

МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ

ОДЕСЬКИЙ НАЦІОНАЛЬНИЙ УНІВЕРСИТЕТ імені І.І.МЕЧНИКОВА

(повне найменування вищого навчального закладу)

Факультет математики, фізики та інформаційних технологій

(повне найменування інституту, назва факультету (відділення))

Кафедра математичного забезпечення комп'ютерних систем

(повна назва кафедри (предметної, циклової комісії))

Дипломна робота

на здобуття ступеня вищої освіти «бакалавр»

(освітньо-кваліфікаційний рівень)

на тему Інформаційна система управління соціального захисту населення
Information system of social protection management department

Виконав: студент денної форми навчання

спеціальності 123 – Комп'ютерна інженерія

(шифр і назва напрямку підготовки, спеціальності)

Прохоров Олександр Олександрович

(прізвище, ім'я, по-батькові)

Керівник Розновець О.І.

(науковий ступінь, вчене звання, прізвище та ініціали)

Рецензент Лісіцина І.М.

(науковий ступінь, вчене звання, прізвище та ініціали)

Рекомендовано до захисту:

Протокол засідання кафедри

№ 11 від «10» червня 2022 р.

Завідувач кафедри

Захищено на засіданні ЕК №

протокол № від « » 2022 р.

Оцінка / /
(за національною шкалою, шкалою ECTS, бали)

Голова ЕК

Євгеній МАЛАХОВ

(підпис)

(прізвище, ініціали)

Надія КАЗАКОВА

(підпис)

(прізвище, ініціали)

Одеса - 2022

АНОТАЦІЯ

Метою дипломного проекту є проектування і створення інформаційної системи для управління соціального захисту населення. Користувачами даної системи є відвідувачі управління, його співробітники та адміністратори.

Інформаційна система реалізована у вигляді веб-застосунку на основі фреймворку Razor Pages, який є частиною ASP.NET Core. Для зберігання даних та маніпулювання ними використана СУБД PostgreSQL. Як основна мова програмування використана C#.

У розробленій інформаційній системі реалізований облік співробітників управління, відвідувачів та призначених їм соціальних виплат, автоматизований процес прийняття документів, необхідних для призначення соціальної допомоги. В створеній системі реалізований захист від несанкціонованого доступу та здійснено розмежування повноважень різних категорій користувачів.

Результатом дипломного проекту є інформаційна система управління соціального захисту населення, що покликана спростити та пришвидшити такі етапи процедури призначення соціальної допомоги, як подання заяви на отримання соціальної допомоги на основі задекларованого доходу, стану здоров'я, кількості підопічних дітей, наявності підопічної людини з засвідченою групою інвалідності, а також надання довідок про отримання (неотримання) соціальних виплат.

ABSTRACT

The purpose of this graduate work is to design and create an information system for the management of social protection department. The users of this system are department visitors, its employees and administrators.

The information system is implemented as a web application based on the Razor Pages framework, which is part of ASP.NET Core. PostgreSQL DBMS is used to store and manipulate data. C# is used as the main programming language.

In the developed information system the account of employees of management, visitors and the social payments appointment is realized, the process of acceptance of the documents which are necessary for social help purpose is automated. The created information system implements protection against unauthorized access and differentiates the privileges of different user categories.

The result of the graduate work is an information system of the social protection department, which aims to simplify and speed up such stages of the procedure for assigning social assistance, such as applying for social assistance based on declared income, health, number of children, the presence of a ward with a certified disability group, as well as providing certificates of receipt (non-receipt) of social benefits.

ЗМІСТ

ВСТУП.....	5
1 АНАЛІЗ ПРЕДМЕТНОЇ ОБЛАСТІ.....	8
1.1 Користувачі інформаційної системи управління соціального захисту населення та їх задачі	8
1.2 Огляд існуючої системи для автоматизації процесу отримання соціальної допомоги.....	15
1.3 Висновок до розділу 1	16
2 ПРОЕКТУВАННЯ ІНФОРМАЦІЙНОЇ СИСТЕМИ УПРАВЛІННЯ СОЦІАЛЬНОГО ЗАХИСТУ НАСЕЛЕННЯ	17
2.1 Архітектура та шаблон проектування	17
2.2 Інформаційне моделювання предметної області	19
2.3 Модель клієнтського застосунку	29
3 ВИБІР ПРОГРАМНОГО ЗАБЕЗПЕЧЕННЯ ДЛЯ СТВОРЕННЯ ІНФОРМАЦІЙНОЇ СИСТЕМИ УПРАВЛІННЯ СОЦІАЛЬНОГО ЗАХИСТУ НАСЕЛЕННЯ	30
4 СТВОРЕННЯ ІНФОРМАЦІЙНОЇ СИСТЕМИ УПРАВЛІННЯ СОЦІАЛЬНОГО ЗАХИСТУ НАСЕЛЕННЯ	32
4.1 Створення бази даних	32
4.2 Запити до бази даних для вирішення поставлених задач.....	33
4.3 Програмна реалізація інтерфейсу	37
4.4 Програмна реалізація представлення-моделі.....	41
4.5 Безпека інформаційної системи.....	42
5 ІНСТРУКЦІЯ КОРИСТУВАЧА	45
ВИСНОВКИ	56
СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ	57
ДОДАТОК А Створення таблиць та типів даних.....	59
ДОДАТОК Б Створення тригерів.....	64

ВСТУП

Надання соціальної допомоги є одним із заходів соціального захисту малозабезпечених громадян і однією з найважливіших функцій будь-якої сучасної держави. Цей процес пов'язаний з багатьма труднощами. Для отримання довідки про соціальні виплати або для оформлення допомоги її здобувачеві доводиться багаторазово відвідувати відповідні заклади та витрачати чимало часу на очікування в чергах іноді лише для того, щоб отримати відмову через відсутність певних документів. Крім того, прийом та оформлення заяв на отримання допомоги також передбачає значні витрати часу та потребує наявності великої кількості фахівців.

Питання організації онлайн-сервісу для забезпечення потреб громадян у сфері соціального захисту існує не перший рік. Раніше громадяни навіть створювали петиції, у яких вказували на незручність постійних походів до найближчого управління праці та соціального захисту населення [1].

Актуальність оптимізації процесу взаємодії працівників управління та громадян підтверджується державними зусиллями у цьому напрямку, зокрема створенням застосунку «Дія», який дозволяє оформити допомогу для декількох випадків (наприклад, при народженні дитини) [2].

Створення спеціальної інформаційної системи для управління соціального захисту населення зможе охопити більшу кількість випадків, за яких громадянин може отримати допомогу від держави.

Призначенням соціальної допомоги населенню займається Управління праці та соціального захисту населення (УПіСЗН). Співробітники управління займаються прийомом відвідувачів, призначають їм соціальну допомогу за різними передбаченими законодавством ознаками і повідомляють, які документи необхідно зібрати для отримання допомоги. Всю цю роботу доводиться виконувати вручну, працюючи при цьому з величезною кількістю паперових документів. Інформаційна система для УПіСЗН покликана спростити роботу співробітникам управління: зменшити

обсяг паперової документації і знизити ймовірність виникнення помилок при оформленні соціальної допомоги.

Інформаційна система, оформлена як веб-застосунок, є вельми корисною і для здобувачів соціальної допомоги (відвідувачів УПіСЗН) – замість постійних телефонних дзвінків і візитів з витрачанням великої кількості часу на очікування в чергах вони зможуть швидко і зручно, просто зі свого дому, самостійно оформити заяву на отримання соціальних виплат. А особа, яка вже отримує соціальну допомогу, матиме змогу швидко у будь-який час отримати інформацію про актуальні соціальні виплати або переглянути історію їх нарахування.

Вищезазначене може суттєво спростити життя усім здобувачам та одержувачам соціальної допомоги, особливо тим, які обмежені власним станом здоров'я (особи з інвалідністю) або є опікунами людей, які потребують постійного догляду (осіб з інвалідністю або дітей). Крім того, подібна інформаційна система значно спростить роботу працівників управління праці та соціального захисту населення, оскільки звільнить їх від більшої частини роботи з відвідувачами. Взаємодія з відвідувачами обмежуватиметься отриманням пакета документів для завершення процесу схвалення заяви.

Таким чином, створений застосунок зможе використовуватися не тільки як інструмент, що спрощує життя пересічних громадян, а й як засіб оптимізації ресурсів управління праці та соціального захисту населення.

Отже, метою дипломного проекту є проектування і створення інформаційної системи для управління соціального захисту населення.

Для досягнення зазначеної мети необхідно вирішити наступні задачі:

- 1) виконати аналіз предметної області, зосередившись на потребах здобувачів соціальної допомоги та працівників управління праці та соціального захисту населення;

- 2) визначити категорії користувачів і сформулювати їх вимоги до створюваної інформаційної системи;

3) обрати архітектуру і шаблон проектування створюваної інформаційної системи;

4) спроектувати базу даних для зберігання інформації про соціальні виплати;

5) обрати технології та засоби реалізації інформаційної системи;

6) з урахуванням вибраних засобів створити БД, а також клієнтський застосунок, який дасть можливість різним категоріям користувачів ефективно маніпулювати даними предметної області відповідно до їх повноважень;

7) забезпечити цілісність і безпеку даних як на рівні БД, так і на рівні клієнтського застосунку;

8) забезпечити захист системи від несанкціонованого доступу і розмежування повноважень з боку різних категорій користувачів.

1 АНАЛІЗ ПРЕДМЕТНОЇ ОБЛАСТІ

1.1 Користувачі інформаційної системи управління соціального захисту населення та їх задачі

На початку розробки інформаційної системи перед розробником постає необхідність дослідити предметну область та зробити висновки, хто саме може бути потенційним користувачем створюваної системи та що саме в предметній області може цікавити потенційного користувача. Отже, перша задача – визначити коло користувачів інформаційної системи, визначити, які задачі постають перед ними та які інструменти вже використовуються для вирішення цих задач.

Інформаційна система управління соціального захисту населення повинна дозволяти співробітникам УПіСЗН:

- реєструвати відвідувачів та їхні документи;
- вносити інформацію про матеріальне становище відвідувача (декларацію про доходи та витрати, сімейне рухоме та нерухоме майно), його сімейний стан, стан здоров'я (персональний або родичів або підопічних);
- переглядати внесену відвідувачем інформацію;
- призначати та скасовувати соціальні виплати відвідувачу;
- відслідковувати стан призначених виплат.

Інформаційна система управління соціального захисту населення повинна дозволяти відвідувачам УПіСЗН:

- подавати заяву на отримання соціальної допомоги;
- вносити інформацію про матеріальне становище (декларацію про доходи та витрати, сімейне рухоме та нерухоме майно), сімейний стан, стан здоров'я (персональний або родичів або підопічних);

– відстежувати інформацію та отримувати довідки про всі (актуальні та неактуальні) призначені виплати.

Для оформлення соціальної допомоги громадянин повинен відвідати найближче УПіСЗН та зареєструвати обліковий запис, пред'явивши свій паспорт, ПІН та контактні дані (адресу та номер телефону). Після цього відвідувач може увійти в свій обліковий запис та почати процес оформлення декларації про доходи та витрати, а також внесення інформації щодо наявних довідок про стан здоров'я, кількість дітей тощо. Додавши необхідні дані, відвідувач може подати заяву на отримання допомоги, додавши до цієї заяви необхідну декларацію та довідки. Після подачі заяви її переглядає співробітник: він перевіряє додану відвідувачем інформацію на відповідність умовам отримання допомоги. Якщо внесена інформація не дає відвідувачу право на отримання соціальної допомоги, співробітник відхиляє заяву, змінюючи її статус на «відхилена», та вказує причину. Якщо ж заява приймається, її статус змінюється на «схвалена», і це означає, що для отримання допомоги відвідувачу потрібно лише принести до УПіСЗН фізичні (паперові) документи, відповідні до тих, інформація про котрі була внесена до системи. Отримавши пакет документів, співробітник фіксує номери документів та дату їх видачі, після чого призначає допомогу відвідувачу на певний період часу. Діаграма бізнес-процесу представлена на рис. 1.1.

Виходячи з вищезазначеного, створювана інформаційна система управління соціального захисту населення повинна підтримувати дві основні категорії користувачів – відвідувач та співробітник, а також адміністратор, який управляє обліковими записами співробітників інформаційної системи.

