

РЕФЕРАТ

Кваліфікаційна робота на здобуття освітнього ступеню бакалавр за спеціальністю 121 – Інженерія програмного забезпечення. Тернопільський національний технічний університет ім. Івана Пулюя, факультет комп'ютерно-інформаційних систем і програмної інженерії, кафедра програмної інженерії, група СП-41, 2023 рік. Пояснювальна записка до кваліфікаційної роботи на здобуття освітнього ступеню бакалавр містить: 62 с., 26 рис., 13 табл., 3 додатки, 19 джерел.

Об'єкт дослідження – документація та погосподарська книга Гримайлівської селищної ради

Мета роботи – автоматизація системи ведення погосподарського обліку у селищних радах в умовах військового стану з метою полегшення опрацювання інформації: пошуку, редагування, введення нових даних, сортування, виведення за умовою (фільтрування) тощо

Метод дослідження – опис предметної області шляхом опрацювання та аналізу документів у селищній раді, проектування бази даних із використанням методів нормалізації; розробка і налагодження програми.

Отримані результати – спроектовано та розроблено інформаційну систему ведення погосподарського обліку у селищних радах в умовах військового стану з використанням технологій .Net

Ключові слова: Погосподарський облік, База даних, Субд, Автоматизація, Погосподарські книги, Інформаційна система, Персональний комп'ютер, Селищна рада.

ANNOTATION

Qualifying work for obtaining a bachelor's degree in the specialty 121 - Software engineering. Ivan Pulyuy Ternopil National Technical University, Faculty of Computer Information Systems and Software Engineering, Department of Software Engineering, Group KP-41, 2023. The explanatory note to the qualification work for obtaining a bachelor's degree contains: 61 pages, 26 figures, 13 tables, 3 appendices, 19 sources.

The object of the study is the documentation and household book of the Hrymailiv settlement council.

The purpose of the work is to automate the system of agricultural accounting in village councils under martial law in order to facilitate information processing: search, editing, entering new data, sorting, output by condition (filtering), etc.

The research method is a description of the subject area by processing and analyzing documents in the village council, designing a database using normalization methods; development and debugging of programs.

According to the obtained results, an information system for keeping agricultural records of village councils under martial law was developed and developed, using .Net technologies.

Keywords: Agricultural accounting, Database, Dbms, Automation, Household books, Information system, Personal computer, Village council.

ЗМІСТ

РЕФЕРАТ	4
ANNOTATION.....	5
ВСТУП.....	7
1 АНАЛІТИЧНА ЧАСТИНА	8
1.1 Опис об’єкта автоматизації.....	8
1.2 Аналіз подібних проектних рішень.....	10
2 ПРОЄКТНА ЧАСТИНА	14
2.1 Постановка задачі.....	14
2.2 Проектування бази даних	15
2.3 Проектування інформаційної системи	22
3 ПРАКТИЧНА ЧАСТИНА	28
3.1 Проектування інтерфейсу користувача	28
3.2 Опис програмних модулів	35
3.3 Опис результатів тестування	38
4 БЕЗПЕКА ЖИТТЄДІЯЛЬНОСТІ, ОСНОВИ ОХОРОНИ ПРАЦІ	47
4.1 Працездатність людини – оператора.....	47
4.2 Заходи пожежної безпеки підприємства, селищної ради.	49
ВИСНОВОК.....	52
ПЕРЕЛІК ПОСИЛАНЬ	53
ДОДАТКИ.....	54
Додаток А Лістинг програми «Автоматизований облік»	56
Додаток Б ER-діаграма бази даних «Duplom»	63
Додаток В Діаграма зав’язків бази даних «Duplom».....	64

ВСТУП

У сучасних умовах, коли держава стикається з різними викликами та потребує мобілізації своїх сил для забезпечення національної безпеки, військовий стан може стати необхідним інструментом для відповіді на загрози та виклики, які ставляться перед нею. Введення військового стану сприяє підвищенню готовності держави до військових дій та ефективному управлінню ресурсами. Одним із ключових аспектів підготовки та функціонування в умовах військового стану є належний погосподарський облік. Він відіграє важливу роль у забезпеченні ефективного управління ресурсами, контролю за фінансовими потоками та раціональному використанні активів.

Селищні ради як місцеві органи влади також потребують належного погосподарського обліку для забезпечення ефективного функціонування та надання необхідних послуг населенню. Умови військового стану вносять додаткові виклики та обмеження в роботу органів місцевого самоврядування, що вимагає спеціальних підходів та рішень у сфері погосподарського обліку.

Актуальність дипломної роботи зумовлена наступним фактором. Що впровадження програмного забезпечення в сільські ради дозволяє автоматизувати наступні процеси: складання і видачу даних з погосподарської книги, складання звітів про кількісні характеристики кожного погосподарства, формування довідок. Розроблювана інформаційна система повинна, дозволяти зберігати велику кількість відомостей в базі даних та економити робочий час за рахунок автоматизації процесів, а також повинна істотно підвищити ефективність роботи сільської ради. Таким чином, тема дипломної роботи не лише актуальна, але й перспективна.

1 АНАЛІТИЧНА ЧАСТИНА

1.1 Опис об'єкта автоматизації

На сьогодні, жодна сфера суспільної діяльності не обходиться без використання інформаційних технологій, оскільки кожна область потребує опрацювання великої кількості інформації та інформаційного обслуговування. Найкращим і універсальним засобом для обробки інформації вважається комп'ютер, який використовується як підсилювач інтелектуальних людських здібностей.

Жодне підприємство чи організація не обходяться у своїй роботі без застосування комп'ютерів, які з успіхом замінюють рутинну роботу, що виконувалася раніше вручну, підвищуючи її ефективність. Простота використання і зручний інтерфейс програм допомагають звичайним користувачам, які мало що знають про бази даних, легко з ними працювати, виконуючи свої посадові обов'язки. Зокрема це стосується і працівників органів місцевого самоврядування та виконавчої влади на місцях [1].

Програмне забезпечення обліку записів погосподарських книг розробляється відповідно до закону України №56 від 11 квітня 2016 р «Інструкції щодо ведення погосподарського обліку сільськими, селищними та міськими радами», в якому описуються основні напрямки політики щодо поліпшення ведення погосподарських книг.

Погосподарські книги є документами первинного адміністративного обліку населення, наявності у нього землі, худоби та іншого майна. Відомості з погосподарської книги є єдиним джерелом інформації про наявність і діяльність особистого погосподарства. Така інформація використовується органами виконавчої влади, офіційною статистикою і кредитними установами. Виписка, яка видається органом місцевого самоврядування з погосподарської книги, є підставою для державної реєстрації права власності громадянина на земельну ділянку і

документом, що підтверджує вирощування сільськогосподарської продукції на особистій ділянці (у тому числі і для отримання пільг по сплаті податку при її реалізації) [1]. Крім того, вона входить до списку документів громадянина-позичальника при подачі заяви про надання субсидій для відшкодування частини витрат на сплату відсотків по кредитах і позиках, а також при подачі заяви до кредитної установи про отримання кредитних ресурсів. Погосподарська книга складається з таких розділів:

- «Список членів домогосподарства»;
- «Худоба в приватній власності домогосподарства»;
- «Житловий будинок у приватній власності громадян»;
- «Земля в приватній власності та користуванні громадян»;
- «Сільськогосподарська техніка в приватній власності громадян».

У розділі «Список членів домогосподарства» голова домогосподарства записується першим, а потім всі інші члени домогосподарства, як присутні, так і відсутні. Усі три рядки, призначені для кожного члена домогосподарства, повинні містити повне написання прізвищ, імен і по батькових кожного члена, без перекручування та скорочення [2].

У розділі «Худоба в приватній власності домогосподарства» дані про поголів'я худоби, кролів, хутрових звірів і птиці записуються в присутності голови або дорослих членів домогосподарства два рази на рік, починаючи з 1 січня та 1 червня.

У розділі «Житловий будинок (квартира) в приватній власності громадян» міститься інформація про житловий будинок, а також усі його компоненти.

Житловий будинок — це будівля з поштовим номером, яка займає половину загальної площі та призначена для постійного проживання, і розташована на земельній ділянці з усіма допоміжними будівлями, спорудами та елементами благоустрою у визначених межах [2].

У підрозділі А розділу «Земля в приватній власності та користуванні громадян», за станом на 1 січня, вказується вся земельна площа різного цільового призначення, що перебуває у власності та користуванні, включаючи землю, яку орендують. У цей перелік входять землі, призначені для використання:

- будівництва;
- обслуговування житлового будинку і погосподарських будівель;
- ведення особистого підсобного погосподарства;
- сільськопогосподарське виробництво, включаючи паї та орендну землю.

У розділі «Сільськопогосподарська техніка в приватній власності громадян» зазначено кількість тракторів, вантажних автомобілів і комбайнів різних типів, марок і потужностей, які належать членам домогосподарства на правах власності на 1 січня, незалежно від їхнього технічного стану (працюючий, непрацюючий або в ремонті).

1.2 Аналіз подібних проєктних рішень

Розглянемо основні аналоги програми «Автоматизований облік» та їх функціональні можливості.

Система «Погосподарство» призначена для автоматизації ведення всіх показників погосподарської книги в сільських муніципальних утвореннях, сільських поселеннях, а також інших муніципальних утвореннях, до складу яких входять сільські населені пункти. Програмне забезпечення дає можливість підготовки та накопичення різнобічної інформації про приватні житлові погосподарства, їх власників, та належне їм майно; дозволяє отримати різні довідки і формувати необхідні форми звітності для інших організацій. Перевагою продукту є те, що програма виконана з урахуванням всіх вимог законодавства та відомчих актів,

сформованих на даний момент [3]. Програмне забезпечення «Погосподарство» поставляється на компакт-диску, який містить файли дистрибутиву для установки програми. Електронний ключ призначений для захисту від незаконного копіювання та подальшого використання. Керівництво користувача містить необхідну інформацію для успішної роботи з програмним забезпеченням. Серед недоліків виділимо те, що корисний термін використання програмного забезпечення, встановлений розробником, п'ять років. Вихідний код програми в загальний доступ не надається, тобто самостійне доопрацювання неможливе.

Програмне забезпечення «Реєстр погосподарства» призначене для роботи з даними, що стосуються погосподарств фізичних осіб. Реєстр забезпечує централізацію погосподарського обліку за рахунок застосування Інтернет-технологій. До переваг можна віднести той факт, що дані по сільських поселеннях вносяться у режимі реального доступу через Web-інтерфейс системи [3]. Недоліком є залежність від доступу в глобальну мережу (у віддалених населених пунктах доступ до мережі Інтернет недостатньо стабільний).

Програмне забезпечення «Попогосподарський облік» дозволяє вводити та зберігати інформацію про ферми сільської адміністрації. Особисте житлове приміщення або його частина, в якому проживають кілька людей і забезпечують себе всім, що потрібно для життя, ведучи спільне погосподарство, повністю або частково об'єднуючи та витрачаючи власні кошти, відоме як погосподарство. Члени погосподарства можуть бути пов'язані з родиною, одруженими або жити разом. Завдяки цій програмі можна працювати з даними в локальній мережі та виправляти структуру даних після збою комп'ютера або неправильного виходу з програми. Багато відгуків користувачів вказують на те, що програма часто зависає.

Автоматизована інформаційна система «Сільське адміністративне утворення» є територіально розподіленою трирівневою автоматизованою інформаційною системою, що функціонує в сільських населених пунктах – на рівні сільської адміністрації [4]. Ця система призначена для комплексної автоматизації

погосподарського обліку, формування, ведення та використання баз даних про населення, земельні ділянки, житловий фонд, рослинницькі, тваринницькі, рибпромислові і мисливські погосподарства, матеріально-технічну базу погосподарств, а також інформаційне обслуговування та надання послуг населенню. До переваг програми можна віднести те, що в її склад входять кілька додаткових підсистем: облік погосподарських книг, облік земельного фонду муніципального освіти, інформаційно-довідкове обслуговування громадян і організацій, облік договорів оренди. Основним недоліком цього продукту є складне налаштування системи.

Програмне забезпечення «Адміністрація населеного пункту» призначене для автоматизації діяльності адміністрацій муніципальних утворень сільських і міських поселень із ведення книг погосподарського обліку, статистичного обліку землі, нерухомості, худоби і техніки відповідно до встановлених законом норм; обліку муніципального майна, обліку землі за всіма видами права і платежів, що надходять від населення, у тому числі орендної плати. Перевагами програми є облік руху по особовому рахунку, що відбувається у процесі ведення книги погосподарського обліку та дозволяє формувати регламентовану звітність (лист особового рахунку, відомості про житловий фонд – форма №1, відомості про худобу – форма № 14 і багато іншого) [4]. До недоліків відноситься те, що продукт не є самостійною програмою і призначений для використання з компонентом «Бухгалтерський облік» системи програм «1С: Підприємство», що веде до додаткових витрат на придбання та налагодження.

Інформаційна система «Поселення» розроблена для спрощення процесів накопичення, опрацювання, зберігання та аналізу даних, які використовуються муніципальними утвореннями, а саме поселеннями та муніципальними районами. Перевагами системи є функції з надання делегованих повноважень: облік даних паспортно-візової служби, військовий облік, реєстрація актів громадянського стану.

Недоліками – платний доступ і надлишковий функціонал, що веде до додаткових витрат на навчання персоналу.

Інформаційна система «Розумне село» не може бути розширена. Її мета полягає в тому, щоб автоматизувати ведення погосподарських книг, статистичного обліку землі, нерухомості, транспорту та свійських тварин відповідно до встановлених законом норм обліку громадянського майна для сільських і селищних рад.

