CONSTRAINT PK_Vacations PRIMARY KEY(VacationID),
    CONSTRAINT FK_Vacations_Employees FOREIGN KEY(EmployeeID) REFERENCES
Employees(EmployeeID),

```

```
        CONSTRAINT CH_Vacation_EndDate CHECK(EndDate > StartDate)
    )
GO

--Журнал посещений работы--
CREATE TABLE WorkLog
(
    LogID INT IDENTITY(1,1),
    EmployeeID INT NOT NULL,
    [Date] DATE DEFAULT GETDATE()

    CONSTRAINT PK_WorkLog PRIMARY KEY(LogID),
    CONSTRAINT FK_WorkLog_Employees FOREIGN KEY(EmployeeID) REFERENCES
Employees(EmployeeID)
)
GO
```

Додаток В

Створення збережених процедур до бази даних

```

USE AccountingStaff
GO

--Удаление сотрудника и всей связанной информации--
CREATE PROCEDURE spDeleteEmployee
@EmployeeID INT
AS
DECLARE @LocalError AS INT;
BEGIN TRANSACTION

DELETE FROM Salaries WHERE EmployeeID = @EmployeeID;
DELETE FROM Bonuses WHERE EmployeeID = @EmployeeID;
DELETE FROM Fines WHERE EmployeeID = @EmployeeID;
DELETE FROM Statuses WHERE EmployeeID = @EmployeeID;
DELETE FROM WorkLog WHERE EmployeeID = @EmployeeID;
DELETE FROM Vacations WHERE EmployeeID = @EmployeeID;
DELETE FROM Employees WHERE EmployeeID = @EmployeeID;

SET @LocalError = @@ERROR;

IF @LocalError = 0
BEGIN
COMMIT TRAN;
PRINT 'Співробітника успішно видалено.';
END
ELSE
BEGIN
ROLLBACK TRAN;
PRINT 'Помилка видалення.';
END
GO

--Удаление информации об конкретном отпуске--
CREATE PROCEDURE spDeleteVacation
@VacationID INT
AS
DECLARE @LocalError INT
BEGIN TRANSACTION

DELETE FROM Vacations WHERE VacationID = @VacationID;

SET @LocalError = @@ERROR;

IF @LocalError = 0
BEGIN
COMMIT TRAN;
PRINT 'Запис про відпустку успішно видалений.';
END
ELSE
BEGIN
ROLLBACK TRAN;
PRINT 'Помилка видалення.';
END
GO

SELECT * FROM WorkLog
GO

--Удаление записи из журнала посещений--

```

```
CREATE PROCEDURE spDeleteLogFromWorkLog
@LogID INT
AS
DECLARE @LocalError INT
BEGIN TRANSACTION

DELETE FROM WorkLog WHERE LogID = @LogID;

SET @LocalError = @@ERROR;

IF @LocalError = 0
BEGIN
COMMIT TRAN;
PRINT 'Запис з журналу відвідувань успішно видалений.';
END
ELSE
BEGIN
ROLLBACK TRAN;
PRINT 'Помилка видалення.';
END
GO
```

Додаток Г

Приклад створення тригера

```
USE AccountingStaff
GO

--Функция подсчета общего количества денег для указанного сотрудника--
CREATE FUNCTION fCalcEmployeeTotalSalary(@EmployeeID AS INT)
RETURNS MONEY
AS
BEGIN
    DECLARE @salaries AS MONEY;
    SELECT @Salaries = SUM(Salary) FROM Salaries WHERE EmployeeID = @EmployeeID;
    DECLARE @bonuses AS MONEY;
    SELECT @bonuses = SUM(Bonus) FROM Bonuses WHERE EmployeeID = @EmployeeID;
    DECLARE @fines AS MONEY;
    SELECT @fines = SUM(Fine) FROM Fines WHERE EmployeeID = @EmployeeID;
    DECLARE @result AS MONEY;
    SET @result = @salaries+@bonuses+@fines;
    RETURN @result;
END
GO
```

Додаток Д

Код головної форми клієнтського додатку