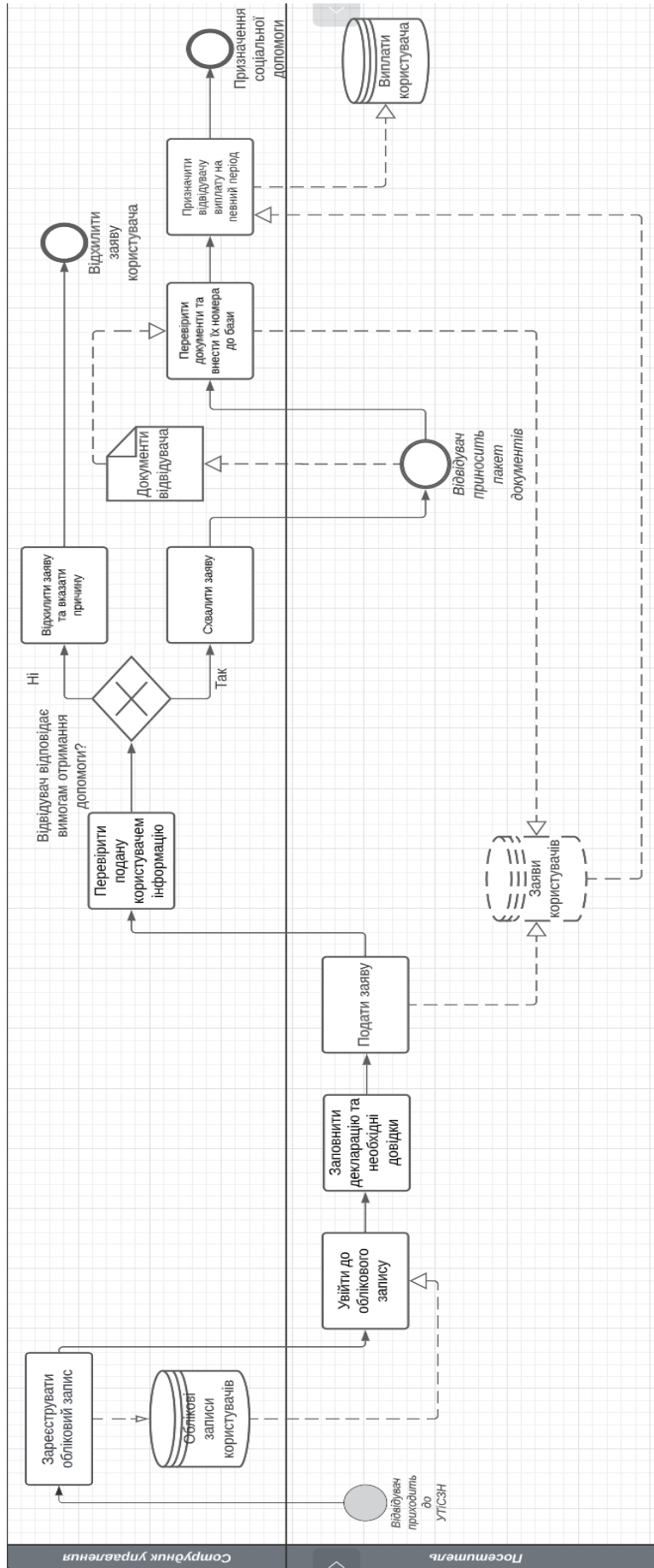


Рисунок 1.1 – Діаграма бізнес-процесу подачі заяви та оформлення соціальної виплати

Задачі користувачів інформаційної системи представлені у табл. 1.1.

Таблиця 1.1 – Задачі користувачів ІС

Номер	Задача	Вхідні дані	Вихідні дані
Адміністратор			
A1	Створити обліковий запис співробітника	Логін, ПІБ, пароль, категорія	Новий користувач-співробітник
A2	Редагувати обліковий запис співробітника	Користувач-співробітник, логін, ПІБ, посада, пароль, категорія	Оновлена інформація про користувача
A3	Видалити обліковий запис співробітника	Користувач-співробітник	Відсутні
Співробітник управління			
C1	Додати обліковий запис відвідувача	Логін, пароль, ПІБ, дата народження, ідентифікаційний код, співробітник, номер мобільного телефону, адреса, дані паспорту (серія, номер)	Новий відвідувач
C2	Редагувати обліковий запис відвідувача	Відвідувач, логін, пароль, ПІБ, дата народження, ПІН, співробітник, дані паспорту (серія, номер)	Оновлена інформація про відвідувача
C3	Видалити обліковий запис відвідувача	Відвідувач	Відсутні
C4	Призначити відвідувачу субсидію за рівнем доходу	Заява, декларація (ПІБ членів родини, ступінь спорідненості, ПІН, примітки; тип, розмір, джерело, період доходу; призначення, вартість, дата витрат; площа, кількість проживаючих осіб, адреса приміщень; площа, тип, призначення земельних ділянок; транспортний засіб (марка, державний номер, дата випуску), признак (призначена); дата початку і кінця виплати, розмір виплати	Оновлений запис про виплати відвідувачу (дата початку і кінця виплат, розмір виплати)

Продовження таблиці 1.1

Номер	Задача	Вхідні дані	Вихідні дані
С5	Призначити відвідувачу соціальні виплати по догляду за дітьми	Заява, декларація (ПБ членів родини, ступінь спорідненості, ППН, примітки; тип, розмір, джерело, період доходу; призначення, вартість, дата витрат; площа, кількість осіб, адреса приміщень; площа, тип, призначення земельних ділянок; марка, державний номер, дата випуску транспортних засобів), признак (призначена) дата початку і кінця виплат, розмір виплати	Оновлений запис про виплати відвідувачу (дата початку і кінця виплат, розмір виплати)
С6	Призначити відвідувачу соціальну компенсацію через наявність інвалідності	Заява, декларація (ПБ членів родини, ступінь спорідненості, ППН, примітки; тип, розмір, джерело та період доходу членів родини; призначення, вартість та дата витрат членів родини; площа, кількість проживаючих осіб, адреса приміщень членів родини; площа, тип та призначення земельних ділянок членів родини; марка, державний номер та дата випуску транспортних засобів членів родини), признак (призначена, довідка про групу інвалідності признак (призначена); дата початку і кінця виплат, розмір виплати	Оновлений запис про виплати відвідувачу (дата початку і кінця виплат, розмір виплати)

Продовження таблиці 1.1

Номер	Задача	Вхідні дані	Вихідні дані
C7	Призначити відвідувачу соціальну компенсацію через опіку над людиною з інвалідністю	Заява, декларація (ПІБ членів родини, ступінь спорідненості, ППН, примітки; тип, розмір, джерело та період доходу членів родини; призначення, вартість та дата витрат членів родини; площа, кількість проживаючих осіб, адреса приміщень у власності членів родини; площа, тип та призначення земельних ділянок у власності членів родини; марка, державний номер та дата випуску транспортних засобів членів родини), признак (призначена), признак (призначена), довідка про групу інвалідності підопічного; дата початку і кінця виплат, розмір виплати	Оновлений запис про виплати відвідувачу (дата початку і кінця виплат, розмір виплати)
C8	Скасувати соціальну виплату відвідувача	Виплата, коментар, признак (скасована), дата скасування	Оновлений запис про виплати відвідувачу
C9	Переглянуті інформацію про соціальні виплати відвідувача	Відвідувач	Усі соціальні виплати (тип, розмір, дата початку та кінця виплати, признак)
C10	Переглянуті інформацію про документи, надані відвідувачем	Відвідувач	ПІБ, дата народження, дані паспорту, адреса ,номер мобільного телефону, ідентифікаційний код, заява відвідувача, довідки відвідувача
C11	Призначити відвідувача іншому співробітнику	Відвідувач, співробітник	Оновлений запис про відвідувача

Продовження таблиці 1.1

Номер	Задача	Вхідні дані	Вихідні дані
C12	Зафіксувати номери паперових документів	Довідка/декларація, номер документу, дата видачі	Оновлена інформація про довідку
	Відвідувач		
B1	Отримати інформацію про актуальні призначені виплати	Відвідувач, дата	Розмір виплати, тип, період, признак (призначена)
B2	Переглянути історію виплат	Відвідувач, період	Розмір виплати, тип, період, признак (призначена чи скасована)
B3	Заповнити заяву на отримання соціальної допомоги	Дата подання заяви, відвідувач, тип соціальної допомоги	Нова заява
B4	Заповнити декларацію про доходи та витрати	Члени родини, проживаючі з відвідувачем (ПІБ, ступінь спорідненості, ІДН), інформація про їх доходи та витрати за період 12 місяців, нерухомість (площа, кількість проживаючих осіб, адреса), засоби пересування (марка автомобіля, державний номер, рік випуску), відвідувач	Нова декларація
B5	Створити довідку про стан здоров'я	Група інвалідності, дата останнього дослідження, дата початку і дата закінчення довідки, коментар, відвідувач	Нова довідка про стан здоров'я
B6	Створити довідку про опіку над дитиною	ПІБ дитини, дата народження дитини, відвідувач	Нова довідка про опіку над дитиною
B7	Створити довідку про опіку над особою з інвалідністю	ПІБ особи, група інвалідності особи, дата народження особи, відвідувач	Нова довідка про опіку над особою з інвалідністю
B8	Видалити документ	Довідка/декларація	Відсутні

1.2 Огляд існуючої системи для автоматизації процесу отримання соціальної допомоги

Далі необхідно провести огляд існуючих застосунків для автоматизації процесу отримання соціальної допомоги у дистанційному режимі. Основними вимогами для такого застосунку є: можливість отримати допомогу у дистанційному режимі, без постійного відвідування відповідних державних структур, та повнота доступних послуг, тобто можливість оформити допомогу на якомога більшу кількість типів соціальної допомоги. В Україні лише застосунок «Дія» [3], створений Міністерством цифрової трансформації, надає подібні послуги громадянам. Але він має декілька проблем технічного характеру: застосунок, покликаний оптимізувати взаємодію громадян з бюрократичними процесами у державі, об'єднує в собі багато функцій, що обумовлює часту нестабільність застосунку по причині великої кількості запитів; крім того, застосунок дозволяє оформляти онлайн допомогу лише у декількох випадках, серед яких оформлення допомоги по безробіттю, народженню дитині або пошуку муніципальної няні. В усіх інших випадках для отримання відповідних послуг громадяни повинні відвідувати Центри надання адміністративних послуг або Управління соціального захисту. Потрібно також зазначити, що «Дія» не зберігає дані, а лише підтягує їх з відповідних реєстрів, що обумовлює безпеку даних користувачів навіть у випадку зламу застосунку злочинцями [4].

Підсумовуючи вищесказане, можна зазначити, що існуючий програмний продукт має певні недоліки, і, таким чином, існує необхідність у створенні окремого застосунку для оформлення соціальної допомоги громадянам, який, зокрема, дозволить охопити більшу кількість випадків, за яких користувач зможе отримати соціальну допомогу від держави.

1.3 Висновок до розділу 1

Аналіз існуючого застосунку для автоматизації процесу отримання соціальної допомоги у дистанційному режимі виявив ряд проблем, пов'язаних з використанням цього застосунку. До найбільш поширених проблем, що виникають при його використанні, можна віднести нестабільність роботи через перевантаженість застосунку іншими функціями, а також підтримку вкрай обмеженої кількості випадків, у яких передбачається надання громадянину соціальної допомоги.

Доцільність реалізації розроблюваної інформаційної системи ґрунтується на можливості вирішення вищезазначених проблем завдяки:

- 1) створенню вузькоспеціалізованого онлайн-ресурсу, який охоплюватиме більшу кількість випадків, за яких громадянин може отримати допомогу від держави (допомогу малозабезпеченим сім'ям, сім'ям з дітьми, інвалідам I, II та III групи, на догляд особи з інвалідністю);

- 2) значному пришвидшенню процесів самостійного подання заяви на нарахування соціальної допомоги і надання довідок про її отримання (неотримання), що спрощує життя пересічних громадян та економить їхній час.

2 ПРОЕКТУВАННЯ ІНФОРМАЦІЙНОЇ СИСТЕМИ УПРАВЛІННЯ СОЦІАЛЬНОГО ЗАХИСТУ НАСЕЛЕННЯ

Ідея створення програмного продукту та його проект представлені на дев'ятнадцятій всеукраїнській конференції студентів і молодих науковців «Інформатика, інформаційні системи та технології», тези доповіді опубліковані [5].

2.1 Архітектура та шаблон проектування

Для розробки ІС обрано трирівневу архітектуру клієнт-сервер, яка складається з трьох рівнів: рівня інтерфейсу користувача, рівня сервера застосунків та рівня бази даних. Розмежування рівнів програмного забезпечення трирівневої архітектури продемонстровано на рис. 2.1.