Окремий розділ системи призначений для автоматизації процедур реєстрації народжень, розлучень, шлюбів громадян (відомий як «розділ РАГС»), а також процедур міграції громадян. Сільські ради можуть планувати свою роботу, щоб скоротити час, необхідний для виписки та видачі довідок громадянам, а також гарантувати, що потрібна інформація (звітність, реєстри звернень громадян, тощо) надходить надійно та вчасно. Впровадження нових інформаційних технологій покращує координацію, керованість і ефективність діяльності сільської ради. На даний момент система «Розумне село» є унікальною в Україні та не має подібних у інших країнах [4].

У результаті проведеного аналізу розглянуто та виявлено переваги і недоліки існуючих систем обліку і ведення погосподарських книг селищних адміністрацій. У дипломному проєкті буде розроблено власне програмне забезпечення, що реалізує весь потрібний функціонал, але не має перерахованих вище недоліків.

2 ПРОЄКТНА ЧАСТИНА

2.1 Постановка задачі

Погосподарський облік, як різновид бухгалтерського обліку, призначений для збору та систематизації інформації, яку збирають сільські, селищні та міські ради на своїх територіях. Ці дані необхідні для проведення державних статистичних спостережень.

Внесення змін до книг погосподарського обліку селищної ради – є процесом безперервним. Постійно відбуваються зміни, які потрібно зображати документально. Оскільки, ведення книг документально є процесом трудомістким і багатозатратним, доцільно створити програмне забезпечення для автоматизації ведення погосподарських книг [6].

Інформаційна система буде складатись з двох частин. Перша частина – це серверна база даних для зберігання інформації про погосподарства, яка буде забезпечувати надійне зберігання і доступ до інформації. Друга частина – клієнтський додаток, що дозволяє автоматизувати багато функцій, пов'язаних з опрацюванням даних про погосподарства. Система повинна зберігати та опрацьовувати інформацію про погосподарства, учасників погосподарства, ресурси погосподарства, технічне забезпечення погосподарства і т. д. Система розроблена для кінцевого користувача, який не має надзвичайної обчислювальної кваліфікації. Таким чином, автоматизована інформаційна система повинна мати простий інтерфейс, який легко використовувати.

Основними функціями програми повинні бути:

- можливість отримання зведених звітів, які базуються на даних, які є комп'ютерним аналогом книг погосподарського обліку;
- ведення обліку населення, житла, яке належить громадянам на правах власності, наявність худоби, сільськопогосподарської техніки та земельних наділів;

- формування різного виду документів, довідок та аналітичних таблиць;
- можливість використовувати швидкий і ефективний пошук, щоб отримати повну та різнобічну інформацію про кожного жителя за будь-який період.

2.2 Проектування бази даних

Відповідно до Конституції в Україні визнано та гарантовано місцеве самоврядування. Право на добровільне об'єднання кількох сіл, селищ і міст у одну територіальну громаду відоме як місцеве самоврядування в Україні.

Предметною областю для дослідження є орган місцевого самоврядування громади. Секретар ради організовує людей та контролює створення інструктивних матеріалів з діловодства, вдосконалення форм і методів опрацювання документів, впровадження комплексної автоматизації діловодства.

Голова села веде погосподарський облік села. Ці дані є своєрідним «кошиком» корисної інформації для самого голови села, а також необхідні для підготовки звітів до органів статистики [7].

Проведення аналізу предметної області є основним завданням для подальшого проектування бази даних. Опис основних компонентів бази даних представлено в таблиці 2.1.

Таблиця 2.1 – Опис основних компонентів бази даних «Diplom»

Ім'я сутності	Опис	Псевдоніми	Особливості використання
1	2	3	4
Farm	Інформація про всі погосподарства	Farm	Використовують для перегляду всіх погосподарств занесених в базу даних
PeopleInfo	Інформація про членів погосподарства	PeopleInfo	Зберігається інформації про членів погосподарства

Livestock	Довідник типів свійських тварин	Livestock	Зберігається інформації про свійських тварин які належать погосподарству
-----------	---------------------------------	-----------	--

Продовження таблиці 2.1

1	2	3	4
Improvements	Інформація про будинок	Improvements	Зберігається інформації про характеристику будинку
FarmChar	характеристика погосподарства	FarmChar	Зберігається інформації про наявність в погосподарствах покращених умов
FarmEarth	Земельні характеристики погосподарства	FarmEarth	Зберігається інформації про земляну ділянку на території погосподарства
FarmTech	Інформація про техніку погосподарства	FarmTech	Зберігається інформації про техніку в користуванні
Users	Список користувачів	Users	Кожен користувач заходить під своїми даними

Для структурованого зберігання інформації про погосподарства було прийнято рішення використовувати реляційну базу даних – MS SQL Server.

Опис атрибутів сутностей бази даних «Duplom» показано в таблиці 2.2.

Таблиця 2.2 – Опис атрибутів сутностей бази даних «Duplom»

Назва сутності	Атрибут	Опис	Домен	Значення за замовчуванням	Псевдонім	Значення Null
PeopleInfo	IdPeople	Код члена погосподарства	Ціле число	-	IdPeople	Ні
	LastName	Прізвище	Стрічка	-	LastName	Ні
	FirstName	Ім'я	Стрічка	-	FirstName	Ні
	MiddleName	По батькові	Стрічка	-	MiddleName	Ні
	Nationality	Національність	Стрічка	-	Nationality	Ні
	IdStatus	Статус в сім'ї	Ціле число	-	IdStatus	Ні
	IdEducation	Освіта	Ціле число	-	IdEducation	Ні
	BirthDay	Дата народження	Дата	-	BirthDay	Ні
IdFarm	Ідентифікатор погосподарства	Ціле число	-	IdFarm	Ні	

Farm	IdFarm	Код погосподарства	Ціле число	-	IdFarm	Hi
------	--------	--------------------	------------	---	--------	----

Продовження таблиці 2.2

1	2	3	4	5	6	7
	Num	Особистий рахунок	Ціле число	-	Num	Hi
	IdAdress	Адреса	Ціле число	-	IdAdress	Hi
Livestoc	IdLivestock	Унікальний ідентифікатор	Ціле число	-	IdLivestock	Hi
	IdLivestockType	Тип свійських тварин	Ціле число	-	IdLivestockType	Hi
	DateS	Контрольна дата	Дата	-	DateS	Hi
	Count	Кількість свійських тварин	Ціле число	-	Count	Hi
	IdFarm	Ідентифікатор погосподарства	Ціле число	-	IdFarm	Hi
	Improvements	IdImprovement	Код будинку	Ціле число	-	IdImprovement
ImprovementType		Вид будинку	Ціле число	-	ImprovementType	Hi
DateConstruction		Дата будівництва	Дата	-	DateConstruction	Hi
WallMaterial		Матеріал стін	Стрічка	-	WallMaterial	Hi
RoofMaterial		Матеріал даху	Стрічка	-	RoofMaterial	Hi
FarmBuildings		Інші будівлі	Стрічка	-	FarmBuildings	Hi
idFarm		Ідентифікатор погосподарства	Ціле число	-	idFarm	Hi
FarmEarth	IdFarmEarth	Унікальний ідентифікатор	Ціле число	-	IdFarmEarth	Hi
	DateE	Контрольна дата	Дата	-	DateE	Hi
	BildEarth	Площа під будівлями	Ціле число	-	BildEarth	Hi
	PosivEarth	Площа під посів	Ціле число	-	PosivEarth	Hi
FarmTech	IdFarmTech	Унікальний ідентифікатор	Ціле число	-	IdFarmTech	Hi
	IdFarmTechType	Тип техніки	Ціле число	-	IdFarmTechType	Hi
	Count	Кількість	Ціле число	-	Count	Hi
	Year	Контрольна дата	Дата	-	Year	Hi
	IdFarm	Ідентифікатор	Ціле число	-	IdFarm	Hi

		погосподарст ва				
--	--	--------------------	--	--	--	--

Продовження таблиці 2.2

1	2	3	4	5	6	7
FarmChar	IdFarmChar	Унікальний ідентифікатор	Ціле число	-	IdFarmChar	Hi
	TotalSpace	Загальна площа	Ціле число	-	TotalSpace	Hi
	LivingSpace	Житлова площа	Ціле число	-	LivingSpace	Hi
	Flats	Кількість кімнат	Ціле число	-	Flats	Hi
	Price	Ціна погосподарст ва	Ціле число	-	Price	Hi
	Sewereg	Каналізація	Стрічка		Sewerage	Hi
	Aqueduct	Водопостачан ня	Стрічка		Aqueduct	Hi
	Gas	Газ	Стрічка	-	Gas	Hi
	HotWater	Гаряча вода	Стрічка	-	HotWater	Hi
	IdFarm	Ідентифікатор погосподарст ва	Ціле число	-	IdFarm	Hi
	DateF	Контрольна дата	Дата	-	DateF	Hi

У процесі створення бази даних слід створити її ER-діаграму. Щоб побудувати ER-діаграму, спочатку потрібно визначити основні ключові сутності, типи зв'язків і їх важливість. Кожна сутність має основний ключ. Зв'язки між сутностями були визначені шляхом аналізу предметної області (див. Табл. 2.3).

Таблиця 2.3 – Опис взаємозв'язків між сутностями бази даних «Diplom»

Імя сутності	Назва зв'язку	Імя сутності	Кардинальність
Farm	Має	FarmChar	1:N
Farm	Має	FarmEarth	1:N
Farm	Має	FarmTech	1:N
Farm	Має	Livestock	1:N
Farm	Має	PeopleInfo	1:1
Farm	Має	Address	N:1

Надалі потрібно обрати модель для проектування бази даних. Для побудови логічної схеми даних ми використаємо реляційну модель даних. Створення реляційної схеми, визначення структури відношень бази даних і вирішення багатьох інших завдань є частиною логічного проектування реляційної моделі даних.

Правила нормалізації використовуються для перевірки правильності логічних моделей, щоб переконатися, що модель даних має структурну узгодженість, логічну цілісність і мінімальну збитковість [8]. На даний момент набір відношень буде визначено на основі структури логічної моделі даних.

На цьому етапі необхідно створити відносини на основі зв'язків і сутностей. Механізм первинних та зовнішніх ключів використовується для моделювання зв'язків між сутностями. Кожна сутність має відношення, і кожна характеристика сутності є характеристикою відповідного відношення. Усі характеристики батьківської сутності включені до кожного відношення, яке відповідає дочірній сутності. Ці характеристики складають основний ключ основної сутності. Цей набір атрибутів стає зовнішнім ключом для дочірньої сутності. Після цього необхідно нормалізувати відносини. Процедура нормалізації гарантує коректність логічної моделі даних.

Нормалізація бази даних включає виявлення функціональних залежностей і усунення надмірності даних. За рахунок уникнення зайвого дублювання наборів даних, усунення надмірності даних гарантує компактність наборів даних. Це також запобігає аномаліям, пов'язаним із внесенням, зняттям і оновленням кортежів після фізичної реалізації бази даних. Термін «функціональна залежність» стосується зв'язку між характеристиками в одному відношенні та їхніми значеннями в іншому.

Хоча правила нормалізації нечисленні та прості, вони досить суворі. Кожне правило описує наступний рівень відповідності вимогам теорії реляційних баз даних і різні ступені нормалізації, коли застосовуються відносини.

Вони мають різні рівні нормалізації: перша нормальна форма, друга нормальна форма, третя нормальна форма, четверта нормальна форма і п'ята нормальна форма.

Досі жодна з реляційних СКБД не підтримує всі п'ять нормальних форм. Це результат суворих вимог до продуктивності [9]. Суть справи полягає в тому, що повністю нормалізована БД для виконання запитів повинна включати таку кількість таблиць, що система не зможе задовольнити потреби користувачів. Отже, на практиці використовуються лише перші три рівня нормалізації: перша нормальна форма, друга нормальна форма і третя нормальна форма.

У першій нормальній формі кожен рядок і кожен стовпець мають одне значення. Відношення першої нормальної форми має повну функціональну залежність від первинного ключа для кожного атрибуту. Це називається другою нормальною формою. Третя нормальна форма — це відношення, яке знаходиться в другій і першій нормальних формах і не має атрибутів, які були б у транзитивній функціональній залежності від первинного ключа [9].

Проаналізувавши відношення спроектованої бази даних, можна стверджувати, що вони будуть нормалізованими до третьої нормальної форми.

Завдяки аналізу предметної області ми створимо структуру таблиць за допомогою запитів SQL. Лістинг створення наведено нижче.