```

using System;
using System.Drawing;
using DataBaseHelper.SelectingForms;
using System.Windows.Forms;
using DataBaseHelper.DataHelpers;
using DataBaseHelper.InsertingForms.Employees;
using DataBaseHelper.InsertingForms.Finance;
using DataBaseHelper.InsertingForms.Vacations;
using DataBaseHelper.InsertingForms.WorkLog;
using DataBaseHelper.DeletingForms;
using DataBaseHelper.SearchingForms;

namespace DataBaseHelper
{
    public partial class MainForm : Form
    {
        public MainForm()
        {
            InitializeComponent();
        }

        public void Closed(object sender, EventArgs e)
        {
            var arr = this.MdiChildren;
            foreach (var item in arr)
            {
                item.Close();
            }
        }

        private void ShowForm(Form form)
        {
            form.Show();
            form.Activate();
        }

        private void SetStartupEmployeesInsert(int command, InfoForm form)
        {
            if(command == 1)
            {
                form.Location = new Point(10, 30);
            }
            else if(command == 2)
            {
                form.Location = new Point(10, 240);
            }
        }

        private void SetStartupFinanceInsert(int command, InfoForm form)
        {
            if(command == 1)
            {

```

```

        form.Location = new Point(10, 30);
    }
    else if(command == 4 || command == 5 || command == 6)
    {
        form.Location = new Point(10, 240);
    }
}

private void SetStartupWorkLogInsert(int command, InfoForm form)
{
    if (command == 1)
    {
        form.Location = new Point(10, 30);
    }
    else if(command == 3)
    {
        form.Location = new Point(10, 240);
    }
}

private void SetStartupVacationsInsert(int command, InfoForm form)
{
    if (command == 1)
    {
        form.Location = new Point(10, 30);
    }
    else if (command == 8)
    {
        form.Location = new Point(10, 240);
    }
}

private void ShowInfoForm(InfoForm form,int command, int operation)
{
    form.MdiParent = this;
    form.StartPosition = FormStartPosition.Manual;

    switch(operation)
    {
        case 1: SetStartupEmployeesInsert(command,form);break;
        case 2: SetStartupFinanceInsert(command, form);break;
        case 3: SetStartupWorkLogInsert(command, form);break;
        case 4: SetStartupVacationsInsert(command, form);break;
    }

    form.Show();
    form.Activate();
    form.IsActivated = true;
}

#region SELECTION_EVENTS

private void співробітникиToolStripMenuItem_Click(object sender, EventArgs e)
{
    ShowForm(new SelectionForm("Список співробітників", 1));
}

private void журналВідвідуваньToolStripMenuItem2_Click(object sender, EventArgs
e)
{
    ShowForm(new SelectionForm("Журнал відвідувань", 2));
}

```

```

private void зарплатиToolStripMenuItem1_Click(object sender, EventArgs e)
{
    ShowForm(new SelectionForm("Зарплати", 3));
}

private void преміїToolStripMenuItem2_Click(object sender, EventArgs e)
{
    ShowForm(new SelectionForm("Премії", 4));
}

private void штрафиToolStripMenuItem2_Click(object sender, EventArgs e)
{
    ShowForm(new SelectionForm("Штрафи", 5));
}

private void відпусткиToolStripMenuItem3_Click(object sender, EventArgs e)
{
    ShowForm(new SelectionForm("Відпустки", 6));
}

#endregion

#region INSERTION_EVENTS

private void додатиToolStripMenuItem2_Click(object sender, EventArgs e)
{
    InfoForm empInfoForm = new InfoForm("Співробітники", 1);
    ShowInfoForm(empInfoForm, 1, 1);

    InfoForm postsInfoForm = new InfoForm("Посади", 2);
    ShowInfoForm(postsInfoForm, 2, 1);

    InsertEmployeeForm insertForm = new InsertEmployeeForm();
    ShowForm(insertForm);

    insertForm.CloseEvent += Closed;
}

private void нарахуванняToolStripMenuItem1_Click(object sender, EventArgs e)
{
    InfoForm empInfoForm = new InfoForm("Співробітники", 1);
    ShowInfoForm(empInfoForm, 1, 1);

    InfoForm postsInfoForm = new InfoForm("Заробітні плати", 4);
    ShowInfoForm(postsInfoForm, 4, 2);

    InsertingFinanceForm insertForm = new InsertingFinanceForm("Зарплата", 1);
    ShowForm(insertForm);

    insertForm.CloseEvent += Closed;
}

private void нарахуванняПреміїToolStripMenuItem_Click(object sender, EventArgs e)
{
    InfoForm empInfoForm = new InfoForm("Співробітники", 1);
    ShowInfoForm(empInfoForm, 1, 1);

    InfoForm postsInfoForm = new InfoForm("Премії", 5);
    ShowInfoForm(postsInfoForm, 4, 2);
}

```