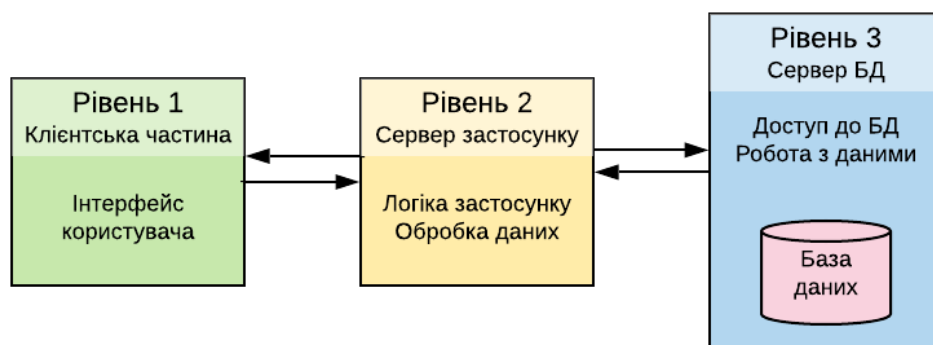


Рисунок 2.1 – Трирівнева архітектура клієнт-сервер

Трирівнева архітектура має наступні переваги [6]:

- можливість змінювати функціональні особливості одного рівня, при цьому не впливаючи на програмне забезпечення інших рівнів;
- надійність і незалежність роботи програмного забезпечення різних рівнів одне від одного;
- можливість масштабувати програмний продукт, додавати інші модулі, при цьому не змінюючи вже існуючі;

– різні розробники можуть незалежно працювати над програмним забезпеченням різних рівнів.

Для розробки інформаційної системи використаний шаблон проектування MVVM (рис. 2.2). Цей шаблон більше підходить до розроблюваної ІС, оскільки у порівнянні з шаблонами проектування MVC та MVP він має такі переваги [7]:

- MVVM абстрагує представлення і таким чином зменшує кількість бізнес-логіки, необхідної в коді, що стоїть за представленням;
- MVVM полегшує паралельну розробку інтерфейсу користувача та класів, які його забезпечують.

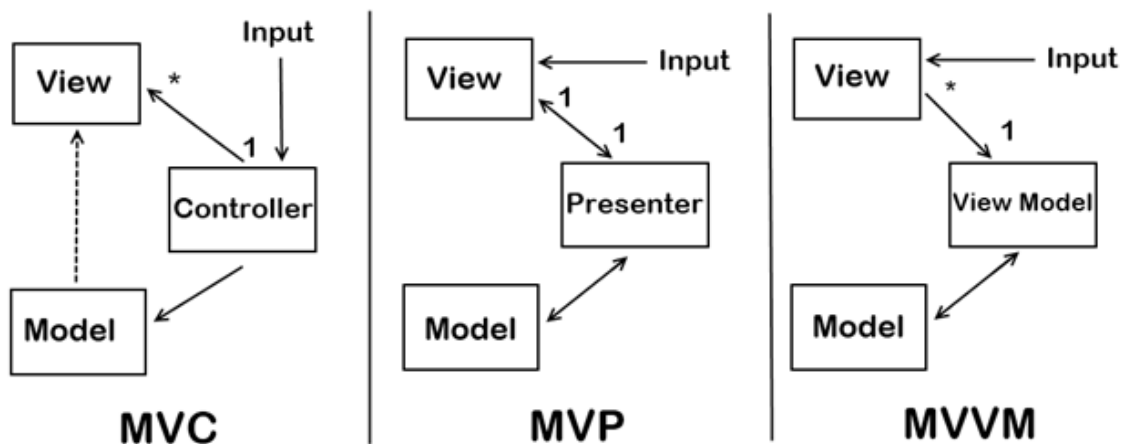


Рисунок 2.2 – Порівняння шаблонів MVC, MVP та MVVM

При використанні шаблону MVVM застосовується зв'язування даних з представлення-моделі з елементами самого представлення. Завдяки механізму зв'язування, представлення та представлення-модель при роботі програми напрямую обмінюються даними один з одним без контролерів – передаючи дані від користувача через представлення до представлення-моделі з послідуною передачею даних до моделі.

2.2 Інформаційне моделювання предметної області

Наступний етап проектування ІС – проектування інформаційної моделі предметної області, що розглядається.

Опис виділених сутностей, їх атрибутів і обмежень цілісності представлений у табл. 2.1.

Таблиця 2.1 – Опис сутностей, їхніх атрибутів та обмежень цілісності

Ім'я атрибута	Призначення атрибута	Обмеження цілісності
childcertificate (довідка про опікунство над дитиною)		
certificate_no	Ідентифікатор довідки	первинний ключ
date_of_birth	Дата народження дитини	не порожнє, значення < current_date
f_name	Ім'я дитини	не порожнє
patronymic	По батькові дитини	не порожнє
l_name	Прізвище дитини	не порожнє
application_no	Ідентифікатор заяви	зовнішній ключ для зв'язку з сутністю application, не порожнє
document_no	Унікальний номер документу	унікальне
date_of_issue	Дата видачі	значення <= current_date
application (заява)		
application_no	Ідентифікатор заяви	первинний ключ
owner_no	Ідентифікатор заявника	зовнішній ключ для зв'язку з сутністю visitor, не порожнє
application_date	Дата подання заяви	не порожнє, значення <= current_date
application_subsidy_type	Тип субсидії	значення зі списку («income level», «child care», «disability grant», «disability care»), не порожнє
status	Статус субсидії	значення зі списку («filed», «approved», «rejected»), не порожнє
comment	Примітки	може бути порожнім

Продовження таблиці 2.1

Ім'я атрибута	Призначення атрибута	Обмеження цілісності
disabledpersoncertificate (довідка про опіку над людиною з інвалідністю)		
certificate_no	Ідентифікатор довідки	первинний ключ
guardian_no	Ідентифікатор опікуна	зовнішній ключ для зв'язку з visitor, не порожнє
date_of_birth	Дата народження	не порожнє, значення < current_date
f_name	Ім'я підопічної людини	не порожнє
patronymic	По батькові підопічної людини	не порожнє
l_name	Прізвище підопічної людини	не порожнє
disability_group	Група інвалідності	значення зі списку ('1','2','3'), не порожнє
application_no	Ідентифікатор заяви	зовнішній ключ для зв'язку з сутністю application, не порожнє
document_no	Унікальний номер документу	унікальне
date_of_issue	Дата видачі	значення <= current_date
healthcertificate (довідка про стан здоров'я)		
certificate_no	Ідентифікатор довідки	первинний ключ
disability_group	Група інвалідності	значення зі списку ('1','2','3'), не порожнє
last_inspection	Дата останнього огляду	не порожнє, значення < period_start
period_start	Дата початку дії довідки	не порожнє значення < period_end
period_end	Дата закінчення дії довідки	не порожнє, значення > period_start
comment	Коментар до довідки	може бути порожнім
owner_no	Ідентифікатор власника	зовнішній ключ для зв'язку з сутністю visitor, не порожнє

Продовження таблиці 2.1

Ім'я атрибута	Призначення атрибута	Обмеження цілісності
document_no	Унікальний номер документа	Унікальне
date_of_issue	Дата видачі	значення $\leq$ current_date
staff (співробітник)		
staff_no	Ідентифікатор співробітника	первинний ключ
f_name	Ім'я співробітника	не порожнє
patronymic	По батькові співробітника	не порожнє
l_name	Прізвище співробітника	не порожнє
position	Посада	значення зі списку ('specialist', 'senior specialist', 'director'), не порожнє
login	Логін для входу	унікальне, не порожнє
visitor (відвідувач)		
visitor_no	Ідентифікатор відвідувача	первинний ключ
f_name	Ім'я відвідувача	не порожнє
patronymic	По батькові відвідувача	не порожнє
l_name	Прізвище відвідувача	не порожнє
date_of_birth	Дата народження	не порожнє, значення $<$ current_date
assigned_to	Ідентифікатор співробітника	зовнішній ключ для зв'язку з сутністю staff, не порожнє
passport_no	Номер паспорту	не порожнє, значення = 9 цифрових символів
idn	ПІН відвідувача	не порожнє, значення = 12 цифрових символів
login	Ідентифікатор користувача	зовнішній ключ для зв'язку з сутністю users, не порожнє

Продовження таблиці 2.1

Ім'я атрибута	Призначення атрибута	Обмеження цілісності
mobile	Мобільний телефон відвідувача	значення = 10 числових символів, не порожнє
address	Адреса відвідувача	не порожнє
subsidy (субсидія)		
payment_no	Ідентифікатор виплати	первинний ключ
appointed_by	Ідентифікатор співробітника	зовнішній ключ для зв'язку з сутністю staff, не порожнє
payment_amount	Сума допомоги	не порожнє, значення > 0.00
type	Тип виплати	значення зі списку («income level», «child care», «disability grant», «disability care»), не порожнє
period_start	Дата початку допомоги	не порожнє, значення >= current_date
period_end	Дата закінчення допомоги	не порожнє, значення > period_start
status	Статус субсидії	значення зі списку («active», «finished», «canceled»), не порожнє
application_no	Ідентифікатор заяви	зовнішній ключ для зв'язку з сутністю application, не порожнє
comment	Коментар	може бути порожнім
users (користувачі)		
login	Ім'я облікового запису користувача	не порожнє, унікальне
password	Пароль користувача	не порожнє, зберігається у вигляді хешу
category	Категорія користувача	значення зі списку («visitor», «administrator», «employee»), не порожнє
incomepropertydeclaration (декларація про доходи та витрати)		
declaration_no	Ідентифікатор декларації	первинний ключ
application_no	Ідентифікатор заяви	зовнішній ключ для зв'язку з сутністю application, не порожнє
document_no	Унікальний номер документу	унікальне
date_of_issue	Дата видачі	значення <= current_date

Продовження таблиці 2.1

Ім'я атрибута	Призначення атрибута	Обмеження цілісності
familymemberscertificate (члени родини відвідувача)		
record_no	Ідентифікатор запису	первинний ключ
declaration_no	Ідентифікатор декларації	зовнішній ключ для зв'язку з сутністю incomepropertydeclaration, не порожнє
f_name	Ім'я члена родини	не порожнє
patronymic	По батькові члена родини	не порожнє
l_name	Прізвище члена родини	не порожнє
degree_relationship	Ступінь спорідненості	не порожнє
idn	ПІН члена родини	не порожнє, значення = 12 цифр
comment	Примітки	може бути порожнім
incomefamily (доходи членів родини)		
record_no	Ідентифікатор запису	первинний ключ
member_no	Ідентифікатор члена родини	зовнішній ключ для зв'язку з сутністю familymemberscertificate, не порожнє
income_type	Тип доходу	значення зі списку («salary», «pension», «allowance», «entrepreneurship»), не порожнє
amount	Розмір доходу	не порожнє, значення > 0.00
source_of_income	Джерело доходу	не порожнє
income_start	ПІН члена родини	не порожнє, значення < current_date
income_finish	Примітки	не порожнє, значення > income_start
landfamily (земельні ділянки у власності членів родини)		
record_no	Ідентифікатор запису	первинний ключ
member_no	Ідентифікатор члена родини	зовнішній ключ для зв'язку з сутністю familymemberscertificate, не порожнє
area	Площа	не порожнє, значення > 0.0
type	Тип	значення зі списку («private», «communal», «state»), не порожнє
purpose	Призначення	порожнє
propertyfamily (житлові об'єкти у власності членів родини)		
record_no	Ідентифікатор запису	первинний ключ

Продовження таблиці 2.1

Ім'я атрибута	Призначення атрибута	Обмеження цілісності
member_no	Ідентифікатор члена родини	зовнішній ключ для зв'язку з сутністю familymemberscertificate, не порожнє
area	Площа	не порожнє, значення > 0.0
person_num	Кількість проживаючих осіб	не порожнє, значення >= 0
address	Адреса	не порожнє
spendingfamily (витрати членів родини)		
record_no	Ідентифікатор запису	первинний ключ
member_no	Ідентифікатор члена родини	зовнішній ключ для зв'язку з сутністю familymemberscertificate, не порожнє
spending_purpose	Призначення	не порожнє
cost	Вартість	не порожнє, значення > 0.00
date_of_purchase	Дата	не порожнє, значення < current_date
vehiclesfamily (транспортні засоби у власності членів родини)		
record_no	Ідентифікатор запису	первинний ключ
member_no	Ідентифікатор члена родини	зовнішній ключ для зв'язку з сутністю familymemberscertificate, не порожнє
car_brand	Марка	не порожнє
car_license_plate	Державний номер	не порожнє
year_of_issue	Дата випуску	не порожнє, значення < current_date

Між сутностями присутні 2 види зв'язку [8]: один-до-одного та один-до-багатьох. Зв'язок один-до-одного реалізовано між наступними таблицями.

1) Visitor і Users – кожному логіну відвідувача відповідає один, і тільки один логін з Users. Формалізація зв'язку: до таблиці Visitor винесено первинний ключ таблиці Users (login) у якості зовнішнього ключа.