Лістинг 2.1 SQL-запит для створення таблиці «PeopleInfo»

```
CREATE TABLE [dbo].[PeopleInfo](
  [IdPeople] [int] IDENTITY(1,1) NOT NULL PRIMARY KEY,
  [LastName] [varchar](100) NOT NULL,
  [MiddleName] [varchar](100) NOT NULL,
  [Nationality] [varchar](100) NOT NULL,
  [FirstName] [varchar](100) NOT NULL,
  [IdStatus] [int] NOT NULL,
  [IdEducation] [int] NOT NULL,
  [Birthday] [date] NOT NULL,
  [IdFarm] [int] NOT NULL,
  [IdSex] [int] NOT NULL)
```

Лістинг 2.3 SQL-запит для створення таблиці «Farm»

```
CREATE TABLE [dbo].[Farm](
  [IdFarm] [int] IDENTITY(1,1) NOT NULL PRIMARY KEY,
```

```
[Num] [int] NOT NULL,  
[IdAddress] [int] NOT NULL)
```

Лістинг 2.2 SQL-запит для створення таблиці «FarmChar»

```
CREATE TABLE [dbo].[FarmChar](  
[IdFarmChar] [int] IDENTITY(1,1) NOT NULL PRIMARY KEY,  
[TotalSpace] [int] NOT NULL,  
[LivingSpace] [int] NOT NULL,  
[Flats] [int] NOT NULL,  
[Price] [int] NOT NULL,  
[Sewerage] [nvarchar](100) NOT NULL,  
[Aqueduct] [nvarchar](100) NOT NULL,  
[Gas] [nvarchar](100) NOT NULL,  
[HotWater] [nvarchar](100) NOT NULL,  
[IdFarm] [int] NOT NULL,  
[DateF] [date] NOT NULL)
```

Лістинг 2.5 SQL-запит для створення таблиці «FarmTech»

```
CREATE TABLE [dbo].[FarmTech](  
[IdFarmTech] [int] NOT NULL PRIMARY KEY,  
[IdFarmTechType] [int] NULL,  
[Count] [int] NULL,  
[Year] [date] NULL,  
[IdFarm] [int] NULL)
```

Лістинг 2.6 SQL-запит для створення таблиці «Livestock»

```
CREATE TABLE [dbo].[Livestock](  
[IdLivestock] [int] IDENTITY(1,1) NOT NULL PRIMARY KEY,  
[IdLivestockType] [int] NOT NULL,  
[DateS] [date] NOT NULL,  
[Count] [int] NOT NULL,  
[IdFarm] [int] NOT NULL)
```

Лістинг 2.7 SQL-запит для створення таблиці «Improvements»

```
CREATE TABLE [dbo].[Improvements](  
[IdImprovement] [int] IDENTITY(1,1) NOT NULL PRIMARY KEY,  
[ImprovementType] [nvarchar](100) NOT NULL,  
[DateConstruction] [date] NOT NULL,  
[WallMaterial] [nvarchar](100) NOT NULL,  
[RoofMaterial] [nvarchar](100) NOT NULL,
```

```
[FarmBuildings] [nvarchar](100) NOT NULL,  
[idFarm] [int] NOT NULL)
```

Лістинг 2.4 SQL-запит для створення таблиці «FarmEarth»

```
CREATE TABLE [dbo].[FarmEarth](  
[IdFarmEarth] [int] IDENTITY(1,1) NOT NULL PRIMARY KEY,  
[DateE] [date] NOT NULL,  
[BildEarth] [int] NOT NULL,  
[PosivEarth] [int] NOT NULL,  
[IdFarm] [int] NOT NULL)
```

Лістинг 2.8 SQL-запит для створення таблиці «User»

```
create table [dbo].[User](  
[Userid] [int] IDENTITY(1,1) not null PRIMARY KEY,  
[LoginName] [varchar](100) not null,  
[Password] [varchar](100) not null,  
[FirstName] [varchar](100) not null,  
[LastName] [varchar](100) not null,  
[Position] [varchar](100) not null)
```

2.3 Проектування інформаційної системи

У проектуванні інформаційної системи розробнику допомагає уніфікована мова моделювання – UML. Основною метою використання мови UML є взаємодія розробників. Моделювання процесу чи системи, як правило, відбувається з метою реалізації у вигляді програмного коду. Однак розмова про деталі моделі за допомогою мови програмування робить розуміння базових понять надзвичайно складним через акцент на деталях реалізації. Крім того, через брак точних означень люди плутаються, коли говорять на природній мові. Таким чином, коли потрібна точність, мову моделювання UML доцільно використовувати [10].

Для проектування інформаційної системи були побудовані такі діаграми:

- Use case (діаграма використання);
- Class diagram (діаграма класів);
- Sequence diagram (діаграма послідовності).

Граф спеціального вигляду, який використовується, для зображення конкретних варіантів використання, акторів, можливих інтерфейсів і зв'язків між ними називається діаграмою використання. Таким чином, можна скласти окремі елементи діаграми в прямокутник, який відобразить проєктовану систему в цілому. Кожен варіант використання визначає послідовність дій, які система повинна виконати в взаємодії з відповідним актором [11].

Для діаграми використання в якості користувачів програмного забезпечення можна виділити 2 типи:

- оператор;
- адміністратор.

Оператор має можливість переглядати дані про погосподарства, додавати нові, редагувати вже наявні, і т.д.

Адміністратор має можливість редагувати інформацію про користувачів і переглядати статистику роботи. Розглянемо декілька варіантів використання програмного забезпечення, а саме «Робота з даними про погосподарства», «Управління користувачами», «Вхід в АІС» Діаграма використання програмного забезпечення наведена на рисунку 2.1.

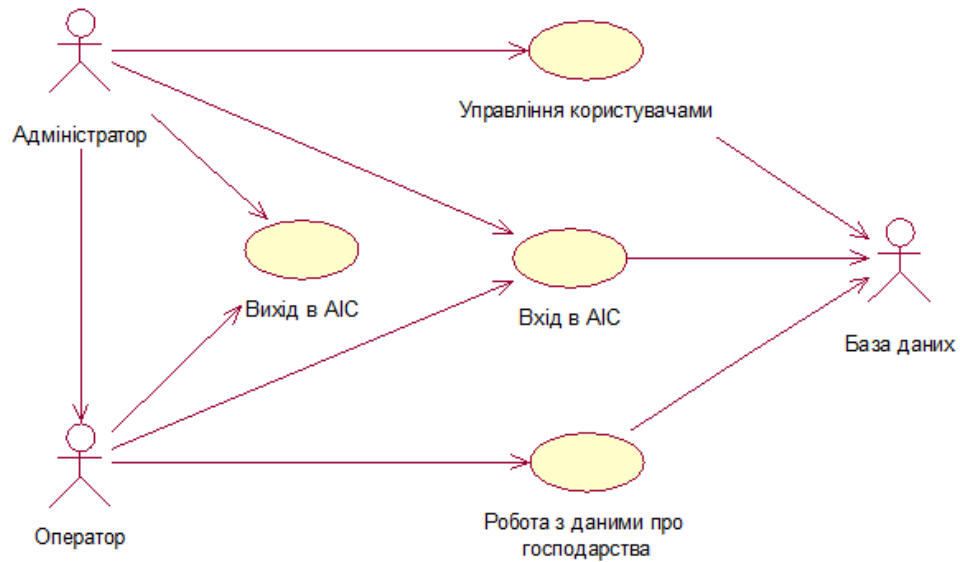


Рисунок 2.1 – Діаграма варіантів використання програмного забезпечення

Варіант «Робота з даними про погосподарства» містить функції: додавання відомостей про погосподарство, редагування відомостей про погосподарство, видалення відомостей про погосподарство. Діаграма варіанта використання «Робота з даними про погосподарства» наведена на рисунку 2.2.

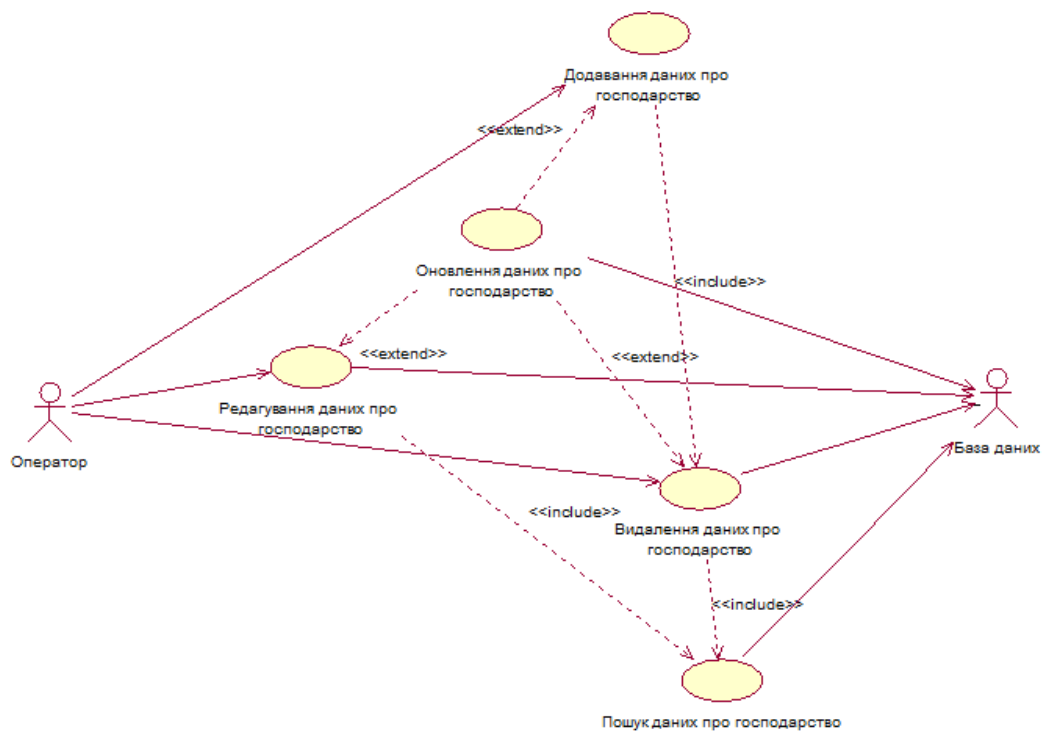


Рисунок 2.2 – Діаграма «Робота з даними про погосподарства»

Варіант «Управління користувачами» містить функції: додавання користувача, редагування користувача, видалення користувача, пошук користувача. Діаграма варіанта використання «Управління користувачами» наведена на рисунку 2.3.

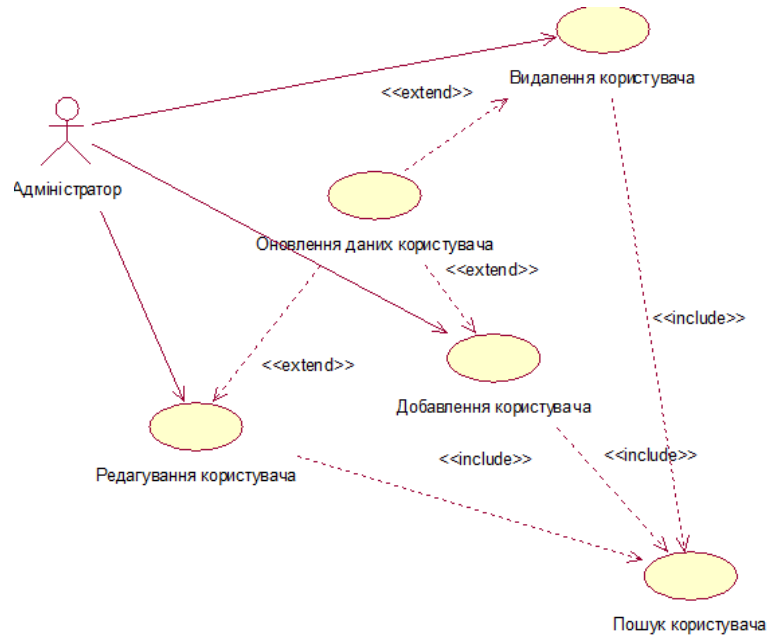


Рисунок 2.3 – Діаграма «Управління користувачами»

Варіант «Вхід в ПО АІС» містить функції: авторизації, аутентифікації користувача, ідентифікації користувача. Діаграма варіанта використання «Вхід в АІС» наведена на рисунку 2.4.

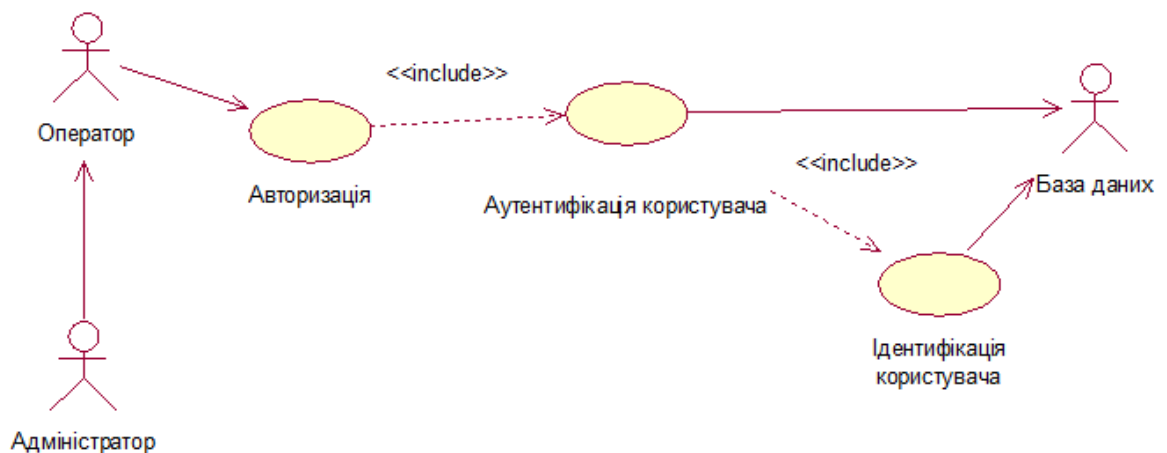


Рисунок 2.4 – Діаграма «Вхід в АІС»

Статичне представлення структури моделі можна знайти за допомогою діаграми класів. Вона нагадує символи класів, а також зв'язки між ними. Іншими словами, діаграма класу може показувати лише імена класів, а також їхні відповідні атрибути, або імена класів, атрибути та операції [11].

Діаграма класів складається з багатьох різних елементів, які в поєднанні демонструють декларативні знання, пов'язані з предметом. Базові поняття мови UML, такі як класи, інтерфейси та відносини між ними та їхніми складовими компонентами, використовуються для інтерпретації цих знань. Діаграма класів для програмного забезпечення наведена на рисунку 2.5.

