

МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ
Національний університет кораблебудування імені адмірала Макарова

Навчально-науковий інститут комп'ютерних наук та управління проєктами
Кафедра програмного забезпечення автоматизованих систем
Спеціальність 121 «Інженерія програмного забезпечення»
Освітня програма «Інженерія програмного забезпечення»

«Допущений до захисту»
Завідувач кафедри

« ____ » _____ 2022 р.

КВАЛІФІКАЦІЙНА РОБОТА
на здобуття ступеня вищої освіти «бакалавр»

на тему: Розробка програмного забезпечення вебсайту для банківської системи "Legalist"

Виконав: студент 4151 групи



(підпис, ПІБ)

Качанов Є.Г.

Керівник роботи:

Викладач, к.т.н

(посада, науковий ступень вчене звання)

Пухалевич А.В.

(підпис, ПІБ)

МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ
Національний університет кораблебудування імені адмірала Макарова

Навчально-науковий інститут комп'ютерних наук та управління проєктами
Кафедра програмного забезпечення автоматизованих систем
Спеціальність 121 «Інженерія програмного забезпечення»
Освітня програма «Інженерія програмного забезпечення»

«ЗАТВЕРДЖУЮ»
Гарант освітньої програми
_____ доц. Макарова Л.М.
(підпис)
«__» _____ 2022 р.

ЗАВДАННЯ
НА КВАЛІФІКАЦІЙНУ РОБОТУ
на здобуття ступеня вищої освіти «бакалавр»

Студенту Качанов Євген Геннадійович _____
(Прізвище, ім'я, по батькові)

1. Тема роботи: Розробка програмного забезпечення вебсайту для банківської системи "Legalist" _____

Керівник роботи Пухалевич Андрій Володимирович _____
Затверджені наказом ректора №256-уч від «11» травня 2022 р.

2. Термін подання роботи: 10.06.2022 р. _____

3. Вихідні дані по роботі: _____

4. Перелік питань, що належать до розробки (найменування розділів) _____
Титульний аркуш; Завдання на кваліфікаційну роботу; Анотація (українською та англійською мовами); Зміст; Перелік умовних позначень, символів, одиниць та термінів (при необхідності); Вступ; Опис предметної галузі та постановка задачі; Проєкт програмного забезпечення (ескізний, технічний та робочий); Результати розробки програмного забезпечення; Розділ з охорони праці; Висновки; Список використаних джерел; Додатки (технічне завдання, опис програми, інструкція користувача, програма та методика випробувань програмного забезпечення).

5. Перелік презентаційних матеріалів 1. Титульний слайд, 2. Мета, актуальність та основні завдання роботи, 3. Алгоритм створення вебсайту, 4. Технології під час розробки, 5. Основні актори у системі та діаграма використання, 6. Діаграма сутність-зв'язок, 7. Логічна та фізична модель бази даних, 8. Прототипи ескізів інтерфейсу, 9. Головна сторінка вебсайту 10. Сторінка авторизації та форми заповнення заявки. 11. Висновки. _____

6. Консультанти розділів роботи

Розділ	Прізвище, ініціали та посада консультанта	Підпис, дата	
		завдання видав	завдання прийняв
4	Гурець Н.В., ст. викладач		

7. Дата видачі завдання 11.05.2022 р.

КАЛЕНДАРНИЙ ПЛАН

Номер	Назва етапів роботи	Терміни виконання	Примітка
1.	Підготовка розділу Вступ	24.03.2022	ПП*
2.	Опис та аналіз предметної галузі	28.03.2022	ПП*
3.	Постановка задачі	31.03.2022	ПП*
4.	Ескізний проєкт програмного забезпечення	27.04.2022	
5.	Технічний проєкт програмного забезпечення	06.05.2022	
6.	Робочий проєкт програмного забезпечення	13.05.2022	
7.	Підготовка розділу Результати розробки програмного забезпечення	17.05.2022	
8.	Підготовка розділу з охорони праці	18.05.2022	ПП*
9.	Підготовка розділу Висновки	19.05.2022	
10.	Оформлення списку використаних джерел та додатків	20.05.2022	
11.	Подання на кафедру ПЗАС тексту остаточного варіанту роботи, підписаного її керівником, у роздрукованому та електронному форматі разом із заявами щодо самостійності виконання роботи та ідентичності друкованої та електронної версій роботи (Додатки 1 і 2 «Порядку здійснення заходів з перевірки робіт на наявність текстових збігів/ідентичності/схожості із використанням програмно-технічних засобів», який введений в дію наказом ректора НУК за №20 від 20.01.2020 р.)	10.06.2022	не пізніше, ніж за 14 діб до захисту (згідно п.4.1 зазначеного Порядку)
12.	Підготовка презентації та доповіді	12.06.2022	
13.	Попередній захист роботи на засіданні кафедри ПЗАС	14.06.2022	
14.	Подання на кафедру ПЗАС електронних версій наступних документів у форматі pdf: кваліфікаційної роботи; файлу-опису кваліфікаційної роботи (згідно Додатку до наказу ректора НУК за №287-уч від 19.05.2020 р.); презентації доповіді	24.06.2022	

* - за результатами переддипломної практики (ПП), яка була з 21.03.2022 до 08.05.2022 р.

Студент



(підпис)

Качанов Є.Г.

(ПБ)

Керівник роботи

(підпис)

Пухалевич А.В.