2) Staff і Users – кожному імені облікового запису відповідає одне, і тільки одне ім'я облікового запису з Users. Формалізація зв'язку: до таблиці Staff винесено первинний ключ таблиці Users (login) у якості зовнішнього ключа.

3) Application і Subsidy – одній заяві може відповідати жодна або одна субсидія, при цьому субсидія не може існувати без заяви. Формалізація зв'язку: первинний ключ таблиці Application (application_no) додається в якості зовнішнього ключа в схему таблиці Subsidy (application_no).

4) HealthCertificate і Application – кожній заяві може відповідати тільки одна або жодна довідка про стан здоров'я. Формалізація зв'язку: первинний ключ таблиці Application (application_no) додається в якості зовнішнього ключа в схему таблиці HealthCertificate (application_no).

5) HealthCertificate і Application – кожній заяві може відповідати тільки одна або жодна довідка про стан здоров'я. Формалізація зв'язку: первинний ключ таблиці Application (application_no) додається в якості зовнішнього ключа в схему таблиці HealthCertificate (application_no).

Зв'язок один-до-багатьох реалізовано між наступними таблицями.

1) Visitor і Subsidy – одному відвідувачу може відповідати безліч субсидій, так як в таблиці містяться дані і про ті виплати, які були завершені або припинені, при цьому субсидія не може існувати без відвідувача, якому вона призначена. Формалізація зв'язку: первинний ключ таблиці Visitor (visitor_no) додається в якості зовнішнього ключа в схему таблиці Subsidy (visitor_no).

2) Visitor і ChildCertificate – одному відвідувачу може відповідати кілька довідок про опікунство над дитиною, при цьому довідка не може існувати без відвідувача, якому вона призначена. Формалізація зв'язку: первинний ключ таблиці Visitor (visitor_no) додається в якості зовнішнього ключа в схему таблиці ChildCertificate (guardian_no).

3) Staff і Visitor – кожному співробітнику може відповідати кілька відвідувачів, при цьому відвідувач обов'язково має закріпленого за ним співробітника. Формалізація зв'язку: первинний ключ таблиці Staff (staff_no) додається в якості зовнішнього ключа в схему таблиці Visitor (assigned_to).

4) Staff і Subsidy – кожному співробітнику може відповідати кілька виплат, при цьому виплата обов'язково має співробітника, який її призначив. Формалізація зв'язку: первинний ключ таблиці Staff (staff_no) додається в якості зовнішнього ключа в схему таблиці Subsidy (appointed_by).

5) Visitor і Application – одному відвідувачу може відповідати кілька заяв на отримання соціальної допомоги, при цьому заява не може існувати без відвідувача, який її подав. Формалізація зв'язку: первинний ключ таблиці Visitor (visitor_no) додається в якості зовнішнього ключа в схему таблиці Application (owner_no).

6) IncomePropertyDeclaration і Visitor – кожному відвідувачу може відповідати жодна або декілька декларацій про доходи та витрати. Формалізація зв'язку: первинний ключ таблиці Visitor (visitor_no) додається в якості зовнішнього ключа в схему таблиці IncomePropertyDeclaration (owner_no).

7) DisabledPersonCertificate і Application – кожній заяві може відповідати жодна або декілька довідок про опіку над людиною з інвалідністю. Формалізація зв'язку: первинний ключ таблиці Application (application_no) додається в якості зовнішнього ключа в схему таблиці DisabledPersonCertificate (application_no).

8) ChildCertificate і Application – кожній заяві може відповідати жодна або декілька довідок про опіку над дитиною. Формалізація зв'язку: первинний ключ таблиці Application (application_no) додається в якості зовнішнього ключа в схему таблиці ChildCertificate (application_no).

9) IncomePropertyDeclaration і FamilyMembersCertificate – кожній декларації може відповідати один або декілька записів про членів родини відвідувача. Формалізація зв'язку: первинний ключ таблиці IncomePropertyDeclaration (declaration_no) додається в якості зовнішнього ключа в схему таблиці FamilyMembersCertificate (declaration_no).

10) FamilyMembersCertificate і IncomeFamily – кожному запису про члена родини відвідувача може відповідати один або декілька записів про його доходи. Формалізація зв'язку: первинний ключ таблиці FamilyMembersCertificate (record_no) додається в якості зовнішнього ключа в схему таблиці IncomeFamily (member_no).

11) FamilyMembersCertificate і SpendingFamily – кожному запису про члена родини відвідувача може відповідати один або декілька записів про витрати та їх призначення. Формалізація зв'язку: первинний ключ таблиці FamilyMembersCertificate (record_no) додається в якості зовнішнього ключа в схему таблиці SpendingFamily (member_no).

12) FamilyMembersCertificate і PropertyFamily – кожному запису про члена родини відвідувача може відповідати один або декілька записів про об'єкти нерухомості у його власності. Формалізація зв'язку: первинний ключ таблиці FamilyMembersCertificate (record_no) додається в якості зовнішнього ключа в схему таблиці PropertyFamily (member_no).

13) FamilyMembersCertificate і LandFamily – кожному запису про члена родини відвідувача може відповідати один або декілька записів про земельні ділянки у його власності. Формалізація зв'язку: первинний ключ таблиці FamilyMembersCertificate (record_no) додається в якості зовнішнього ключа в схему таблиці LandFamily (member_no).

14) FamilyMembersCertificate і VehiclesFamily – кожному запису про члена родини відвідувача може відповідати один або декілька записів про належні йому транспортні засоби. Формалізація зв'язку: первинний ключ таблиці FamilyMembersCertificate (record_no) додається в якості зовнішнього ключа в схему таблиці VehiclesFamily (member_no).

ER-діаграма для предметної області «Управління соціального захисту населення» приведена на рис. 2.1. Для побудови ER-діаграми використаний програмний продукт Navicat.

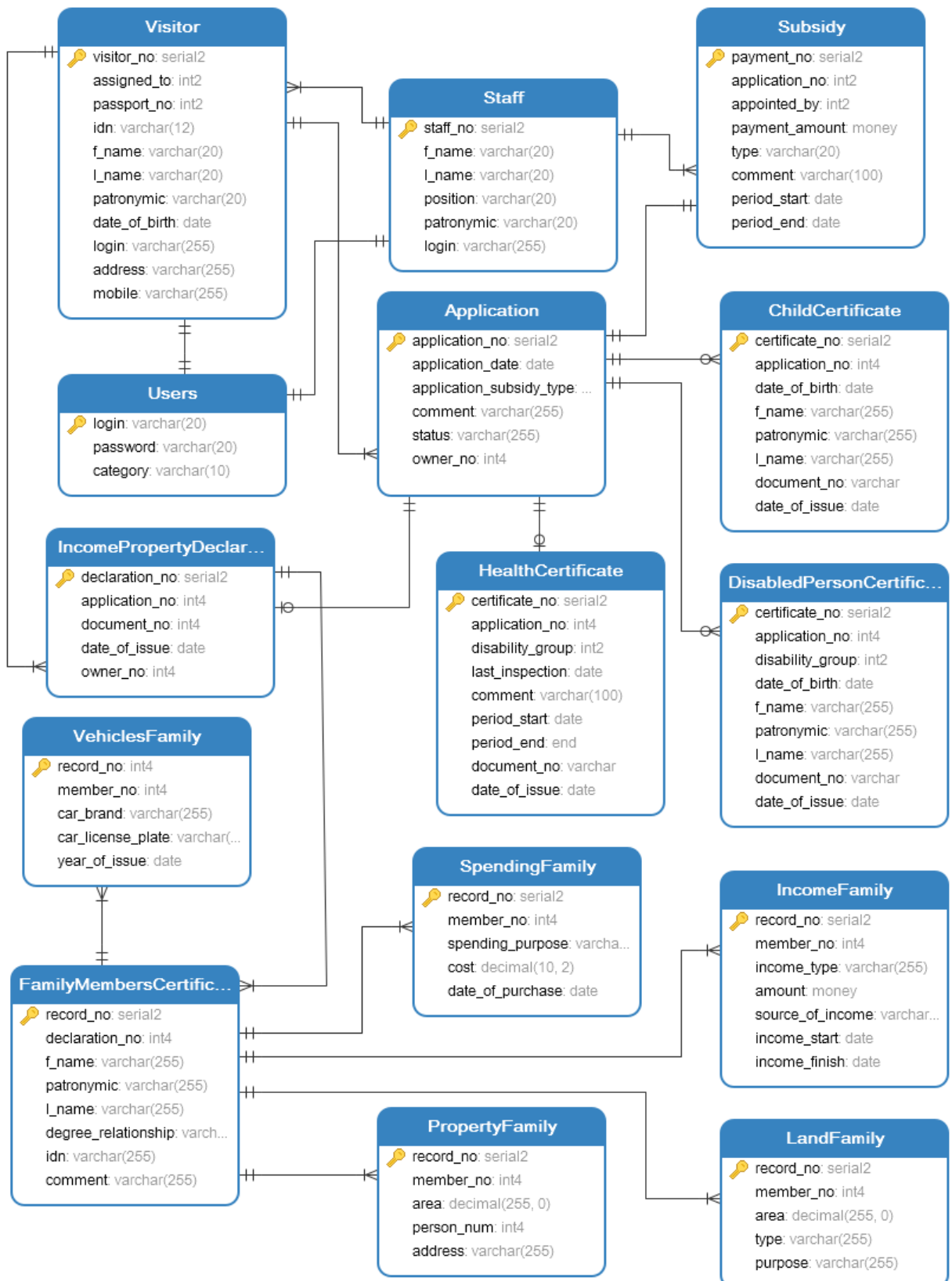


Рисунок 2.1 – ER-діаграма БД управління соціального захисту населення

2.3 Модель клієнтського застосунку

Ієрархію веб-сторінок застосунку зображено на рис. 2.2. Перехід з форми авторизації відбувається на основі даних, введених до форм адміністратора, співробітника та відвідувача.

Форма адміністратора дозволяє користувачеві створювати, редагувати та видаляти облікові записи співробітників управління. Форма співробітника дає користувачеві можливість працювати з обліковими записами користувачів (створювати нові, редагувати вже існуючі та призначати користувача іншому співробітнику) та з субсидіями (переглядати історію виплат певного користувача, переглядати його декларації та довідки, а також призначати йому виплати на підставі заяви). Форма відвідувача дозволяє користувачеві переглядати історію власних виплат, шукати виплати по датах та статусу, а також заповнювати заяви, декларації та довідки.

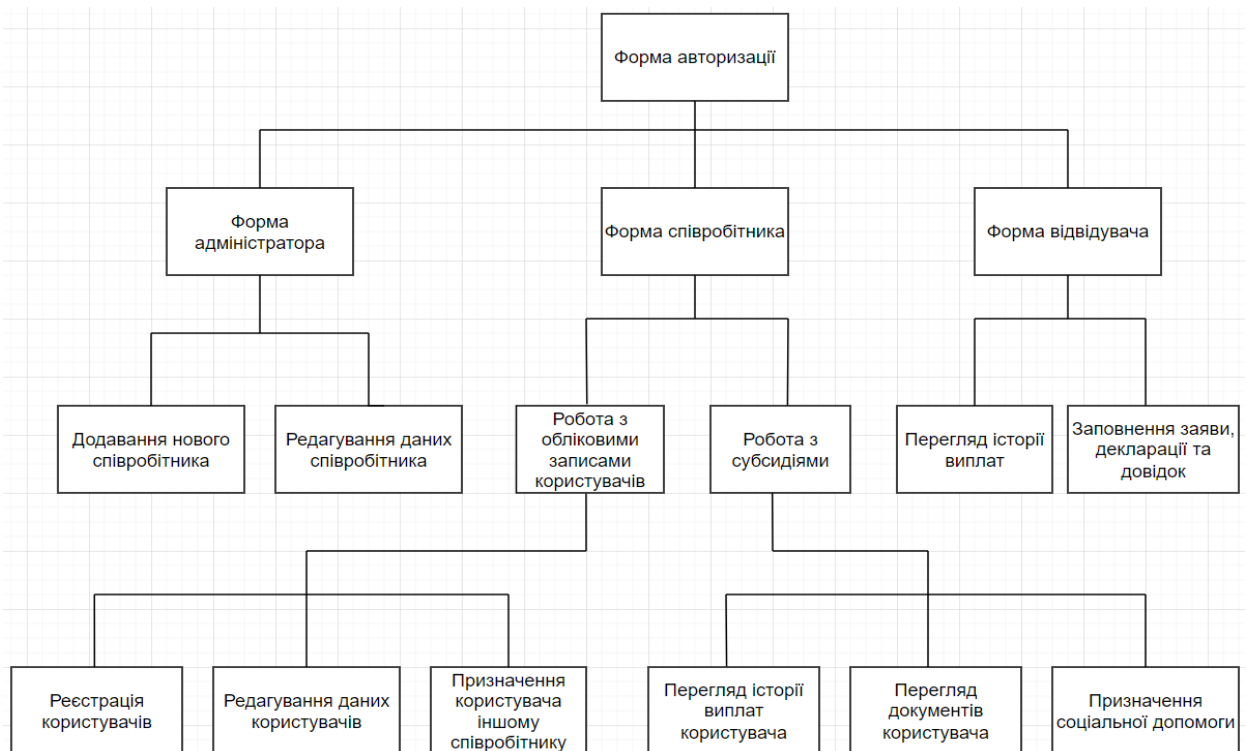


Рисунок 2.2 – Ієрархія екранних форм ІС управління соціального захисту населення

З ВИБІР ПРОГРАМНОГО ЗАБЕЗПЕЧЕННЯ ДЛЯ СТВОРЕННЯ ІНФОРМАЦІЙНОЇ СИСТЕМИ УПРАВЛІННЯ СОЦІАЛЬНОГО ЗАХИСТУ НАСЕЛЕННЯ

В якості засобів реалізації ІС обрані наступні інструменти:

- СУБД PostgreSQL;
- мова програмування C#;
- середовище розробки Visual Studio 2022 Professional Edition;
- модулі інтерфейсу ASP.NET Core Razor Pages;
- об'єктно-орієнтована технологія ADO.NET Entity Framework;
- бібліотека класів Npgsql для роботи з СУБД PostgreSQL.