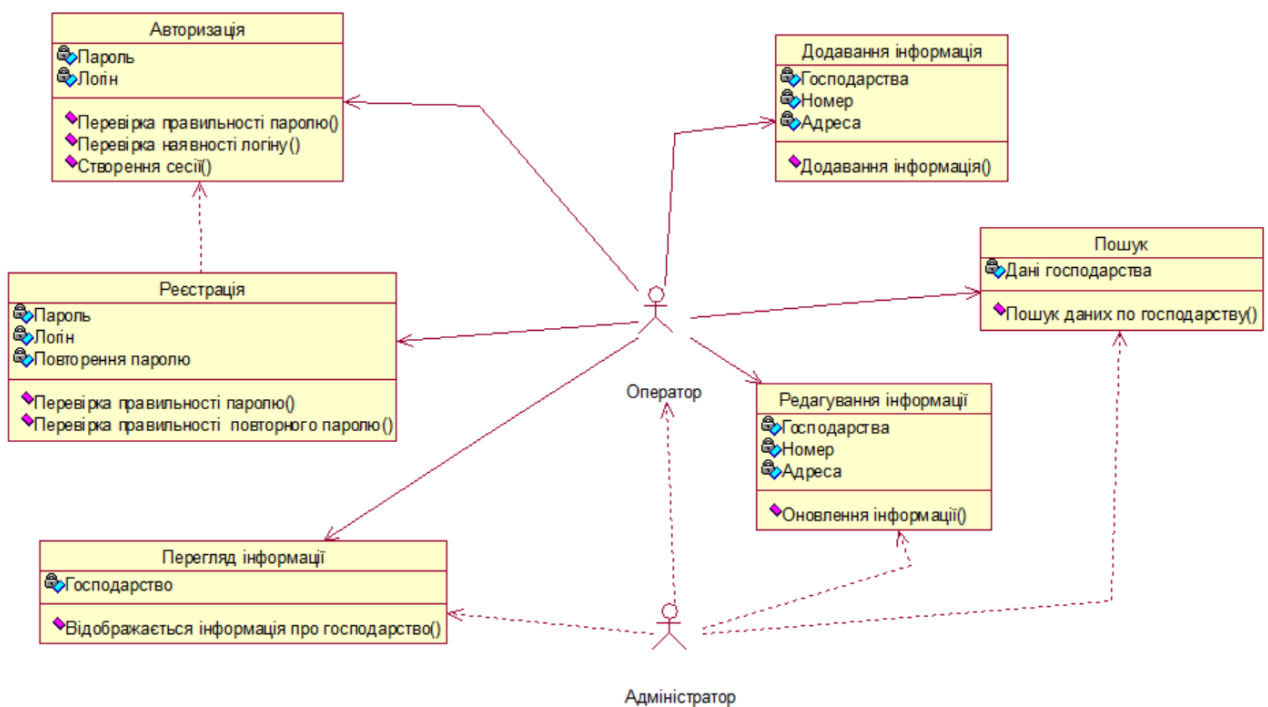


Рисунок 2.5 – Діаграма класів для програмного забезпечення

Діаграма послідовності — це графічне представлення поведінки об'єктів у сценаріях. Кожен об'єкт сценарію побудований відповідно до своєї лінії життя, яка відображає хід подій. Упорядкована послідовність подій представляє взаємодію між

лініями життя в діаграмі послідовності. це самий різноманітний варіант діаграми впливів [14]. Через те, що лінії життя легко розмістити та з'єднати на діаграмі, моделювання іноді починають за допомогою комунікаційної діаграми. На діаграмі вона представлена вертикальною пунктирною лінією, на верхній частині якої в прямокутнику зображується назва об'єкта між його створенням і руйнуванням. Діаграму послідовності для варіанту використання «Додавання відомостей про погосподарство» наведена на рисунку 2.6



Рисунок 2.6 – Діаграма послідовності «Додавання відомостей про погосподарство»

У процесі проектування UML-діаграм використовувався IBM Rational Rose Enterprise Edition. UML-діаграми легко сприймаються та легко генеруються комп'ютерами, а також вони об'єднали в собі найкращі сучасні технічні прийоми моделювання та розробки програмного забезпечення [15].

3 ПРАКТИЧНА ЧАСТИНА

3.1 Проектування інтерфейсу користувача

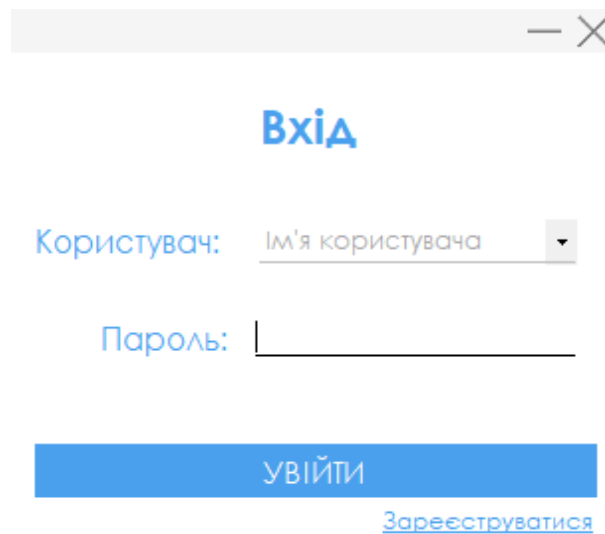
Користувацький інтерфейс програми — це сукупність елементів, що дозволяють користувачеві програми керувати її роботою і отримувати необхідні результати.

Метою дипломного проєкту є розробка інформаційної системи. Розроблюване програмне забезпечення повинне відповідати міжнародній класифікації вимог до користувацького інтерфейсу, а саме:

- відповідність до завдань, що вирішуються користувачем;
- легкістю в застосуванні;
- керованістю;
- відповідністю до очікувань користувача;
- стійкістю до помилок;
- адаптованістю;
- легкістю в вивченні.

Довільне програмне забезпечення орієнтоване на кінцевого користувача, що, зазвичай, не володіє високою кваліфікацією в області обчислювальної техніки. Тому автоматизована інформаційна система повинна володіти простим і зручним інтерфейсом.

Перед початком роботи з програмою користувача зустрічає форма авторизації (див. Рис 3.1).



Вхід

Користувач:

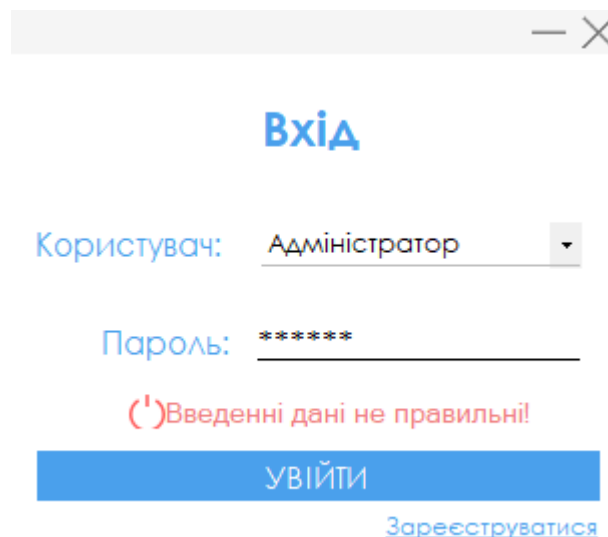
Пароль:

УВІЙТИ

[Зареєструватися](#)

Рисунок 3.1 – Вікно авторизації

У випадку, якщо користувач введе неправильні дані авторизації, з'явиться повідомлення про помилку у відповідному полі введення або повідомлення про відсутність таких даних в базі (див. Рис 3.2).



Вхід

Користувач:

Пароль:

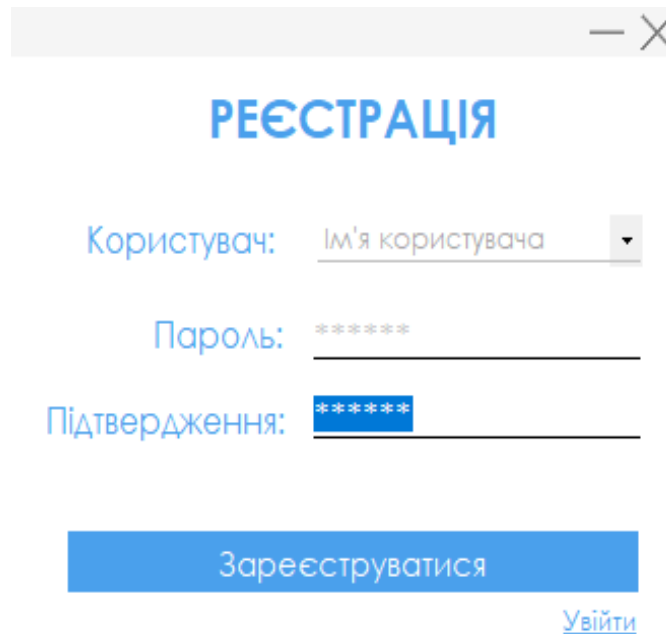
(!) Введенні дані не правильні!

УВІЙТИ

[Зареєструватися](#)

Рисунок 3.2 – Вікно авторизації з повідомленням про помилку

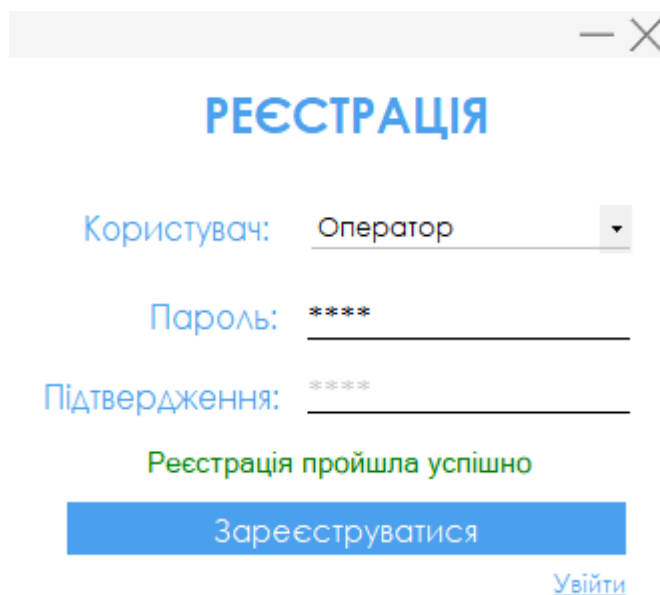
У нових користувачів є можливість створення власного профіля. Для цього потрібно натиснути кнопку «Зареєструватися» яка відкриває вікно для реєстрації (див. Рис 3.3).



The image shows a registration window with a title bar containing a close button (X). The main heading is 'РЕЄСТРАЦІЯ' in blue. Below it are three input fields: 'Користувач:' with a dropdown menu showing 'Ім'я користувача', 'Пароль:' with a masked password '*****', and 'Підтвердження:' with a masked password '*****'. At the bottom, there is a large blue button labeled 'Зареєструватися' and a smaller blue link labeled 'Увійти'.

Рисунок 3.3 – Вікно реєстрації

Щоб провести реєстрацію, у відкритому вікні потрібно заповнити всі поля і натиснути кнопку «Зареєструватися». У разі успішної реєстрації користувач отримає повідомлення про те що реєстрація даних закінчена (див. Рис 3.4).



РЕЄСТРАЦІЯ

Користувач:

Пароль:

Підтвердження:

Реєстрація пройшла успішно

[Зареєструватися](#)

[Увійти](#)

Рисунок 3.4 – Вікно реєстрації з повідомленням про успішну реєстрацію

Після успішної реєстрації та авторизації користувача зустрічає вікно завантаження програми на якому показані дані користувача, якщо ввійшла зареєстрована людина або пуста поле якщо увійшов новий користувач (див. Рис 3.5).

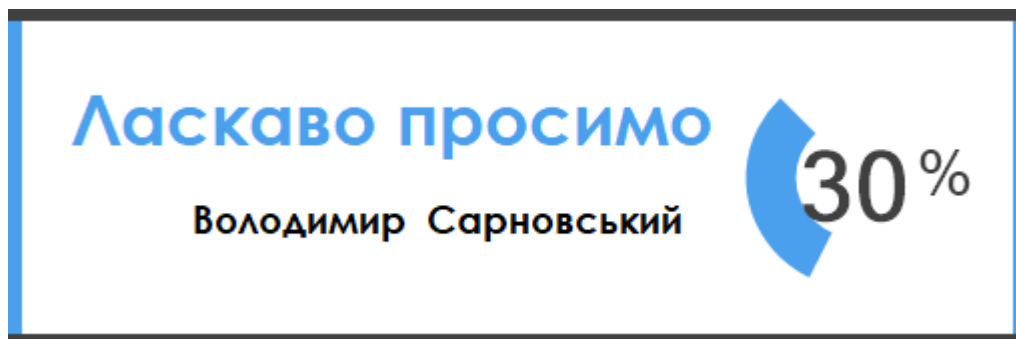


Рисунок 3.5 – Вікно завантаження програми

Після завантаження користувача вітає головне вікно програми (див. Рис 3.6).

The screenshot shows the main window of a program. On the left is a blue sidebar menu with the following items: 'Перегляд даних' (Data View), 'Оновлення даних' (Update Data), 'Пошук даних' (Search Data), 'Документи' (Documents), and 'Статистика даних' (Data Statistics). The top header area includes the user profile 'Володимир Сарновський, Адміністратор', the title 'Список господарств с. Кут-Товсте', the date '20 червня 2023 р.', and the time '22:32:09'. A 'Загалом:' (Summary) box shows 'Господарства' (Households) with a count of 4. Below the header is a table with the following data:

Особистий рахунок №	ZIP-код	Будинок №	Країна	Регіон	Область
11	43445	4	Україна	Гусятинський	Тернопільська
223	43445	4	Україна	Гусятинський	Тернопільська
231	35466	3	Україна	Гусятинський	Тернопільська
514	45343	6	Україна	Гусятинський	Тернопільська

Рисунок 3.6 – Головне вікно програми

У головному вікні розташовано список всіх погосподарств, меню програми, а також інформація про профіль користувача. Головне меню програми виконує наступні функції:

- додавання нових даних про погосподарства;
- перегляд даних про погосподарства;
- пошук необхідних даних за різними критеріями;
- створення необхідної документації;

Для перегляду даних про погосподарство потрібно вибрати необхідне погосподарство із списку та натиснути ПКМ (див. Рис 3.7) в контекстному меню вибрати «Переглянути дані», після чого буде відкрито вікно з інформацією про вибране погосподарство (див. Рис 3.8).