```

        InsertingFinanceForm insertForm = new InsertingFinanceForm("Премія", 2);
        ShowForm(insertForm);

        insertForm.CloseEvent += Closed;
    }

private void нарахуватиШтрафToolStripMenuItem_Click(object sender, EventArgs e)
{
    InfoForm empInfoForm = new InfoForm("Співробітники", 1);
    ShowInfoForm(empInfoForm, 1, 1);

    InfoForm postsInfoForm = new InfoForm("Штрафи", 6);
    ShowInfoForm(postsInfoForm, 4, 2);

    InsertingFinanceForm insertForm = new InsertingFinanceForm("Штраф", 3);
    ShowForm(insertForm);

    insertForm.CloseEvent += Closed;
}

private void додатиToolStripMenuItem3_Click(object sender, EventArgs e)
{
    InfoForm empInfoForm = new InfoForm("Співробітники", 1);
    ShowInfoForm(empInfoForm, 1, 1);

    InfoForm vacationsForm = new InfoForm("Відпустки", 8);
    ShowInfoForm(vacationsForm, 8, 4);

    InsertingVacaationForm insertForm = new InsertingVacaationForm();
    ShowForm(insertForm);

    insertForm.CloseEvent += Closed;
}

private void додатиЗаписToolStripMenuItem_Click(object sender, EventArgs e)
{
    InfoForm empInfoForm = new InfoForm("Співробітники", 1);
    ShowInfoForm(empInfoForm, 1, 1);

    InfoForm workLogForm = new InfoForm("Журнал відвідувань", 3);
    ShowInfoForm(workLogForm, 3, 3);

    InsertingWorkLogFrom insertForm = new InsertingWorkLogFrom();
    ShowForm(insertForm);

    insertForm.CloseEvent += Closed;
}

#endregion

#region DELETING_EVENTS

private void видалитиToolStripMenuItem2_Click(object sender, EventArgs e)
{
    InfoForm empInfoForm = new InfoForm("Співробітники", 1);
    ShowInfoForm(empInfoForm, 1, 1);

    DeletingForm delForm = new DeletingForm("Співробітник", 1);
    ShowForm(delForm);

    delForm.CloseEvent += Closed;
}

```

```

}

private void видалитиТoolStripMenuItem3_Click(object sender, EventArgs e)
{
    InfoForm workLogInfoForm = new InfoForm("Відпустки", 8);
    ShowInfoForm(workLogInfoForm, 4, 8);

    DeletingForm delForm = new DeletingForm("Відпустка", 4);
    ShowForm(delForm);

    delForm.CloseEvent += Closed;
}

private void видалитиЗаписТoolStripMenuItem_Click(object sender, EventArgs e)
{
    InfoForm workLogInfoForm = new InfoForm("Журнал відвідувань", 3);
    ShowInfoForm(workLogInfoForm, 3, 3);

    DeletingForm delForm = new DeletingForm("Запис журналу", 3);
    ShowForm(delForm);

    delForm.CloseEvent += Closed;
}

#endregion

#region SEARCHING_EVENTS

private void пошукЗаІменемТoolStripMenuItem_Click(object sender, EventArgs e)
{
    SearchingEmployeeForm searching = new SearchingEmployeeForm(1);
    ShowForm(searching);
}

private void пошукЗаАдресомТoolStripMenuItem_Click(object sender, EventArgs e)
{
    SearchingEmployeeForm searching = new SearchingEmployeeForm(2);
    ShowForm(searching);
}

private void зарплатиТoolStripMenuItem3_Click(object sender, EventArgs e)
{
    SearchFinanceAndLogForm searching = new SearchFinanceAndLogForm(1);
    ShowForm(searching);
}

private void преміїТoolStripMenuItem3_Click(object sender, EventArgs e)
{
    SearchFinanceAndLogForm searching = new SearchFinanceAndLogForm(2);
    ShowForm(searching);
}

private void штрафиТoolStripMenuItem3_Click(object sender, EventArgs e)
{
    SearchFinanceAndLogForm searching = new SearchFinanceAndLogForm(3);
    ShowForm(searching);
}

private void журналВідвідуваньТoolStripMenuItem3_Click(object sender, EventArgs
e)
{
    SearchFinanceAndLogForm searching = new SearchFinanceAndLogForm(4);

```

```
        ShowForm(searching);
    }

    #endregion

    private void проПрограммуToolStripMenuItem_Click(object sender, EventArgs e)
    {
        MessageBox.Show("Програму створено студентом групи Ек.м-61а\nТараном Дмитром  
Ігоровичем\n19.12.2017");
    }
}
}
```