PostgreSQL є проектом з відкритим кодом та можливістю вільного поширення, завдяки чому дана СУБД придбала велику спільноту користувачів та розробників, які займаються виправленням помилок і створенням нових версій. Тому, PostgreSQL по функціональності і зручності використання не поступається багатьом іншим комерційним СУБД.

Однією з сильних сторін PostgreSQL є її архітектура. Як і багато комерційних СУБД, PostgreSQL може застосовуватися в середовищі клієнт-сервер, що дає масу переваг розробникам та користувачам. Ще однією перевагою PostgreSQL є розширювана система вбудованих мов програмування, що дозволяє використовувати найбільш поширені МП для написання функцій на базі PostgreSQL [9].

Microsoft Visual Studio — це інтегроване середовище розробки від компанії Microsoft. Воно використовується для багатьох задач, зокрема розробки різних комп'ютерних програм, веб-сайтів, а також додатків для мобільних платформ. Visual Studio використовує платформи розробки програмного забезпечення Microsoft, такі як Windows API, Windows Forms та багато інших. Середовище підтримує 36 різних мов програмування і дозволяє редактору коду підтримувати майже будь-яку мову

програмування. Великою перевагою Visual Studio є можливість вільного використання та велика спільнота розробників.

C# — це універсальна мова, підтримує концепцію об'єктно-орієнтованого програмування та має велику кількість вбудованих функцій та додаткових бібліотек, які значно спрощують створення проектів різного напрямку, зокрема веб-систем та інформаційних систем, заснованих на використанні баз даних.

Razor Pages — це модель програмування веб-додатків, яка спрощує процес взаємодії представлення з моделлю. Такий підхід дозволив усунути більшу частину труднощів, які виникали при роботі з ASP.NET MVC, приймаючи підхід до маршрутизації на основі файлів. Кожен файл Razor Pages, знайдений у каталозі Pages, прирівнюється до кінцевої точки. Кожна сторінка має пов'язаний з нею об'єкт з розширенням .cs, який називається моделлю сторінки і відповідає за контроль поведінки кожної сторінки.

Entity Framework — це платформа ORM з відкритим вихідним кодом для застосунків .NET, яку підтримує Microsoft. За допомогою Entity Framework розробники можуть працювати на більш високому рівні абстракції, коли вони мають справу з даними, а також можуть створювати та підтримувати орієнтовані на дані програми з меншою кількістю коду в порівнянні з традиційними програмами [10].

4 СТВОРЕННЯ ІНФОРМАЦІЙНОЇ СИСТЕМИ УПРАВЛІННЯ СОЦІАЛЬНОГО ЗАХИСТУ НАСЕЛЕННЯ

4.1 Створення бази даних

При створенні бази даних знадобилось створити не тільки таблиці, а й власні типи даних, зокрема: `subsidy_type` для визначення типу субсидії, `application_status` для визначення статусу заяви, `category` для визначення категорії користувача, `position_staff` для визначення категорії співробітника, `subsidy_status` для визначення статусу субсидії, `income_type` для визначення типу доходу та `ownership_type` для визначення типу власності земельної ділянки.

Нижче представлений SQL-запит на створення одного з типів - `subsidy_type`, який необхідний для обмеження типів субсидії – виплата та компенсація:

```
create type subsidy_type as enum ('income level', 'child care',
'disability grant',' disability care')
```

У лістингу 4.1 розглянуто створення таблиці `Subsidy`. При створенні таблиці використовується власний тип `subsidy_type`, який визначає тип субсидії, призначений відвідувачеві. Поле `payment_no` є первинним ключом (`primary key`) і має тип `serial`. Зовнішні ключі `foreign key` (`application_no`), зв'язують таблицю `Subsidy` з таблицею `application` за номером заяви. Зовнішній ключ `foreign key` (`visitor_no`) зв'язує таблицю `Subsidy` з таблицею `Visitor` – отримувачем субсидії. Зовнішній ключ `foreign key` (`appointed_by`) зв'язує таблицю `Subsidy` з таблицею `Staff`, та вказує на співробітника, який призначив субсидію відвідувачу. При створенні запису працює перевірка поля `payment_amount` `money not null check(payment_amount::decimal>0.00)`, яка не допускає внесення від'ємного значення суми виплати.

```

create table Subsidy (
payment_no serial primary key,
visitor_no integer not null,
application_no integer not null,
appointed_by integer not null,
payment_amount money not null
check(payment_amount::decimal>0.00),
"type" subsidy_type not null,
period_start date not null,
period_end date not null,
comment varchar(100),
foreign key (visitor_no) references Visitor(visitor_no) on
delete cascade on update cascade,
foreign key (appointed_by) references Staff(staff_no) on delete
restrict on update cascade),
foreign key (application_no) references
Application(application_no) on delete restrict on update cascade);

```

Лістинг 4.1 – Створення таблиці Subsidy

Повний код створення бази даних наведений у додатку А.

4.2 Запити до бази даних для вирішення поставлених задач

Для вирішення задач А1 і С1 виконуються запити, наведені у лістингу 4.2.

```

create role {staffMember.login} with login password \'employee\';
grant employee to {staffMember.login}; " +
insert into users
values(\'{staffMember.login}\',\'{user.password}\',\'employee\');
update users set \'password\' = \'password\' where login =
\'{staffMember.login}\';
update users set \'password\' = \'{user.password}\' where login
= \'{staffMember.login}\';
insert into staff(f_name, patronymic, l_name, position, login)
values(\'{staffMember.f_name}\',\'{staffMember.patronymic}\',
\'{staffMember.l_name}\',\'{staffMember.position}\',
\'{staffMember.login}\');

```

Лістинг 4.2 – Додавання нових користувачів

Для вирішення задач А2 і С2 виконуються запити, наведені у лістингу 4.3.

```
alter role {OldStaffLogin} rename to {staffMember.login};
update users set password = '{user.password}' where login =
'{OldStaffLogin}';
update users set login = '{staffMember.login}' where login =
'{OldStaffLogin}';
update staff set f_name='{staffMember.f_name}',
patronymic='{staffMember.patronymic}',
l_name='{staffMember.l_name}',
position = '{staffMember.position}' where staff_no = {id};
```

Лістинг 4.3 – Редагування даних користувачів

Для вирішення задач А3 і С3 виконуються запити, наведені у лістингу 4.4.

```
drop role назва_ролі;
delete from users where login in (select login from visitor
where visitor_no = номер_відвідувача);
```

Лістинг 4.4 – Видалення користувача

Для вирішення задач С4 – С7 виконується наступний запит:

```
insert into subsidy(visitor_no,appointed_by,payment_amount,
"type",period_start,period_end,"comment",status, application_no)
values (номер_відвідувача, номер_співробітника, сума виплати,
тип_виплати, дата_початку_виплат, дата_кінця_виплат, коментар,
'active');
```

Для вирішення задач С4 – С7 створено тригер active_subsidy_trigger, який при оновленні чи додаванні записів до таблиці subsidy викликає тригерну функцію subsidy_added(), яка перевіряє, що відвідувач немає активних виплат (тобто, він не буде мати кілька активних виплат водночас). Повний текст створення тригера можна знайти у Додатку Б (лістинг Б.2).

Для вирішення задачі С8 виконується наступний запит:

```
update subsidy set status = 'canceled' where payment_no =
номер_виплати;
```

Для вирішення задачі C9 виконується наступний запит:

```
select * from subsidy where visitor_no = номер_відвідувача;
```

Для вирішення задачі C10 виконуються запити, наведені у лістингу 4.5.

```
SELECT * FROM healthcertificate where owner_no = {Id};
SELECT * FROM childcertificate where owner_no = {Id};
SELECT * FROM disabledpersoncertificate where guardian_no = {Id};
select * from incomepropertydeclaration where owner_no = {Id};
```

Лістинг 4.5 – Перегляд наданих відвідувачем документів

Для вирішення задачі C11 виконується наступний запит:

```
update visitor
set assigned_to = номер_співробітника where visitor_no =
номер_відвідувача;
```

Для вирішення задачі C12 виконуються запити, наведені у лістингу 4.6.

```
update childcertificate
set document_no = номер_співробітника, date_of_issue =
дата_видачі
where owner_no = номер_відвідувача;
update disabledpersoncertificate
set document_no = номер_співробітника, date_of_issue =
дата_видачі
where owner_no = номер_відвідувача;
```

Лістинг 4.6 – Фіксування номерів і дат видачі документів

Для вирішення задачі B1 виконується наступний запит:

```
SELECT * FROM subsidy where visitor_no = {Id} and status =
({subsidy_status}::subsidy_status)
```

Для вирішення задачі B2 виконується наступний запит:

```
SELECT * FROM subsidy where visitor_no = {Id} and (period_start
between ({date_start.ToString("yyyy-MM-dd")}>::date
and {date_end.ToString("yyyy-MM-dd")}>::date or period_end
between {date_start.ToString("yyyy-MM-dd")}>::date
and {date_end.ToString("yyyy-MM-dd")}>::date)
```

Для вирішення задачі В3 виконується наступний запит:

```
insert into application(owner_no, application_date,
application_subsidy_type, status, \"comment\") values ('{id}',
\"{date}\", \"{application.application_subsidy_type}\", 'filed',
\"{application.comment}\");
```

Для вирішення задачі В4 виконуються запити, наведені у лістингу 4.7.

```
insert into incomefamily(member_no, income_type, amount,
source_of_income, income_start, income_finish) values({MemberId},
\"{incomeFamilyRecords[i].income_type}\",
\"{incomeFamilyRecords[i].amount}\",
\"{incomeFamilyRecords[i].source_of_income}\",
\"{incomeFamilyRecords[i].income_start.ToString("yyyy-MM-dd")}\",
\"{incomeFamilyRecords[i].income_finish.ToString
("yyyy-MM-dd")}\" );
insert into spendingfamily(member_no, spending_purpose, cost,
date_of_purchase) values\"({MemberId},
\"{spendingFamilyRecords[i].spending_purpose}\",
\"{spendingFamilyRecords[i].cost}\",
\"{spendingFamilyRecords[i].date_of_purchase.ToString
("yyyy-MM-dd")}\" );
```

Лістинг 4.7 – Оформлення нової декларації

Для вирішення задачі В5 виконується наступний запит:

```
insert into healthcertificate(last_inspection, period_start,
period_end, owner_no, disability_group, \"comment\")
values(\"{healthCertificate.last_inspection.ToString("yyyy-MM-
dd")}\", \"{healthCertificate.period_start.ToString("yyyy-MM-
dd")}\", \"{healthCertificate.period_end.ToString("yyyy-MM-
dd")}\", {Id}, \"{healthCertificate.disability_group}\",
\"{healthCertificate.comment}\");
```

Для вирішення задачі В6 виконується наступний запит:

```
insert into childcertificate(date_of_birth, f_name, patronymic,
l_name, owner_no) values(\'{childCertificate.date_of_birth}\',
\'{childCertificate.f_name}\', \'{childCertificate.l_name}\',
\'{Id}\')
```

Для вирішення задачі В7 виконується наступний запит:

```
insert into disabledpersoncertificate(date_of_birth, f_name,
patronymic, l_name, guardian_no, disability_group)
values(\'{disabledCertificate.date_of_birth.ToString("yyyy-MM-
dd")}\', \'{disabledCertificate.f_name}\',
\'{disabledCertificate.patronymic}\',
\'{disabledCertificate.l_name}\', '{Id}',
\'{disabledCertificate.disability_group}\')
```