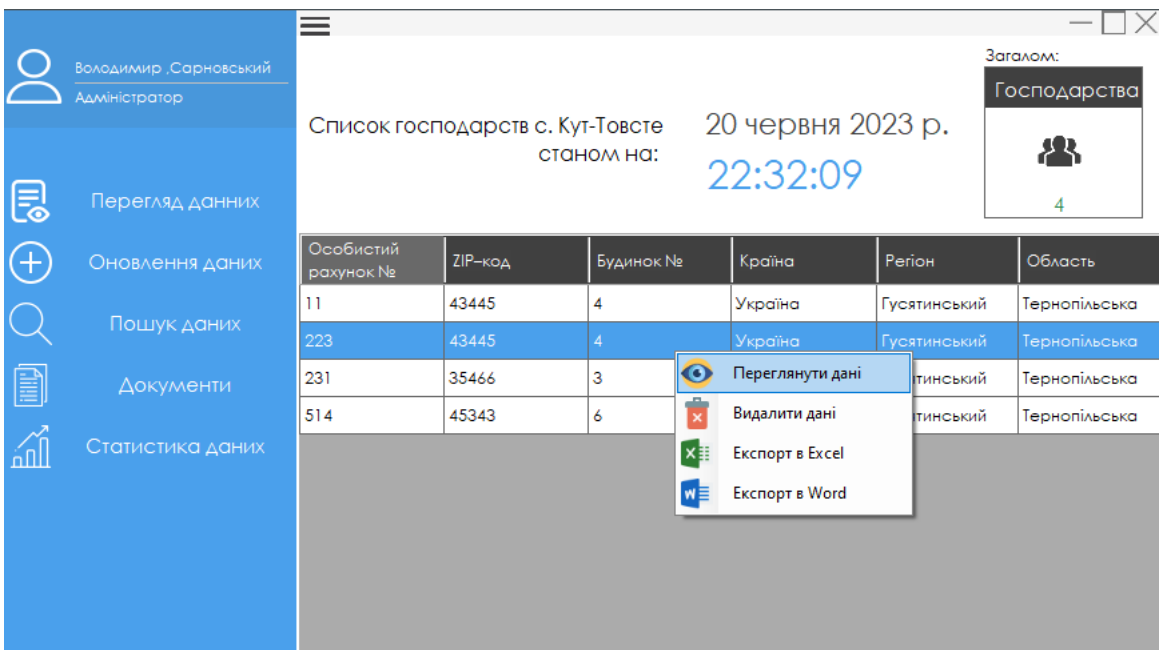


Рисунок 3.7 – Контекстне меню

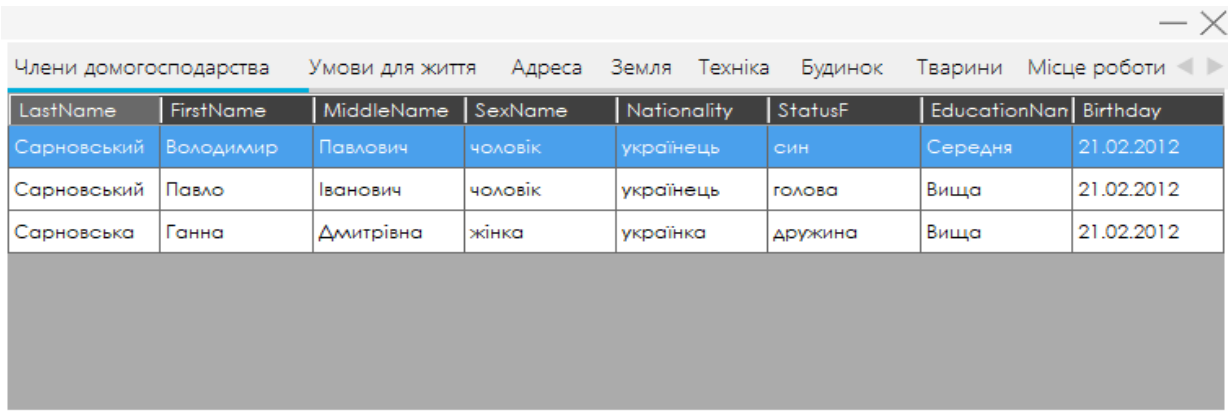


Рисунок 3.8 – Вікно перегляду даних

Що додати або оновити дані потрібно вибрати «Оновлення даних» з бокового меню, після чого буде відкрита форма заповнення даних (див. Рис 3.9). Після заповнення всіх полів потрібно натиснути «Додати» у разі успішного заповнення з’явиться відповідне повідомлення.

Володимир Сарновський
Адміністратор

Адреса Техніка Будинок **Умови для життя** Члени домогосподарства Місце роботи

Господарство: _____ Водопостачання

Загальна площа: _____ Газ

Житлова площа: _____ Каналізація

Дата перепису: _____ Гаряча вода

Кімнати: 0 _____

Додати

Рисунок 3.9 – Вікно додавання та оновлення даних

Для зручнішого додавання даних, пересування по вкладках можна використовувати стрілки розташовані внизу вікна.

Щоб пошук потрібних даних був легким і займав небагато часу в програмному засобі організовано таку можливість (див. Рис 3.10). Для пошуку необхідно заповнити вибрані критерії та натиснути «Пошук», в відкритому вікні буде відображений результат пошуку (див. Рис 3.11).

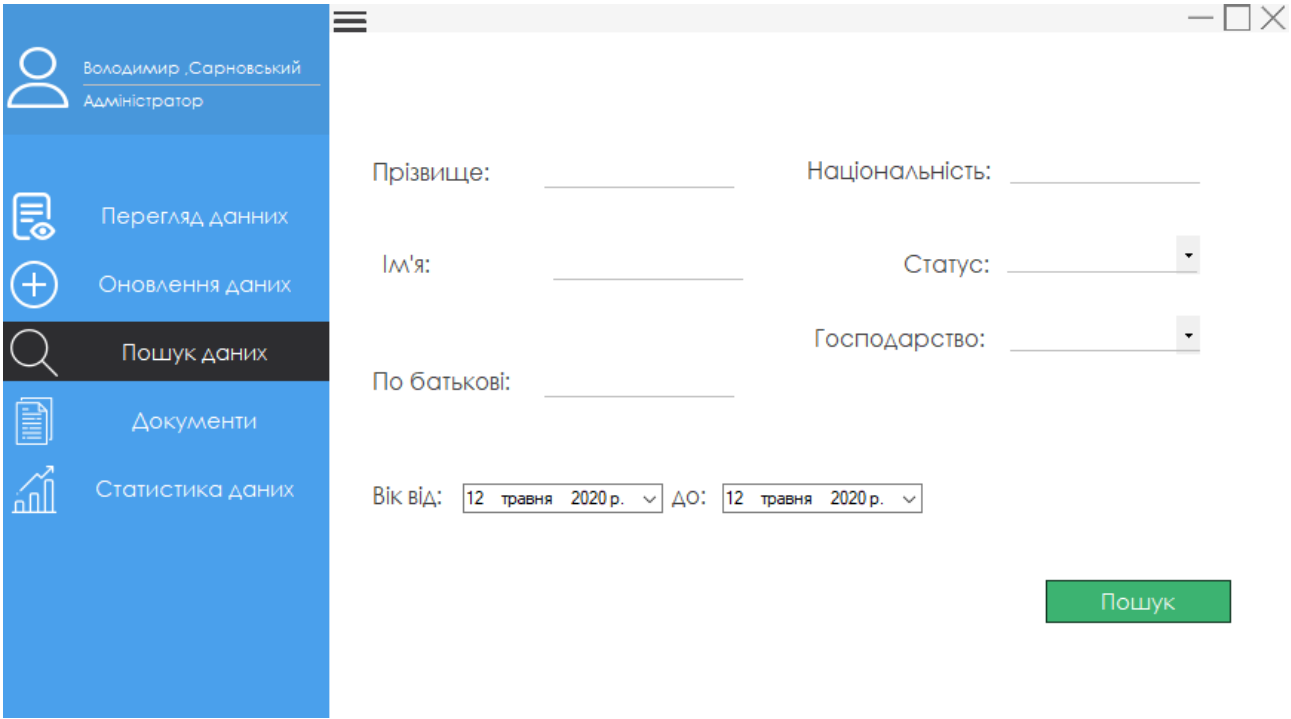


Рисунок 3.10 – Вікно пошуку даних

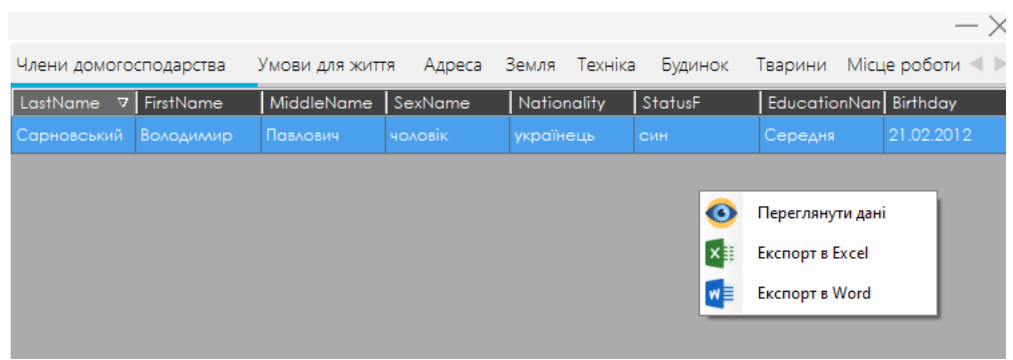


Рисунок 3.11 – Результат пошуку даних

Результати пошуку можна експортувати в Microsoft Word або Excel залежно від бажання користувача. Для цього потрібно натиснути ПКМ на таблицю результатів і вибрати тип експорту, який потрібен користувачу.

Для отримання виписок з погосподарської книги, довідок та іншого типу документації, потрібно скористатися меню «Документи» (див. Рис 3.12).

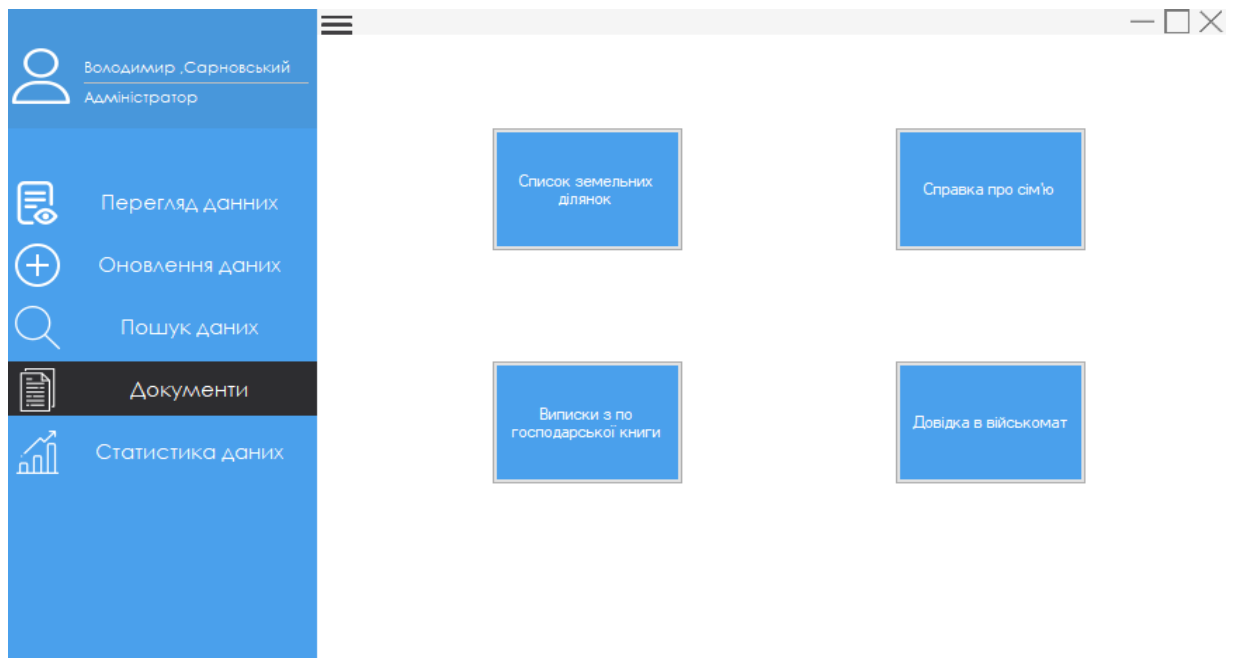


Рисунок 3.12 – Вікно списку документів

Користувач має можливість перегляду або редагування даних свого профілю, для цього потрібно натиснути на іконку в лівому верхньому куті (див. Рис 3.13). Для редагування даних користувача а також і його паролю потрібно натиснути «Редагувати» і в відкритому вікні змінити дані (див. Рис 3.13).

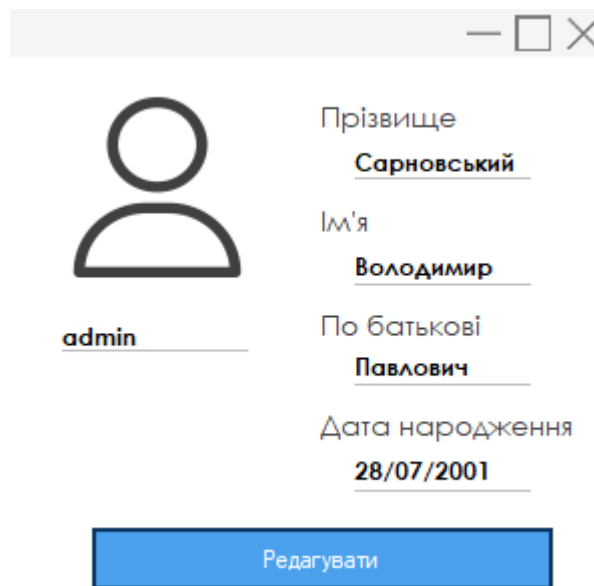


Рисунок 3.12 – Профіль користувача

— □ ×

Прізвище **Сарновський**

Ім'я **Володимир**

По батькові **Павлович**

Дата народження **28/07/2001**

Змінити пароль

Старий пароль *******

Новий пароль |

Зберегти

Рисунок 3.13 – Вікно редагування профілю

3.2 Опис програмних модулів

Для структурованого зберігання інформації про погосподарства було прийнято рішення використовувати реляційну базу даних в якості СУБД - MS SQL Server, а в якості інформаційно-логічного мови баз даних - SQL. Вибір полягає в тому, що SQL є поширеною мовою для роботи з базами даних. В даний час майже всі найбільші розробники СУБД створюють свої продукти за допомогою мови SQL або SQL-інтерфейсу. Розробники та користувачі дуже його люблять. Він був включений до архітектури додатків і є стратегічним вибором багатьох великих і впливових організацій. Для розробки програмного забезпечення було обрано інтегроване середовище розробки Microsoft Visual Studio 2019. Даний продукт є передовим рішенням для розробки, що дозволяє командам будь-якого розміру проєктувати і створювати привабливі додатки, які задовольняють найвибагливіші запити замовників [15]. В якості мови програмування обрано С#. Ця мова програмування дозволяє використовувати сучасні концепції, технології та бібліотеки. Також С#

прекрасно підходить для розробки клієнт-серверних додатків, оскільки взаємодіяти йому з базою даних дозволяють такі технології як ADO.NET, Entity Framework.

Короткий опис основних програмних модулів наведено у таблиці 3.1.

Таблиця 3.1 – Короткий опис основних програмних модулів.

№	Назва модуля	Опис модуля
1	Loginu.cs	Форма, за допомогою якої користувач може зареєструватися в програмі.
2	Register.cs	Форма, яка використовується для реєстрації користувача в програмі.
3	Loader.cs	Форма прелоадер.
4	MainMenucs	Основна форма програми.
5	InventorCard.cs	Форма, яка показує інвентарну картку.
6	HomePage.cs	Форма головного меню.
7	Inform.cs	Форма перегляду даних про погосподарства.
8	FormInsert.cs	Форма для додавання даних нового погосподарства.
9	EditProf.cs	Профіль користувача.
10	Search.cs	Форма пошуку.
11	Documemts.cs	Форма для створення документів.

Таблиця 3.1 містить опис основних програмних модулів, які допомагають зрозуміти структуру програмного засобу, а також інформацію про його призначення, взаємодію та варіанти використання в структурі проєкту.

3.3 Опис результатів тестування

За останні два десятиліття комп'ютерні системи і програмне забезпечення проникло майже в усі області людської діяльності. Розповсюдження програмного

забезпечення призвело до зростання попиту на ефективне програмне забезпечення серед суспільства. Необхідність розробки великої кількості програмних продуктів за короткий проміжок часу зростає. Ця ситуація, зокрема, спонукає до підвищених вимог щодо забезпечення необхідної якості програмного забезпечення. Виготовлення високоякісного програмного забезпечення залежить від тестування.

Тестування програмного забезпечення — це метод вимірювання якості розробленого програмного забезпечення. Якість зазвичай визначається такими речами, як коректність, повнота та безпека, але може включати більше технічних вимог, які описані в стандарті ISO 9126 [16].

Існує кілька ознак, за якими прийнято робити класифікацію видів тестування. Розглянемо класифікацію за об'єктом тестування (функціональне тестування), а також класифікацію за ступенем автоматизації у вигляді ручного тестування (manual testing). Виявлення невідповідностей між реальною поведінкою реалізованих функцій і очікуваною поведінкою відповідно до специфікації та вимог називається функціональним тестуванням. Процес ручної перевірки програмного забезпечення на помилки називається ручним тестуванням. Тестувальник повинен відігравати користувача програми та використовувати її властивості, щоб виявити помилки в роботі програми.

Результати тестувань наведено у таблицях 3.2 та 3.5.

Таблиця 3.2 – Test-case для автоматизованої інформаційної системи «Автоматизований облік»

ID	Тип	Опис	Очікуваний результат	Реальний результат	Pass/ Fail
1	2	3	4	5	6
01_All	positive	Відображення на різних розширеннях монітора	У інтерфейсі немає спотворень, усі компоненти знаходяться на своїх місцях.	У інтерфейсі немає спотворень, усі компоненти знаходяться на своїх місцях.	PASS

Продовження таблиці 3.2

02_All	positive	Відображення паролю char *	Коли ви вводите пароль, усі символи будуть відображатися у вигляді *	Коли ви вводите пароль, усі символи будуть відображатися у вигляді *	PASS
03_All	positive	Функція переходу по компонентах клавішею Tab	Перехід курсора між полями введення та кнопками повинен відбуватися.	Перехід курсора між полями за допомогою Tab працює у всіх випадках	PASS
04_All	negative	У поле логіну при реєстрації введений вже існуючий логін	Поле з помилкою	Поле з помилкою	PASS
05_All	positive	У поле паролю користувача введено невірний пароль	Відображення повідомлення із вказаною помилкою	Відображення повідомлення із вказаною помилкою	PASS
06_All	positive	У поле логіну і паролю введено вірні дані	Вхід в систему за обліковим записом	Вхід в систему за обліковим записом	PASS
07_All	positive	Вхід під профілем користувача	Ті функції програми, які визначають права доступу, доступні.	Ті функції програми, які визначають права доступу, доступні.	PASS
08_All	positive	У поле пошуку введено неправильні дані.	Відображення повідомлення помилки	Відображення повідомлення помилки	PASS

Для перевірки коректного відображення форми написано тести, результати яких наведено у таблицях 3.3 – 3.6.

Таблиця 3.3 – Опис тест-кейсу «Перевірка відображення форми»

Дія	Очікуваний результат	Результат тесту
Відкрити форму «Авторизації»	<ul style="list-style-type: none"> – Форма «Авторизації» – назва форми – SingIn; – на формі два поля для введення даних «Користувач» і «Пароль»; – кнопка «Увійти» доступна; – кнопка «Зареєструватися» доступна 	PASS

Назва: Перевірка відображення форми.

Дія: Відкрити форму «Авторизації».

Перевірка: Перевірити, щоб форма, яка відображається, співпадала з формою на рисунку 3.14.

Рисунок 3.14 – Форма «Авторизації»

Таблиця 3.4 – Опис тест-кейсу «Перевірка відображення форми»

Дія	Очікуваний результат	Результат тесту
Відкрити форму «Головне вікно»	<ul style="list-style-type: none"> – Форма «Головне вікно»; – Назва форми –MainMenu; – У лівій частині форми є меню для взаємодії користувача з програмою – В лівому верхньому куті вікна розташовується профіль користувача – В центрі форми розташована інформація про погосподарства 	PASS

Назва: Перевірка відображення форми.

Дія: Відкрити форму «Головне вікно».

Перевірка: Перевірити, щоб форма, яка відображається, співпадала з формою, що на рисунку 3.15.

Особистий рахунок №	ZIP-код	Будинок №	Країна	Регіон	Область
11	43445	4	Україна	Гусятинський	Тернопільська
223	43445	4	Україна	Гусятинський	Тернопільська
231	35466	3	Україна	Гусятинський	Тернопільська
514	45343	6	Україна	Гусятинський	Тернопільська

Рисунок 3.15 – Головне вікно програми

Таблиця 3.5 – Опис тест-кейсу «Перевірка успішного входу в програму»

Дія	Очікуваний результат	Результат тесту
Відкрити форму «Авторизації» Ввести вірні дані для входу	– Форма «Авторизації» – введення даних користувача – після успішного входу перехід на головну сторінку програми	PASS

Назва: Перевірка відображення успішного входу в програму.

Дія: Відкрити форму «Авторизації», ввести вірні дані для входу.

Перевірка: Перевірити можливість входу в систему відповідно до облікового запису (див. Рис. 3.16)

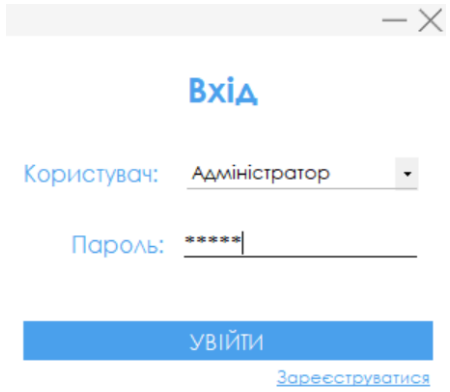


Рисунок 3.16 – Форма «Авторизації»

Після введення коректних даних відкривається головне вікно програми. (див. Рис. 3.17)

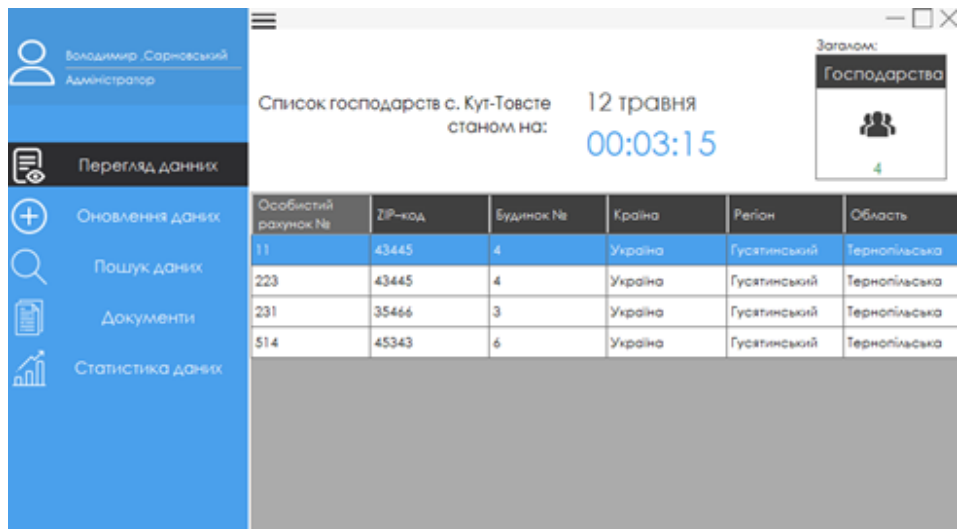


Рисунок 3.17 – Головне вікно програми

Таблиця 3.6 – Опис тест-кейсу «Перевірка відображення помилки при введенні некоректних даних у вікно пошуку»

Дія	Очікуваний результат	Результат тесту
У поле для пошуку введені некоректні дані	<ul style="list-style-type: none"> Форма «Головне вікно»; у поле пошуку вводяться невірні дані виведення помилки про некоректність даних 	PASS

Назва: У поле для пошуку введені некоректні дані.

Дія: Введення некоректних даних у поле пошуку.

Перевірка: Виведення відповідної помилки про некоректність введених даних (див. Рис. 3.18).

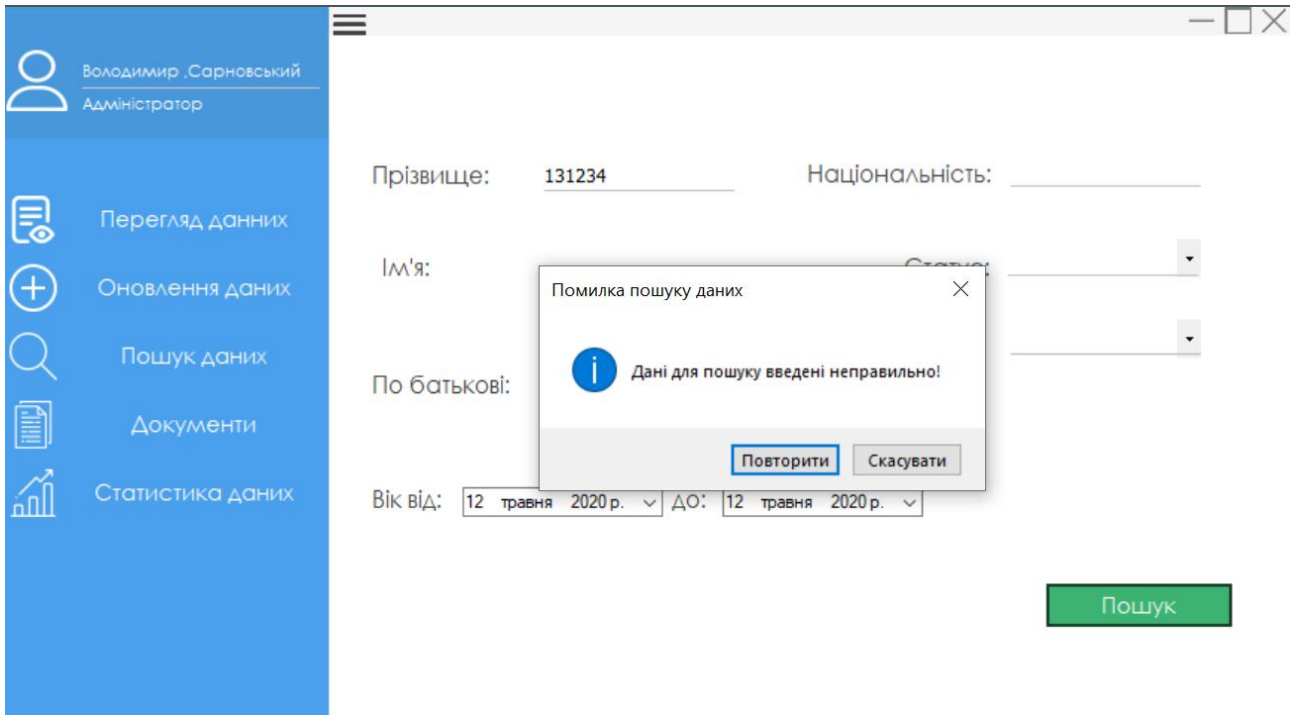


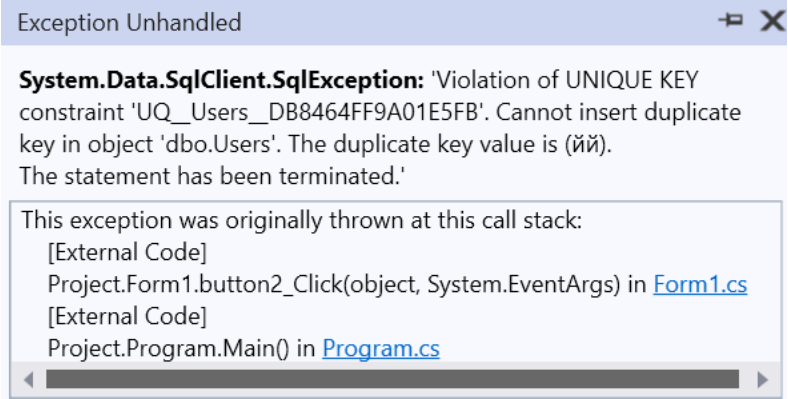
Рисунок 3.18 – Повідомлення про помилку

У процесі тестування програми були виявлені недоліки, які описано в таблиці 3.5.