4.3 Програмна реалізація інтерфейсу

У лістингу 4.8 розглянуто одну із типових операції в розробленому застосунку – підключення до БД.

```
public IndexModel(ApplicationDbContext db)
{
    _context = db;
}
public void OnGet()
{
    users = _context.users.AsNoTracking().ToList();
}
```

Лістинг 4.8 – Під’єднання до БД

У даному випадку програма отримує доступ до бази даних у змінну `_context`, з якого потім витягує значення таблиці `Users` у відповідну змінну-список. У лістингу 4.9 показано, як для цього у файлі проекту `appsetting.json` вказується стрічка з’єднання з БД.

```
{
  "Logging": {
    "LogLevel": {
      "Default": "Information",
      "Microsoft.AspNetCore": "Warning"
    }
  },
  "AllowedHosts": "*",
  "ConnectionStrings": {
    "DefaultConnection":
      "Host=localhost;Username=postgres;Password=gilbert;
      Database=DiplomaProject"
  }
}
```

Лістинг 4.9 – Налаштування файлу appsetting.json

Далі, при натисканні кнопки «Увійти до системи», система зчитує введене користувачем ім'я облікового запису і пароль, і намагається підключитись до БД від ролі з іменем «логін». У лістингу 4.10 продемонстровано, як при цьому використовується запит, який перетворює введений користувачем пароль у хеш-код, та порівнює його з паролем, який зберігається у БД.

```
public IActionResult OnPost(string login, string password){
  if (!ModelState.IsValid){
    return Page();
  }
  var user = _context.users.FromSqlInterpolated($"SELECT * FROM
  users where login={login}").SingleOrDefault();
  connection.Open();
  string sql = $"select
  convert_from(encode(digest('{password}':::bytea, 'sha256'),
  'base64')::bytea, 'utf-8')";
  var cmd = new NpgsqlCommand(sql, connection);
  string digestPassword = cmd.ExecuteScalar().ToString();
  connection.Close();
```

Лістинг 4.10 – Хешування паролю та його перевірка

```

if (user == null || digestPassword != user.password)
{
    Message = $"Введено неправильне ім'я облікового запису та/або
    пароль";
    return Page();
}
else if (digestPassword == user.password) {
    login = user.login;
    if (user.category == "employee")
    {
        return RedirectToPage("Employee/EmployeeStart", new { login });
    }
    else if (user.category == "visitor")
    {
        return RedirectToPage("Client/VisitorStart", new { login });
    }
    else if (user.category == "administrator")
    {
        return RedirectToPage("Admin/Administrator", new { login });
    }
}
return Page();
}

```

Лістинг 4.10, лист 2

Якщо система знайшла співпадіння серед імені та пароллю облікового запису та в залежності від категорії користувача, триває перехід від форми авторизації до однієї з трьох форм користувачів.

У лістингу 4.11 для прикладу роботи з БД розглянуто операцію додавання нового відвідувача. Співробітник управління має ввести дані відвідувача, а також логін та пароль від його персонального облікового запису в поля форми. Після цього він натискає на відповідну кнопку, після чого викликається функція додавання нового відвідувача.

```

public IActionResult OnPostAsync() {
    if (ModelState.IsValid) {
        connection.Open();
    }
}

```

Лістинг 4.11 – Реєстрація нового відвідувача

```

visitor.assigned_to = GetStaffNo(Login);
string date = visitor.date_of_birth.ToString("yyyy-MM-dd");
user.category = cat;
string sql = $"create role {visitor.login} with login password
\'{user.password}\';
grant visitor to {visitor.login};
insert into users
values(\'{visitor.login}\',\'{user.password}\',\'visitor\');
update users set \'password\' = \'password\' where login =
\'{visitor.login}\';
update users set \'password\' = \'{user.password}\' where login
= \'{visitor.login}\';
insert into visitor
(f_name,patronymic,l_name,date_of_birth,assigned_to,passport_no,
idn, login, mobile, address)
$"values(\'{visitor.f_name}\',\'{visitor.patronymic}\',\'{visito
r.l_name}\',\'{date}\',\'{visitor.assigned_to}\',
\'{visitor.passport_no}\',\'{visitor.idn}\',\'{visitor.login}\',
\'{visitor.mobile}\', \'{visitor.address}\');
var cmd = new NpgsqlCommand(sql, connection);
cmd.ExecuteNonQuery();
connection.Close();
return RedirectToPage("Employee", new { Login });
return Page();

```

Лістинг 4.11, лист 2

На основі зчитаних з кожного поля даних формується запит на додавання нового запису до таблиці Visitor. При натисканні кнопки відбувається спроба додати запис про нового відвідувача до БД, при цьому відбувається валідація даних за допомогою фреймворку Entity Framework та відображення відповідних повідомлень – наприклад, при внесенні даних про паспорт та ПІН, які мають виглядати як стрічка цифр, у вигляді текстових символів або відсутності даних у тих полях, які обов’язково мають бути заповнені. Програма обробляє повідомлення від моделі та виводить їх на форму, що вказує співробітнику на помилки внесених даних при створенні нового відвідувача.

4.4 Програмна реалізація представлення-моделі

Для реалізації застосунку створено ряд класів, що використовуються для опису існуючих у БД сутностей. Кожен клас описує атрибути кожної таблиці БД, які використовуються для роботи з даними при роботі застосунку. Для організації взаємодії БД зі застосунком також створено окремий клас `ApplicationContext`, атрибутами котрого є класи відповідних сутностей. На рисунку 4.1 представлена діаграма класів створеного застосунку.



Рисунок 4.1 – Діаграма класів створеного застосунку

4.5 Безпека інформаційної системи

Безпека на рівні БД забезпечується шляхом створення ролей і наділення їх відповідними привілеями. В інформаційній системі управління соціального захисту населення реалізовано три ролі: Адміністратор, Співробітник і Відвідувач. У табл. 4.1 наведені привілеї ролей на таблиці БД. При цьому використані наступні позначення: С (Create) – додавання даних, R (Read) – зчитування даних, U (Update) – оновлення даних, D (Delete) – видалення даних, E (Execute) – виконання функцій.

Таблиця 4.1 – Привілеї ролей на об'єкти БД

Об'єкти БД	Ролі		
	Адміністратор (administrator)	Співробітник (employee)	Відвідувач (visitor)
Таблиці			
HealthCertificate		R	CRUD
Application		R	CRD
DisabledPersonCertificate		R	CRUD
ChildCertificate		R	CRUD
Visitor		CRUD	R
Staff	CRUD	R	
Subsidy		CRUD	
Users	CRUD	CRUD	RU
FamilyMembersCertificate		R	CRD
IncomePropertyDeclaration		R	CRUD
LandFamily		R	CRD
PropertyFamily		R	CRD
IncomeFamily		R	CRD
SpendingFamily		R	CRD
VehiclesFamily		R	CRD

Продовження таблиці 4.1

Об'єкти БД	Ролі		
	Адміністратор (administrator)	Співробітник (employee)	Відвідувач (visitor)
Функції			
subsidy_added		E	
encrypt_password	E	E	E
subsidy_cancelling		E	

Для забезпечення безпеки збережених у БД паролів створено тригер `make_encrypted_password`, який при оновленні чи додаванні записів до таблиці `users` викликає тригерну функцію `encrypt_password()`, що виконує шифрування пароля користувача. Повний текст створення тригерної функції можна знайти у Додатку Б (лістинг Б.1).

Створення ролей користувачів ІС та надання їм привілеїв відповідно до табл. 4.1 продемонстровано у лістингу 4.12.

```
// Відвідувач
create role visitor;
grant select on visitor, disabledpersoncertificate,
childcertificate, healthcertificate, subsidy to visitor;
grant insert on disabledpersoncertificate, childcertificate,
healthcertificate, application, incomepropertydeclaration,
incomefamily, spendingfamily, vehiclesfamily, landfamily,
propertyfamily;
grant update on users;

// Співробітник
create role employee;
grant connect on database "DiplomaProject" to employee;
grant select on all tables in schema "public" to employee;
grant usage on schema "public" to employee;
grant insert on visitor, subsidy, users to employee;
grant update on visitor, subsidy, users to employee;
```

Лістинг 4.12 – Створення ролей та надання їм привілеїв

```
// Адміністратор
create role administrator;
grant insert, update, select, delete on all tables in schema
public to administrator;
grant execute on all functions in schema public to administrator;
```

Лістинг 4.12, лист 2

При спробі увійти до облікового запису система шукає у таблиці Users запис з відповідним логіном та паролем. У зв'язку з тим, що пароль зберігається у вигляді хешу, до БД надсилається SQL-запит, який шифрує введений користувачем пароль за допомогою модулю pgcrypto СУБД PostgreSQL, що продемонстровано нижче.

```
select convert_from(encode(digest(\'{password}\':::bytea,
'sha256'),'base64')::bytea, 'utf-8');
```

Після цього отриманий хеш порівнюється з тим, що зберігається у таблиці Users. При знаходженні співпадаючого логіну та хешу пароля, система зчитує категорію користувача зі стовпця category. В залежності від отриманого результату, до застосунку передається інформація про групову роль, від імені якої буде відбуватися наступне з'єднання з БД. Таким чином, користувач при з'єднанні з БД отримує відповідні привілеї відповідно лістингу 4.23, а застосунок надає користувачеві відповідний інтерфейс – стартову сторінку адміністратора, співробітника або відвідувача.

5 ІНСТРУКЦІЯ КОРИСТУВАЧА

Починаючи роботу з програмою, спочатку користувач системи бачить вікно автентифікації (рис. 5.1):

Управління праці та соціального захисту населення

**Введіть ім'я та пароль
облікового запису**

Ім'я

Пароль

Рисунок 5.1 – Вікно автентифікації

Коли користувач вводить свої дані, система перевіряє їх співпадіння з даними у БД. Якщо співпадіння не знайдено, на екран виводиться відповідне повідомлення (рис. 5.2).

Управління праці та соціального захисту населення

**Введено неправильне ім'я
облікового запису та/або
пароль**

Ім'я

Пароль

Рисунок 5.2 – Вікно автентифікації при введені некоректних даних

При автентифікації кожен користувач переходить на відповідну до своєї категорії веб-сторінку (рис. 5.3 – 5.5).



Облікові записи співробітників

Перейдіть на цю форму, щоб зареєструвати нового користувача або змінити дані вже зареєстрованого

[Перейти до роботи з обліковими записами користувачів](#)

Рисунок 5.3 – Початкова сторінка адміністратора



Облікові записи користувачів

Перейдіть на цю форму, щоб зареєструвати нового користувача або змінити дані вже зареєстрованого

[Перейти до роботи з обліковими записами користувачів](#)



Соціальна допомога

Перейдіть на цю форму, щоб почати роботу з виплатами користувачів

[Перейти до призначення соціальної допомоги](#)

Рисунок 5.4 – Початкова сторінка співробітника



Інформація про виплати

Перейдіть на цю форму, щоб отримати інформацію про свої активні та закінчені виплати

[Перегляд своїх виплат](#)



Оформлення допомоги

Перейдіть на цю форму, щоб оформити заяву на отримання соціальної допомоги

[Подача заяви](#)



Реєстрація документів

Перейдіть на цю форму, щоб внести інформацію про свої документи

[Оформлення документів](#)

Рисунок 5.5 – Початкова сторінка відвідувача

При переході на форму обліку співробітників, адміністратор може побачити усіх зареєстрованих співробітників (рис. 5.6).



Ім'я	По-батькові	Прізвище	Посада	Ім'я облікового запису		
Ольга	Іванівна	Прохорова	senior specialist	prohorova	Змінити дані	Видалити
Герман	Ігорович	Стерненко	specialist	germ_ster	Змінити дані	Видалити
Іван	Олександрович	Зась	senior specialist	zas_ivan	Змінити дані	Видалити

[Зареєструвати співробітника](#)

Рисунок 5.6 – Зареєстровані співробітники

При реєстрації нового відвідувача адміністратор переходить на наступну форму (рис. 5.7).

Ім'я
Олексій

По-батькові
Ігорович

Прізвище
Червоненко

Посада
спеціаліст ▼

Ім'я облікового запису:
chervonenko

Початковий пароль:
.....

[Зареєструвати співробітника](#)

Рисунок 5.7 – Реєстрація нового співробітника

При реєстрації нового співробітника адміністратор вводить ім'я, по-батькові, прізвище, обирає посаду нового співробітника, а також ім'я облікового запису та пароль. На рис. 5.8 показане вікно з обліковими записами співробітників.

Ім'я	По-батькові	Прізвище	Посада	Ім'я облікового запису		
Ольга	Іванівна	Прохорова	senior specialist	prohorova	Змінити дані	Видалити
Герман	Ігорович	Стерненко	specialist	germ_ster	Змінити дані	Видалити
Іван	Олександрович	Зась	senior specialist	zas_ivan	Змінити дані	Видалити
Олексій	Ігорович	Червоненко	specialist	chervonenko	Змінити дані	Видалити

Зареєструвати співробітника

Рисунок 5.8 – Оновлене вікно з обліковими записами співробітників

Далі розглядається інтерфейс співробітника.

Співробітник може створювати облікові записи відвідувачів та змінювати дані вже зареєстрованих записів, при цьому він бачить список лише тих відвідувачів, які закріплені за ним (рис. 5.9).

Ім'я	По-батькові	Прізвище	Дата народження	Паспорт	Адреса	ІДН	Моб. номер	Логін		
Іван	Генрихович	Древоронов	1997-05-19	093812485	Одеса, вул. Буніна, 32	093821122353	0972187716	drevoron1997	Змінити дані	Видалити
Роман	Романович	Романов	1995-01-13	384767376	Ізмаїл, пр. Миру, 58	384432162167	0913456681	roman	Змінити дані	Видалити
Олександр	Олександрович	Прохоров	2000-05-19	333455585	Одеса, вул. Кінна, 10	093842152353	0352747787	scott	Змінити дані	Видалити
Микита	Сергійович	Гєєвський	2003-05-30	095612485	Одеса, вул. Пастера 6	093841822085	0972167157	mykyta	Змінити дані	Видалити
Альберт	Олександрович	Грач	1998-02-04	093812411	Ізмаїл, пр. Миру, 23	093841822311	0972167711	grachalbert	Змінити дані	Видалити
Аніта	Викторівна	Андреева	1999-10-08	384767362	Ізмаїл, вул. Репіна, 13	384432162123	0877137419	anita	Змінити дані	Видалити

Зареєструвати відвідувача

Рисунок 5.9 – Вікно зареєстрованих відвідувачів

Окрім роботи з обліковими записами користувачів, співробітник може призначати виплати, розглядати заяви відвідувачів, переглядати історію їх виплат та призначати їх іншим співробітникам (рис. 5.10, 5.11).

Відвідувачі зі схваленими заявами

Ім'я	По-батькові	Прізвище	Дата народження	Паспорт	ІДН			
Микита	Сергійович	Геевський	2003-05-30	095612485	093841822085	Переглянути історію виплат	Призначити виплату	Призначити відвідувача іншому співробітнику

Відвідувачі з поданими заявами

Ім'я	По-батькові	Прізвище	Дата народження	Паспорт	ІДН			
Іван	Генрихович	Древоронов	1997-05-19	093812485	093821122353	Переглянути історію виплат	Розглянути заяву	Призначити відвідувача іншому співробітнику
Альберт	Олександрович	Грач	1998-02-04	093812411	093841822311	Переглянути історію виплат	Розглянути заяву	Призначити відвідувача іншому співробітнику

Усі призначені відвідувачі

Ім'я	По-батькові	Прізвище	Дата народження	Паспорт	ІДН		
Іван	Генрихович	Древоронов	1997-05-19	093812485	093821122353	Переглянути історію виплат	Призначити відвідувача іншому співробітнику
Роман	Романович	Романов	1995-01-13	384767376	384432162167	Переглянути історію виплат	Призначити відвідувача іншому співробітнику
Олександр	Олександрович	Прохоров	2000-05-19	333455585	093842152353	Переглянути історію виплат	Призначити відвідувача іншому співробітнику
Микита	Сергійович	Геевський	2003-05-30	095612485	093841822085	Переглянути історію виплат	Призначити відвідувача іншому співробітнику

Рисунок 5.10 – Вікно призначень соціальної допомоги

Активна виплата

Тип субсидії	Розмір виплати	Початок виплати	Кінець виплати	Статус	Примітки
--------------	----------------	-----------------	----------------	--------	----------

Скасовані та закінчені виплати

Тип субсидії	Розмір виплати	Початок виплати	Кінець виплати	Статус	Примітки
income level	1200.00	2022-05-19	2022-06-06	canceled	Canceled subsidy
child care	1500.00	2022-06-05	2022-06-06	canceled	-
child care	2600.00	2022-06-10	2022-06-10	canceled	активна виплата

Рисунок 5.11 – Історія виплат відвідувача

Далі описаний інтерфейс відвідувача.

Відвідувач може переглядати інформацію про призначені йому виплати наступним чином:

- шукаючи виплати по статусу (рис. 5.12);
- шукаючи виплати по періоду часу (рис. 5.13);
- отримувати дані про історію своїх виплат (рис. 5.14).

- ☒ active
- ☐ canceled
- ☐ finished

Пошук виплат за статусом

Тип субсидії	Розмір виплати	Початок виплати	Кінець виплати	Статус	Примітки
--------------	----------------	-----------------	----------------	--------	----------

Рисунок 5.12 – Пошук виплат по статусу (активна виплата відсутня)

Дата початку виплати

28.04.2022

Дата закінчення виплати

20.05.2022

Пошук виплат за датою

- ☐ active
- ☐ canceled
- ☐ finished

Пошук виплат за статусом

Тип субсидії	Розмір виплати	Початок виплати	Кінець виплати	Статус	Примітки
income level	1200.00	2022-05-19	2022-06-06	canceled	Canceled subsidy

Рисунок 5.13 – Пошук виплат за періодом часу (знайдено одну виплату)

Дата початку виплати

07.06.2022

Дата закінчення виплати

07.06.2022

Пошук виплат за датою

- ☐ active
- ☐ canceled
- ☐ finished

Пошук виплат за статусом

Тип субсидії	Розмір виплати	Початок виплати	Кінець виплати	Статус	Примітки
income level	1200.00	2022-05-19	2022-06-06	canceled	Canceled subsidy
child care	1500.00	2022-06-05	2022-06-06	canceled	-
child care	2600.00	2022-06-10	2022-06-10	canceled	активна виплата

Рисунок 5.14 – Вікно історії виплат відвідувача

Для того щоб отримати соціальну допомогу, відвідувач повинен ввести до системи необхідні дані: якнайменш заповнити декларацію про доходи та витрати, а також, якщо він претендує на відповідний тип допомоги, довідки про стан здоров'я, опіку над дитиною або людиною з інвалідністю (рис. 5.15).



Декларація про доходи та витрати

Перейдіть на цю форму, щоб заповнити декларацію про доходи та витрати

Оформлення декларації



Довідка про стан власного здоров'я

Перейдіть на цю форму, щоб внести інформацію про свої довідки по стану здоров'я

Оформлення довідки



Довідка про опіку над дитиною

Перейдіть на цю форму, щоб внести інформацію про довідки по опіці над дитиною

Оформлення довідки



Довідки про опіку над людиною з інвалідністю

Перейдіть на цю форму, щоб внести інформацію про довідки по опіці над людиною з інвалідністю

Оформлення довідки



Переглянути усі внесені довідки

Перейдіть на цю форму, щоб переглянути інформацію щодо внесених у систему даних

Перегляд документів

Рисунок 5.15 – Вікно реєстрації документів

Перш за все, відвідувач повинен заповнити декларацію про доходи та витрати. В цій декларації він має вказати членів своєї родини, їх доходи, витрати, земельні ділянки, об'єкти власності та транспортні засоби, що перебувають у власності членів родини (рис. 5.16, 5.17).

Заповнення декларації

Члени родини

Прізвище	Ім'я	По-батькові	Ступінь зв'язку	ИДН	Примітки
			Ступінь спорідненості ▼		
			Ступінь спорідненості ▼		
			Ступінь спорідненості ▼		
			Ступінь спорідненості ▼		
			Ступінь спорідненості ▼		

Доходи членів родини

Прізвище	Ім'я	По-батькові	Тип	Розмір	Джерело	З	До
			Тип доходу ▼			ДД.ММ.ГГГГ	ДД.ММ.ГГГГ
			Тип доходу ▼			ДД.ММ.ГГГГ	ДД.ММ.ГГГГ
			Тип доходу ▼			ДД.ММ.ГГГГ	ДД.ММ.ГГГГ
			Тип доходу ▼			ДД.ММ.ГГГГ	ДД.ММ.ГГГГ
			Тип доходу ▼			ДД.ММ.ГГГГ	ДД.ММ.ГГГГ

Рис. 5.16 – Члени родини та їхні доходи

Витрати членів родини

Прізвище	Ім'я	По-батькові	Призначення	Вартість	Дата
					дд.мм.гггг
					дд.мм.гггг
					дд.мм.гггг
					дд.мм.гггг
					дд.мм.гггг

Приміщення у власності членів родини

Прізвище	Ім'я	По-батькові	Площа	Кількість проживаючих осіб	Адреса

Рисунок 5.17 – Витрати членів родини та приміщення у їх власності

Крім декларації, відвідувач також може заповнити декілька типів довідок: довідку про стан власного здоров'я, про опіку над дитиною та опіку над людиною з інвалідністю (рис. 5.18).

Створення довідки

Дата останнього дослідження

Дата видачі довідки

Дата закінчення

Примітки

Група інвалідності

☐ I☐ II☐ III

Рисунок 5.18 – Створення довідки про стан власного здоров'я

Крім того, відвідувач може переглянути усі свої зареєстровані в системі документи (рис. 5.19, 5.20).

Довідки про стан здоров'я

Дата останнього дослідження	Дата видачі довідки	Дата закінчення	Група інвалідності	Примітки	Номер документу	Дата видачі довідки
-----------------------------	---------------------	-----------------	--------------------	----------	-----------------	---------------------

Довідки про опіку над дитиною

Ім'я	По-батькові	Прізвище	Дата народження	Номер документу	Дата видачі довідки
Андрій	Іванович	Древоронов	2020-06-03		Видалити довідку
Євген	Генріхович	Древоронов	2020-02-05		Видалити довідку

Довідки про опіку над людиною з інвалідністю

Ім'я	По-батькові	Прізвище	Дата народження	Група інвалідності	Номер документу	Дата видачі довідки
Генріх	Вальтерович	Древоронов	1952-01-02	II		Видалити довідку
Олександр	Ігорович	Кондратюк	1992-02-05	III		Видалити довідку

Рисунок 5.19 – Зареєстровані в системі довідки

Декларація про доходи та витрати

Члени родини					
Прізвище	Ім'я	По-батькові	Ступінь зв'язку	ИДН	Примітки
Древоронов	Олег	Олегович	II	093876122353	двоюрідний брат
Древоронов	Дмитрій	Генріхович	I	873876422353	рідний брат

Доходи членів родини							
Прізвище	Ім'я	По-батькові	Тип	Розмір	Джерело	З	До
Древоронов	Олег	Олегович	salary	9000.00	державне підприємство	2021-02-02	2022-02-09

Витрати членів родини					
Прізвище	Ім'я	По-батькові	Призначення	Вартість	Дата
Древоронов	Олег	Олегович	оплата навчання	3500.00	2022-01-06

Приміщення у власності членів родини					
Прізвище	Ім'я	По-батькові	Площа	Кількість проживаючих осіб	Адреса
Древоронов	Дмитрій	Генріхович	49 м. кв.	2 особи	Одеса, вул. Пастера, 12

Рисунок 5.20 – Перші розділи зареєстрованої декларації

Оформивши декларацію та необхідні довідки, відвідувач може перейти до процесу заповнення заяви на отримання допомоги. Форма заяви наведена на рис. 5.21.

Оформлення заяви на отримання соціальної допомоги

До органу праці та соціального захисту населення

від Древоронов Іван Генрихович,

зареєстрований за адресою: Одеса, вул. Буніна, 32

контактний телефон: 0972187716

паспорт: 093812485

Прошу призначити

Обрати необхідне ▾

Оберіть потрібну декларацію та документи:

Оберіть декларацію ▾

Оберіть довідку про стан здоров'я ▾

Андрій Іванович Древоронов 2020-06-03 ▲
Євген Генрихович Древоронов 2020-02-05

Генріх Вальтерович Древоронов 1952-01-02 II ▲
Олександр Ігорович Кондратюк 1992-02-05 III

Підтверджую відсутність змін у складі сім'ї

Про відмову в призначенні або припинення виплати призначеної соціальної допомоги у разі подання неповних чи недостовірних відомостей про доходи та майновий стан сім'ї мене попереджено

Подати заяву

Рисунок 5.21 – Форма заяви на отримання соціальної допомоги

Заповнивши заяву, відвідувач має зачекати допоки співробітник її перевірить та вирішить, чи може відвідувач претендувати на отримання соціальної допомоги.

Відвідувач може відслідковувати стан своєї заяви та перевірити додані до цієї заяви документи (рис. 5.22, 5.23).