Таблиця 3.5 – Bug-report для автоматизованої інформаційної системи «Автоматизований облік»

Автоматизована інформаційна система «Автоматизований облік»	
1	2
ID номер	Тест 1
Короткий опис (Summary)	Якщо в поле логіну при реєстрації введений вже існуючий логін програма зависає
Проект (Project)	«Автоматизований облік»

Продовження таблиці 3.5

1	2
Компонент додатку (Component)	Реєстрація
Номер версії (Version)	Версія, при якій було виявлено баг 1.2.7
Серйозність (Severity)	S2 Критичний (Critical)
Пріоритет (Priority)	P1 Високий (High)
Автор (Author)	Сарновський Володимир
Призначення для виправлення (AssignedTo)	Сарновський Володимир
ОС	Операційна система Windows 10 build 1803, .NetFramework 4.7 Процесор: Intel(R) Core(TM) i7-5600U 2.60GHz ОЗУ: 8 Гб Тип системи: 64-розрядна операційна система.
Опис	
Кроки відтворення (StepstoReproduce)	Запуск ПЗ Реєстрація Ввести логін існуючого користувача
Фактичний результат (Result)	Зависання програми
Очікуваний результат (ExpectedResult)	Повідомлення про співпадіння
Atteachment	

Вирішення проблеми було досягнуто за допомогою системи опрацювання помилок. Лістинг системи опрацювання помилок наведено нижче

Лістинг 3.1 – Метод вирішення помилки

```

public bool Login(string user, string pass)
{
    connection.Open();
    using (var command = new SqlCommand())
    {
        command.Connection = connection;
        command.CommandText = "select * from Users where LoginName=@user and Password=@pass";
        command.Parameters.AddWithValue("@pass", pass);
        command.Parameters.AddWithValue("@user", user);
        command.CommandType = CommandType.Text;
        SqlDataReader reader = command.ExecuteReader();
        if (reader.HasRows)
            {
                while (reader.Read())
                    {
                        UserCache.IdUsers = reader.GetInt32(0);
                        UserCache.LoginNames = reader.GetString(1);
                        UserCache.Passwords = reader.GetString(2);
                        UserCache.FirstNames = reader.GetString(3);
                        UserCache.LastNames = reader.GetString(4);
                        UserCache.Positions = reader.GetString(5);
                    }
            }
        return true;
    }
    else
        return false;
}

UserModel user = new UserModel();
var validLogin = user.LoginUser(textBox3.Text, textBox4.Text);
if (validLogin == true)
    {
        ErrorReg("Введенні дані існують");
    }
}

```

Алгоритм опрацювання помилок перевіряє, чи є схожі логіни в базі даних. Якщо логін знайдено – виводить помилку про повторну спробу ввести дані.

Отже, після проведеного набору тестів, можна сказати, що в програмі були знайдені деякі помилки, які в результаті опрацювання були виправлені. Після тестування і опрацювання помилок якість ПЗ значно підвищилась.

4 БЕЗПЕКА ЖИТТЄДІЯЛЬНОСТІ, ОСНОВИ ОХОРОНИ ПРАЦІ

4.1 Працездатність людини – оператора.

Під працездатністю людини розуміють можливість її виконувати роботу з необхідною якістю та в установлений час. Працездатність людини залежить як від зовнішніх чинників, так і від внутрішнього стану (внутрішні чинники). До зовнішніх чинників належать: кількість та форма отриманої інформації, зручність робочого місця, характер взаємостосунків в колективі, вплив чинників середовища існування. До внутрішніх чинників належать: рівень підготовки, тренуваність людини та її емоційна стійкість. У процесі роботи людина переживає різні функціональні стани, які зумовлюють різні рівні її працездатності[18].

Виділяють 4 фази працездатності: пристосування до праці, стійкої працездатності, субкомпенсації, втоми. Тривалість усіх фаз та усього циклу роботи залежить від рівня підготовки людини до роботи.

Фаза пристосування до праці – це час, протягом якого людина адаптується до майбутніх умов праці. Основний показник поступово досягає свого встановленого значення.[18] Тривалість періоду пристосування організму до умов праці залежить від багатьох чинників, серед яких основними є інтенсивність роботи (чим інтенсивніша робота, тим цей період коротший) та рівень готовності людини до майбутньої роботи.

Значного скорочення фази пристосування до праці можна досягти за рахунок попередньої підготовки людини до роботи (виконання фізичних вправ, адаптації зору, слуху та ін.) та шляхом посиленого навчального навантаження. Суть останнього полягає в тому, що оператор перед початком роботи проводить короткочасне тренування щодо розв'язання однієї чи кількох задач підвищеної складності.

Фаза стійкої працездатності характеризується найвищою якістю праці при оптимальних рівнях функціонування фізіологічних систем організму. Тривалість цього періоду залежить від інтенсивності роботи. Чим інтенсивніша праця, тим коротший цей період. Найоптимальніша динамічна робота, коли цей період може бути в десятки разів довшим, ніж при статичній діяльності. На процес стійкої працездатності великий вплив справляють емоції. Негативні (страх, невпевненість, поганий настрій) знижують працездатність. Позитивні (впевненість, спокій, бадьорий настрій) значно продовжують період стійкої працездатності[18].

Продовження періоду стійкої працездатності можна забезпечити:

- оптимальним рівнем напруги психофізіологічних функцій;
- комфортними умовами праці;
- правильним поєднанням режимів праці та відпочинку;
- емоційним розвантаженням;
- використанням тонізуючих напоїв (кава, чай), фармакологічних засобів, зокрема препаратів рослинного походження (вітаміни, препарати, які впливають на енергетичні та метаболічні процеси);
- інформуванням людини про наслідки її діяльності, наглядом та контролем її роботи.

Фаза субкомпенсації розглядається як початок розвитку втоми. В цей період якість праці ще зберігається на високому рівні, але тільки за рахунок перенапруги і відповідних функцій організму.

Фаза втоми (з моменту у характеризується чітко вираженим зниженням якості роботи при подальшому погіршенні функціонального стану людини. Об'єктивними показниками втоми є зміна частоти пульсу, дихання, зорової та слухової чутливості.

Наступною фазою життєдіяльності людини повинна бути фаза відновлення працездатності (відпочинку), яка може тривати від 3 до 5 хвилин; 60 – 90 хв. і навіть декілька діб.

4.2 Заходи пожежної безпеки підприємства, селищної ради.

Заходи пожежної безпеки є надзвичайно важливими для забезпечення безпеки працівників та майна на підприємстві або в межах селищної ради. Відповідно до законодавства багатьох країн, включаючи Україну, підприємства та селищні ради зобов'язані вживати заходів для попередження пожеж та забезпечення ефективного реагування у разі їх виникнення. У цій статті розглянемо деякі основні аспекти пожежної безпеки, а також запропонуємо конкретні заходи, які можуть бути впроваджені на підприємстві або в межах селищної ради.

Один із найважливіших аспектів пожежної безпеки - це попередження пожеж. У цьому контексті слід приділити особливу увагу плануванню та розміщенню пожежних сповіщувачів і знаків евакуації, а також системі пожежного спостереження. На підприємстві або в межах селищної ради повинні бути встановлені відповідні пожежні сповіщувачі, які забезпечують оперативну інформацію про виникнення пожежі. Знаки евакуації повинні бути чіткими та легко розпізнаваними, а плани евакуації повинні бути розроблені та періодично практикуватися з метою підготовки персоналу до швидкого та безпечного покидання будівлі у разі небезпеки. Крім того, системи пожежного спостереження, такі як пожежні датчики та камери спостереження, можуть виявити пожежу на ранніх стадіях та дати можливість оперативно зреагувати на неї.

На підприємстві або в межах селищної ради також необхідно мати належно обладнані та функціональні системи пожежогасіння. Вони включають в себе вогнегасники, гідранти, автоматичні пожежні системи та інші засоби пожежогасіння.

Персонал повинен бути навчений використовувати ці засоби, а також регулярно перевіряти їх стан та наявність необхідних запасних частин.

Безпека електричних систем також має велике значення для запобігання пожежам. Регулярна перевірка електрообладнання, виявлення потенційних проблем, таких як перегрів або коротке замикання, та своєчасне проведення ремонтів і заміні можуть значно знизити ризик пожежі, спричиненого електричними системами.

Крім того, особливу увагу слід приділити організації навчань та тренувань з пожежної безпеки для персоналу. Всі працівники повинні бути свідомі пожежних загроз та знати процедури поведінки у разі пожежі. Регулярні навчання та тренування допоможуть усвідомити ризики, навчитися ефективно реагувати та запобігати пожежам, а також забезпечити швидку та безпечну евакуацію у разі небезпеки.

Також важливо проводити періодичні перевірки пожежної безпеки та аудити, щоб переконатися, що всі заходи пожежної безпеки виконуються належним чином. Це може включати перевірку пожежного обладнання, виконання планів евакуації, перевірку електричних систем та інші заходи. Результати таких перевірок повинні фіксуватися та вживатися необхідні заходи для усунення виявлених недоліків.

Необхідно також встановити відповідну структуру управління пожежною безпекою на підприємстві або в межах селищної ради. Це може включати призначення відповідальних осіб, які будуть відповідати за планування та впровадження заходів пожежної безпеки, контроль за дотриманням правил та процедур, а також співпрацю з пожежними службами та іншими відповідними організаціями

Залучення спеціалістів з пожежної безпеки може бути додатковою корисною практикою. Ці фахівці можуть провести детальний аналіз пожежного ризику на підприємстві або в межах селищної ради, розробити рекомендації з покращення пожежної безпеки та надати консультації з усунення потенційних загроз.

Узагальнюючи, заходи пожежної безпеки на підприємстві або в межах селищної ради є важливою складовою безпеки працівників та майна. Вони включають попередження пожеж, наявність належно обладнаних систем пожежогасіння, регулярні навчання та тренування персоналу, перевірку електричних систем, розміщення пожежних сповіщувачів та знаків евакуації, а також аудити пожежної безпеки та співпрацю зі спеціалістами з пожежної безпеки.

Реалізація цих заходів вимагає систематичного підходу та постійного контролю. Планування, впровадження та оновлення заходів пожежної безпеки повинні бути частиною загальної стратегії безпеки на підприємстві або в межах селищної ради. Постійна комунікація з працівниками та мешканцями, своєчасна інформація про правила та процедури пожежної безпеки, а також систематичні перевірки та оцінки ризиків є ключовими елементами успішної реалізації цих заходів.

Забезпечення пожежної безпеки на підприємстві або в межах селищної ради - це не тільки вимога законодавства, але й моральна та соціальна відповідальність перед співробітниками, мешканцями та громадою загалом. Це допомагає створити безпечне та надійне середовище для всіх присутніх, а також запобігти втратам життя та майнових цінностей.

Зрештою, реалізація заходів пожежної безпеки на підприємстві або в межах селищної ради вимагає постійного вдосконалення та адаптації до змінних умов та вимог. Регулярний аналіз ефективності заходів, врахування нових технологій та стандартів безпеки, а також взаємодія з пожежними органами і відповідними організаціями з пожежної безпеки допомагають забезпечити постійну безпеку та захист від можливих пожежних небезпек.

Крім запобігання пожежам, також важливо мати плани надзвичайних ситуацій та евакуації на випадок пожежі. Ці плани повинні бути розроблені і регулярно оновлюватися для кожного підприємства або території селищної ради. Вони повинні включати точні інструкції щодо евакуації працівників та мешканців, визначення

безпечних шляхів та зон збору, а також забезпечувати взаємодію з відповідними аварійними службами. Регулярні тренування та навчання з практичними ситуаціями евакуації допомагають перевірити ефективність планів та підготувати персонал до швидкої та безпечної евакуації.

ВИСНОВОК

При виконанні дипломної роботи було розроблено АІС «Автоматизований облік», яке дозволяє вести облік даних погосподарських книг. Саме програмне забезпечення складається з клієнтського додатку і серверної частини. Клієнтську частину програму реалізовано за допомогою мови C # в інтегрованому середовищі розробки Visual Studio 2019.

Серверна частина а саме база даних програмного забезпечення реалізована за допомогою мови SQL в СУБД Microsoft SQL Server Management Studio 18. Створена база даних включає в себе таблиці, уявлення і збережені процедури. На етапі аналізу якості програмного продукту було проведено функціональне і модульне тестування. На основі аналізу результатів тестування можна зробити висновок, що розроблене програмне забезпечення працює коректно.

Таким чином, дипломна робота призвела до розробки програмного забезпечення, яке задовольняє всі вимоги до технічного завдання. Розроблене програмне забезпечення буде успішно використовуватися в Гримайлівській селищній раді.

ПЕРЕЛІК ПОСИЛАНЬ

1. Про затвердження Інструкції з ведення попогосподарського обліку в сільських, селищних та міських радах : наказ Міністерства юстиції України від 11.04.2016 № 56. Офіційний вісник України. 2016. № 39, С. 278-283.
2. Рекомендації по введенню попогосподарських книг: URL: http://www.kr.ukrstat.gov.ua/include/mvc/view/other/pog_oblik.php (дата звернення 14.02.2023).
3. Про місцеве самоврядування в Україні : Закон України від 21.05.1997 р. № 280/97-ВР. Дата оновлення: 28.11.2019. URL: <https://zakon.rada.gov.ua/laws/show/280/97> (Дата звернення 20.02.2023).
4. Порядок реєстрації осіб, які здійснюють погосподарську діяльність URL: <http://slovyanska.rda.org.ua/news/1582893921/> (дата звернення 22.02.2023).
5. Про особисте селянське погосподарство : Закон України від 15.05.2003р. № 742-IV. Дата оновлення: 12.02.2015. URL: <https://zakon.rada.gov.ua/laws/show/742/15> (Дата звернення 20.03.2023).
6. Вказівки по веденню попогосподарського обліку в сільських Радах народних депутатів: URL: <https://zakononline.com.ua/court-decisions/show/83301546> (дата звернення 30.03.2023).
7. Автоматизована система «Розумне село»: URL: <https://edt.in.ua/software/279-rozumne-selo.html> (дата звернення 05.04.2023).
8. Коннолли Томас. Бази даних: проєктування, реалізація і супроводження. Теорія і практика: під. пос., пер. з англ. СПб. : BHV, 2011. 1096 с.
9. Гольцман В.І. MySQL 5.0. Бібліотека програміста. Спб. : Пітер, 2010. 253 с.

10. Чому SQL Server data management system? URL: <https://www.microsoft.com/en/server-cloud/datamanagement/why.aspx> (дата звернення 20.04.2023).
11. Дейт К. Дж. Введення в систему баз даних: пров. з англ. М. : Вільямс, 2005. 1328 с.: ил.
12. Нейгел, К. C# 5.0 і платформа .NET 4.5 для професіоналів. М. : Вільямс, 2014. 1440 с.
13. Практичний посібник зі створення UML-діаграм. URL: https://flexberry.github.io/ru/gpg_practical-guides-uml.html (дата звернення 29.04.2023).
14. Макгрегор Джон, Сайкс Девид. Тестування об'єктно-орієнтованого програмного забезпечення. Практичний посібник: пров. з англ. К. : ООО «ТИД «ДС», 2002. 432 с.
15. Брукс Ф. Міфічний людино-місяць або як створюються програмні системи. СПб. : Символ-Плюс, 1999. 304 с.
16. Охорона праці при роботі з комп'ютером. URL: <https://www.victorija.ua/dovidnik/osnovni-pravyyla-dotrymannyya-ohorony-pratsi-pry-roboti-na-personalnyh-eom.html> (дата звернення 12.05.2023).
17. Працездатність людини – оператора: URL: <https://school.hometask.com/pracezdatnist-lyudini-operatora-xarakteristika-zhittyediyalnosti-lyudini-u-sistemi-lyudina-mashina-seredovishhe-isnuvannya/> (дата звернення 14.05.2023).
18. Заходи пожежної безпеки підприємства, селищної ради: <https://arbtg.gov.ua/news/1642403700/> (дата звернення 15.05.2023).
19. ДСТУ 8604:2015. Дизайн і ергономіка. Робоче місце для виконання робіт у положенні сидячи. Загальні ергономічні вимоги.[Чинний від 2017-07-01]. Вид. офіц. Київ, 2015. 10 с.(Інформація та документація).

ДОДАТКИ

Додаток А

Код програми «Автоматизований облік»

```
using System;
using System.Collections.Generic;
using System.ComponentModel;
using System.Data;
using System.Drawing;
using System.Linq;
using System.Text;
using System.Threading.Tasks;
using System.Data.SqlClient;
using System.Windows.Forms;
namespace Project
{

public partial class Form1 : Form
    {

public Form1()
    {
InitializeComponent();
    }
public SqlConnection connection = new SqlConnection("DataSource=DESKTOP-
KR70FIK;Initial Catalog=duplo;Integrated Security=True");
private void textBox2_TextChanged(object sender, EventArgs e)
    {

    }

private void label7_Click(object sender, EventArgs e)
    {
if (label7.Text == "Реєстрація")
    {
        label7.Text = "Вхід";
        timer1.Start();
    }
elseif (label7.Text == "Вхід")
    {
        label7.Text = "Реєстрація";
        timer1.Start();
    }
}private void timer1_Tick(object sender, EventArgs e)
    {
```

Продовження Додатку А

```

if (label7.Text == "Вхід")
    {
        panel3.Left -= 35;
        panel4.Left -= 35;
if (panel3.Left <= -245)
    {
        timer1.Stop();
    }
    }
elseif (label7.Text == "Реєстрація")
    {
        panel3.Left += 35;
        panel4.Left += 35;
if (panel3.Left >= 0)
    {
        timer1.Stop();
    }
    }
}
privatevoid button1_Click(objectsender, EventArgs e)
    {
if (textBox2.Text != "Ім'я користувача")
    {
if (textBox1.Text != "Пароль")
    {
UserModeluser = newUserModel();
varvalidLogin = user.LoginUser(textBox2.Text, textBox1.Text);
if (validLogin == true)
    {
this.Hide();
Loaderloader = newLoader();
loader.ShowDialog();
MainMenumainMenu = newMainMenu();

mainMenu.Show();

    }

else
    {
ErrorLogin("Введенні дані не правильні!");
    }

    }
}
}

```

Продовження Додатку А

```

else
    {
    ErrorLogin(" Введіть пароль!");
    }
else
    {
    ErrorLogin(" Введіть ім'я користувача!");
    }
if (textBox2.Text == "Ім'я користувача" && textBox1.Text == "Пароль")
    {
    ErrorLogin(" Введіть дані!");
    }
privatevoidErrorLogin(stringer)
    {
    ErrorlBLog.Text = "      " + er;
    ErrorlBLog.Visible = true;
    }
privatevoidErrorReg(stringer)
    {
    ErrorlBReg.Text = "      " + er;
    ErrorlBReg.Visible = true;
    }
privatevoid textBox2_Enter(objectsender, EventArgs e)
    {
    if (textBox2.Text == "Ім'я користувача")
        {
        textBox2.Text = "";
        textBox2.ForeColor = Color.Black;
        }
    }
privatevoid textBox2_Leave(objectsender, EventArgs e)
    {
    if (textBox2.Text == "")
        {
        textBox2.Text = "Ім'я користувача";
        textBox2.ForeColor = Color.DarkGray;
        }
    }

privatevoid textBox1_Enter(objectsender, EventArgs e)
    {
    if (textBox1.Text == "Пароль")

```


Продовження Додатку А

```

        {
            textBox1.Text = "";
            textBox1.ForeColor = Color.Black;
        }
    }

private void textBox1_Leave(object sender, EventArgs e)
    {
    if (textBox1.Text == "")
        {
            textBox1.Text = "Пароль";
            textBox1.ForeColor = Color.DarkGray;
        }
    }

private void button2_Click(object sender, EventArgs e)
    {

connection.Open();

UserModel user = newUserModel();
var validLogin = user.LoginUser(textBox3.Text, textBox4.Text);
if (validLogin == true)
    {
    ErrorReg("Введенні дані існують!");
    }
else
using (var command = new SqlCommand())
    {
    command.Connection = connection;
    command.Parameters.AddWithValue("@LoginName", textBox3.Text);

    command.Parameters.AddWithValue("@Password", textBox4.Text);
    command.CommandText = "InsertintoUsers(LoginName, Password, FirstName,
    LastName, Position) Values(@LoginName, @Password, ' ', ' ', ' ')";

    command.CommandType = CommandType.Text;
    command.ExecuteNonQuery();
    ErrorLBReg.Visible = true;
    ErrorLBReg.ForeColor = Color.Green;
    ErrorLBReg.Text = ("Реєстрація пройшла успішно");
    }
connection.Close();

    }

```

Продовження Додатку А

```
privatevoid textBox3_Enter(objectsender, EventArgs e)
    {
    if (textBox3.Text == "Ім'я користувача")
        {
            textBox3.Text = "";

            textBox3.ForeColor = Color.Black;
        }
    }

privatevoid textBox3_Leave_1(objectsender, EventArgs e)
    {
    if (textBox3.Text == "")
        {
            textBox3.Text = "Ім'я користувача";
            textBox3.ForeColor = Color.DarkGray;
        }
    }

privatevoid textBox4_Enter_1(objectsender, EventArgs e)
    {
    if (textBox4.Text == "Пароль")
        {
            textBox4.Text = "";

            textBox4.ForeColor = Color.Black;
        }
    }

privatevoid textBox4_Leave_1(objectsender, EventArgs e)
    {
    if (textBox4.Text == "")
        {
            textBox4.Text = "Пароль";
            textBox4.ForeColor = Color.DarkGray;
        }
    }

privatevoid label9_Click(objectsender, EventArgs e)
    {
    }

privatevoid error1B_Click(objectsender, EventArgs e)
    {
```

Продовження Додатку А

```

        }
    }
}
using System;
using System.Collections.Generic;
using System.ComponentModel;
using System.Data;
using System.Drawing;
using System.Linq;
using System.Text;
using System.Threading.Tasks;

using System.Windows.Forms;

namespace Project
{
    public partial class MainMenu : Form
    {
        public MainMenu()
        {
            InitializeComponent();
        }
        private void btnMenu_Click(object sender, EventArgs e)
        {
            if (panel1.Width == 200)
            {
                panel1.Width = 60;
            }
            else
            {
                panel1.Width = 200;
            }
        }

        public void Opform( object forms)
        {
            if(this.panel3.Controls.Count > 0)

                this.panel3.Controls.RemoveAt(0);
            Form f = forms as Form;
            f.TopLevel = false;
            f.Dock = DockStyle.Fill;
            this.panel3.Controls.Add(f);
            this.panel3.Tag = f;
        }
    }
}

```

Продовження Додатку А

```

f.Show();

    }
privatevoidMainMenu_Load(objectsender, EventArgs e)
    {
btnlogoInicio_Click(null, e);
profes.Text = UserCache.LoginName;
username.Text = UserCache.FirstName + "," + UserCache.LastName;
    }
DataGriddata = newDataGrid();

HomePageho = newHomePage();

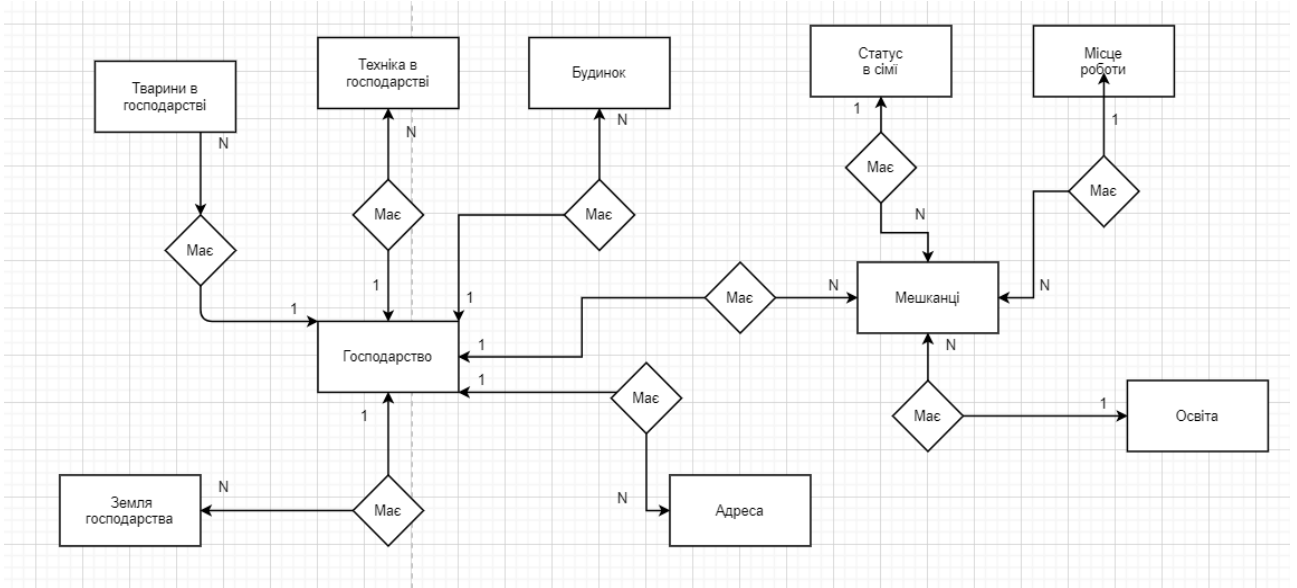
privatevoidbtnlogoInicio_Click(objectsender, EventArgs e)
    {
Opform(newHomePage());
    }
privatevoid pictureBox1_Click(objectsender, EventArgs e)
    {
        User op = new User();
op.Show();
    }

privatevoidbtnprod_Click(objectsender, EventArgs e)
    {
Opform(newFormInsert());
    }
privatevoid button1_Click(objectsender, EventArgs e)
    {
Opform(newHomePage());
    }
privatevoid button2_Click(objectsender, EventArgs e)
    {
Opform(newSearch());
    }
privatevoid button3_Click(objectsender, EventArgs e)
    {
Opform(newDocumemts());
    }
    }
}

```

Додаток Б

ER-діаграма бази даних «Diplom»



Додаток В

Діаграма зав'язків бази даних «Diplom»