Подача заяви

Перейдіть на цю форму, щоб оформити заяву на отримання соціальної допомоги

Подача заяви



Стан заяви

Перейдіть на цю форму, щоб перевірити стан поданої заяви

Перевірити стан заяви

Рисунок 5.22 – Вікно оформлення допомоги

Заява

Дата подання: 2022-06-05
Статус заяви: заява на розгляданні
Примітки: -

Додані до заяви документи

Декларація про доходи та витрати

Члени родини							
Прізвище	Ім'я	По-батькові	Ступінь зв'язку	ИДН	Примітки		
Древоронов	Олег	Олегович	II	093876122353	двоюрідний брат		
Древоронов	Дмитрій	Генріхович	I	873876422353	рідний брат		
Доходи членів родини							
Прізвище	Ім'я	По-батькові	Тип	Розмір	Джерело	З	До
Древоронов	Олег	Олегович	salary	9000.00	державне підприємство	2021-02-02	2022-02-09
Витрати членів родини							
Прізвище	Ім'я	По-батькові	Призначення		Вартість	Дата	
Древоронов	Олег	Олегович	оплата навчання		3500.00	2022-01-06	

Рисунок 5.23 – Вікно відслідковування стану заяви

ВИСНОВКИ

У ході дипломного проектування виконано аналіз предметної області «Соціальний захист громадян» та виділені основні проблеми існуючого аналогу застосунку, зокрема нестабільна робота сервісу внаслідок великої кількості сторонніх функцій, не пов'язаних з наданням соціальної допомоги, та обмежена кількість доступних у ньому типів допомоги.

Створена інформаційна система вирішує означені проблеми, оскільки є застосунком більш вузької спеціалізації, який дозволяє оформляти соціальну допомогу різних типів, зокрема державну допомогу малозабезпеченим сім'ям, інвалідам I, II та III групи, сім'ям з дітьми та особам, які доглядають осіб з I, II та III групою інвалідності. Спроектowana система розрахована на мінімальну очну взаємодію співробітників із відвідувачами, яка обмежується реєстрацією облікового запису відвідувача та принесенням необхідного пакету паперових документів до УПіСЗН.

Для зберігання даних спроектована та створена база даних під управлінням СУБД PostgreSQL. Для реалізації інформаційної системи використані сучасні засоби створення веб-застосунків, зокрема трирівнева архітектура клієнт-сервер, шаблон проектування MVVM, мова програмування C#, технологія програмування веб-додатків Razor Pages та фреймворк EntityFramework із бібліотекою Npgsql.

Безпека інформаційної системи забезпечується шляхом використання розмежування доступу до бази даних за допомогою механізму ролей і привілеїв, а також хешування паролів за допомогою модулю pgcrypto СУБД PostgreSQL.

Розроблена система підтримує три категорії користувачів: адміністраторів, співробітників та відвідувачів, при цьому:

- адміністратор керує обліковими записами співробітників (створює нові облікові записи, редагує та видаляє старі);

– співробітник керує обліковими записами відвідувачів, перевіряє їх заяви та призначає їм виплати на основі наданих до заяви документів;

– відвідувач може заповнювати декларації та довідки, необхідні для отримання соціальної допомоги, подавати заяви на отримання допомоги та переглядати призначені йому виплати.

Створений програмний продукт призначений, з одного боку, для зменшення кількості часу, який відвідувачі витрачають на оформлення соціальної допомоги, а з другої – на оптимізацію часу співробітників шляхом автоматизації процесу оформлення документів та призначення виплат. Застосунок має великий потенціал до розширення та додавання додаткових функцій (наприклад, підтримка інших випадків призначення соціальної допомоги) у контексті соціального захисту населення.

Ідея створення програмного продукту та його проект представлені на дев'ятнадцятій всеукраїнській конференції студентів і молодих науковців, тези доповіді опубліковані [5].



СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ

1. Онлайн-оформлення допомоги при народженні дитини. e-Dem [Електронний ресурс] – Режим доступу: <https://petition.e-dem.ua/n/Petition/View/516>
2. Допомога при народженні дитини [Електронний ресурс] – Режим доступу: <https://diia.gov.ua/services/dopomoga-pri-narodzhenni-ditini>
3. Що таке Дія? [Електронний ресурс] – Режим доступу <https://diia.gov.ua/faq/8>
4. Наскільки захищений сервіс Дія? [Електронний ресурс] – Режим доступу: <https://diia.gov.ua/faq/17>
5. Прохоров О.О., Розновець О.І. Інформаційна система управління соціального захисту населення / Інформатика, інформаційні системи та технології: тези доповідей дев'ятнадцятої всеукраїнської конференції студентів і молодих науковців. Одеса, 29 квітня 2022 р. – Одеса, 2022. – с. 95-97
6. 5 Benefits of a 3-Tier Architecture [Електронний ресурс] – Режим доступу: <https://insightsoftware.com/blog/5-benefits-of-a-3-tier-architecture/>
7. Learning JavaScript Design Patterns [Електронний ресурс] – Режим доступу: <https://www.oreilly.com/library/view/learning-javascript-design/9781449334840/ch10s07.html>
8. Малахов Є.В. Основи проектування БД: Конспект лекцій. [Електронний ресурс] – Режим доступу: [http://computer-science.onu.edu.ua/Малахов Организация баз данных.pdf](http://computer-science.onu.edu.ua/Малахов%20Организация%20баз%20данных.pdf).
9. PostgreSQL: The World's Most Advanced Open Source Relational Database [Електронний ресурс] – Режим доступу: <https://www.postgresql.org/>
10. What is Entity Framework? [Електронний ресурс] – Режим доступу: <https://www.entityframeworktutorial.net/what-is-entityframework.aspx>

ДОДАТОК А

Створення таблиць та типів даних

А.1 Створення власних типів

```
create type subsidy_type as enum
('allowance',
'compensation');
create type category as enum
('administrator',
'employee',
'visitor');
create type position_staff as enum
('specialist',
'senior specialist',
'director');
create type subsidy_status as enum (
'active', 'canceled', 'finished'
);
create type application_status as enum (
'filed', 'approved', 'rejected'
);
create type disability_group_type as enum (
'I', 'II', 'III'
);
create type subsidy_type as enum (
'child care', 'income level', 'disability grant', 'disability
care'
);
create type ownership_type as enum (
'private', 'communal', 'state'
);
create type income_type as enum (
'salary', 'pension', 'entrepreneurship', 'allowance'
);
```

А.2 Створення таблиць

```
create table Staff (
staff_no serial primary key,
f_name varchar(20) not null,
```

```

patronymic varchar(20) not null,
l_name varchar(20) not null,
"position" position_staff not null,
login varchar(30) not null
);

```

```

create table HealthCertificate (
certificate_no serial primary key,
owner_no integer not null,
disability_group disability_group_type not null,
last_inspection date not null check(last_inspection <
period_start),
period_start date not null check(period_start <= current_date),
period_end date not null check(period_end > period_start),
comment varchar(100),
foreign key (owner_no) references Visitor (certificate_no) on
delete cascade on update cascade
);

```

```

create table Users (
login varchar(30) not null primary key,
"password" varchar(30) not null,
category category not null
);

```

```

create table Visitor (
visitor_no serial primary key,
f_name varchar(20) not null,
patronymic varchar(20) not null,
l_name varchar(20) not null,
date_of_birth date not null check(date_of_birth < current_date),
assigned_to integer not null,
passport_no varchar(9) not null check(passport_no similar to
'[0-9]{9}'),
idn varchar(12) not null check(idn similar to '[0-9]{12}'),
address varchar(30) not null,
mobile varchar(10) unique check(mobile similar to '[0-9]{10}'),
login varchar(30) not null,
unique(passport_no),
unique(idn),
unique(login),
foreign key (assigned_to) references Staff(staff_no) on delete
restrict on update cascade,
foreign key (login) references Users(login) on delete restrict
on update cascade
);

```

```

create table ChildCertificate (
certificate_no serial primary key,
application_no integer,
guardian_no integer not null,
date_of_birth date not null check(date_of_birth < current_date),
f_name varchar(20) not null,
patronymic varchar(20) not null,
l_name varchar(20) not null,
foreign key (guardian_no) references Visitor(visitor_no) on
delete restrict on update cascade,
);

```

```

create table DisabledPersonCertificate (
certificate_no serial primary key,
guardian_no integer not null,
application_no integer,
disability_group disability_group_type not null,
date_of_birth date not null,
f_name varchar(20) not null,
patronymic varchar(20) not null,
l_name varchar(20) not null,
foreign key (guardian_no) references Visitor(visitor_no) on
delete restrict on update cascade,
);

```

```

create table Subsidy (
payment_no serial primary key,
visitor_no integer not null,
application_no integer not null,
appointed_by integer not null,
payment_amount money not null
check(payment_amount::decimal>0.00),
"type" subsidy_type not null,
period_start date not null check(period_start >= current_date),
period_end date not null check(period_end > period_start),
comment varchar(100),
status subsidy_status not null
foreign key (visitor_no) references Visitor(visitor_no) on
delete restrict on update cascade,
foreign key (appointed_by) references Staff(staff_no) on delete
restrict on update cascade
foreign key (application_no) references
Application(application_no) on delete restrict on update cascade
);

```

```

create table if not exists IncomePropertyDeclaration (
declaration_no serial primary key,
application_no integer not null,
owner_no integer not null,
document_no varchar(10) unique,
date_of_issue date check (date_of_issue < current_date)
);

```

```

create table if not exists IncomeFamily (
record_no serial primary key,
declaration_no integer not null,
member_no integer not null,
"type" income_type not null,
amount money not null,
source_of_income varchar(30) not null,
income_start date not null,
income_finish date not null check(income_finish>income_start and
income_finish<=current_date),
foreign key (member_no) references familymemberscertificate
(record_no) on delete restrict on update cascade
);

```

```

create table if not exists PropertyFamily (
record_no serial primary key,
declaration_no integer not null,
member_no integer not null,
area decimal not null check(area>0.0),
person_num integer not null,
adress varchar(30) not null,
foreign key (member_no) references familymemberscertificate
(record_no) on delete restrict on update cascade
);

```

```

create table if not exists LandFamily (
record_no serial primary key,
declaration_no integer not null,
member_no integer not null,
area decimal not null check(area>0.0),
"type" ownership_type not null,
purpose varchar(30) not null,
foreign key (member_no) references familymemberscertificate
(record_no) on delete restrict on update cascade
);

```

```

create table if not exists SpendingFamily (
record_no serial primary key,

```

```
declaration_no integer not null,  
member_no integer not null,  
"type" varchar (30) not null,  
price money not null,  
date_of_purchase date not null check(date_of_purchase >  
(current_date - 365)),  
foreign key (member_no) references familymemberscertificate  
(record_no) on delete restrict on update cascade  
);
```

```
create table if not exists Application (  
application_no serial primary key,  
owner_no integer not null,  
application_date date not null,  
application_subsidy_type subsidy_type not null,  
status application_status not null,  
comment  
);
```

ДОДАТОК Б

Створення тригерів

Тригерна функція `encrypt_password` та тригер для хешування паролю (лістинг Б.1):

```
create or replace function encrypt_password()
returns trigger as $$
begin
    if (tg_op = 'insert' or new.password not like old.password)
    then
        new.password = encode(digest(new.password::bytea, 'sha256'),
        'base64');
    end if;
    RETURN NEW;
end;
$$ language plpgsql;

create trigger make_encrypted_password
before insert or update on users
for each row execute procedure encrypt_password();
```

Лістинг Б.1 – Тригерна функція та тригер для хешування паролю

Тригерна функція `subsidy_canceling` та тригер для оновлення дати закінчення виплати (лістинг Б.2):

```
create or replace function subsidy_canceling()
returns trigger as $$
begin
    if new.status = 'canceled' then new.period_end = current_date;
    end if;
    return new;
end;
$$ language plpgsql;
```

Лістинг Б.2 – Тригерна функція та тригер для оновлення дати
закінчення виплати

```
create trigger status_canceled
before insert or update on subsidy
for each row execute procedure subsidy_canceling();
```

Лістинг Б.2, лист 2

Тригерна функція `subsidy_added` та тригер для перевірки наявності активної виплати (лістинг Б.3):

```
create or replace function subsidy_added()
returns trigger as $$
declare active_subsidy int;
begin
if new.status = 'active' then
select payment_no
into active_subsidy
from subsidy
where visitor_no = new.visitor_no and status = 'active';
if active_subsidy is not null then
raise exception 'Посетитель % имеет активную выплату %',
    new.visitor_no, active_subsidy;
end if;
end if;
return new;
end;
$$ language plpgsql;

create trigger active_subsidy_trigger
before insert or update on subsidy
for each row
execute procedure subsidy_added();
```

Лістинг Б.3 – Тригерна функція та тригер для перевірки наявності активної виплати