(ПБ)

АНОТАЦІЯ

Дана кваліфікаційна робота присвячена розв'язанню практичної задачі інженерії програмного забезпечення, що характеризується комплексністю та невизначеністю умов: розробці проєкту програмного забезпечення вебсайту системи для консультацій у банківській сфері «Legalist» та його реалізації, що дозволить працівникам банківської системи "Legalists" спростити та пришвидшити свою роботу.

У роботі наведено аналіз вимог до предметної області за темою кваліфікаційної роботи, постановка задачі, проєкт програмного забезпечення, а також результати розробки та розділ з охорони праці.

Дана робота виконана на 104 сторінках, містить 50 рисунків, 29 таблиць, 5 додатки та список використаних джерел з 12 найменувань.

ABSTRACT

This qualification work is devoted to solving the practical problem of software engineering, characterized by complex and uncertain conditions: the development of software project website for banking consulting "Legalist" and its implementation, which will allow employees of the banking system "Legalists" to simplify and speed up their work.

The paper presents an analysis of the requirements for the subject area on the topic of qualification work, problem statement, software project, as well as the results of development and the section on labor protection.

This work is performed on 104 pages, contains 50 figures, 29 tables, 5 appendices and a list of used sources of 12 items.

ЗМІСТ

ВСТУП	7
1 АНАЛІЗ ПРЕДМЕТНОЇ ОБЛАСТІ ТА МЕТОДІВ СТВОРЕННЯ ПРОГРАМНОГО ЗАБЕЗПЕЧЕННЯ ДИНАМІЧНИХ ВЕБСАЙТІВ ДЛЯ КОНСУЛЬТАЦІЙ У БАНКІВСЬКІЙ СФЕРІ. ПОСТАНОВКА ЗАДАЧІ НА РОЗРОБКУ ПРОГРАМНОГО ЗАБЕЗПЕЧЕННЯ ВЕБСАЙТУ СИСТЕМИ ДЛЯ КОНСУЛЬТАЦІЙ У БАНКІВСЬКІЙ СФЕРІ «LEGALISTS»	9
1.1 Аналіз предметної області та методів створення програмного забезпечення динамічних вебсайтів для консультацій у банківській сфері.....	9
1.2 Аналіз існуючого аналогічного програмного забезпечення	13
1.3 Постановка задачі для розробки програмного забезпечення вебсайту системи для консультацій у банківській сфері «Legalists»	15
2 ПРОЄКТ ПРОГРАМНОГО ЗАБЕЗПЕЧЕННЯ ВЕБСАЙТУ СИСТЕМИ ДЛЯ КОНСУЛЬТАЦІЙ У БАНКІВСЬКІЙ СФЕРІ «LEGALIST»» ТА ЇЇ РЕАЛІЗАЦІЯ	17
2.1 Ескізний проєкт програмного забезпечення вебсайту системи для консультацій у банківській сфері «Legalist».....	17
2.1.1 Вибір мови моделювання.....	17
2.1.2 Побудова моделі варіантів використання	17
2.1.3 Специфікації варіантів використання.....	20
2.1.4 Розробка концептуальної моделі.....	28
2.1.5 Розробка ескізу інтерфейсу.....	29
2.2 Технічний проєкт програмного забезпечення вебсайту системи для консультацій у банківській сфері «Legalist».....	34
2.2.1 Логічна модель бази даних	34
2.2.2 Діаграма класів програмного забезпечення	36
2.2.3 Динамічна модель програмного забезпечення	37
2.3 Робочий проєкт програмного забезпечення вебсайту системи для консультацій у банківській сфері «Legalist».....	52

	6
2.3.1 Обґрунтування вибору мови програмування та СКБД.....	52
2.3.2 Фізична модель бази даних.....	52
2.3.3 Реалізація та тестування основних класів еквівалентності програмного забезпечення.....	55
2.3.4 Випробування програмного забезпечення	58
3 РЕЗУЛЬТАТИ РОЗРОБКИ	60
4 ОХОРОНА ПРАЦІ.....	62
4.1 Аналіз шкідливих та небезпечних факторів на робочому місці	62
4.2 Заходи щодо зменшення впливу шкідливих факторів.....	64
ВИСНОВКИ.....	66
СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ.....	67
ДОДАТОК А ТЕХНІЧНЕ ЗАВДАННЯ.....	69
ДОДАТОК Б ТЕКСТ ПРОГРАМИ	77
ДОДАТОК В ОПИС ПРОГРАМИ.....	90
ДОДАТОК Г ІНСТРУКЦІЯ КОРИСТУВАЧА	93
ДОДАТОК Д ПРОГРАМА ТА МЕТОДИКА ВИПРОБУВАНЬ ПРОГРАМНОГО ЗАБЕЗПЕЧЕННЯ	102

ВСТУП

Дана кваліфікаційна робота передбачає розв'язання практичної задачі інженерії програмного забезпечення, що характеризується комплексністю та невизначеністю умов: розробку програмного забезпечення вебсайту системи для консультацій у банківській сфері «Legalist».

У сучасному світі існує велика кількість різноманітних послуг у банківській сфері, однією з яких є надання консультацій у цій сфері. Саме дану послугу буде надавати банківська система «Legalist». Задля полегшення процесу надання консультацій клієнтам розробимо програмне забезпечення вебсайту системи для консультацій у банківській сфері «Legalist». Це дозволить автоматизувати надання необхідної для клієнта інформації та консультацій, автоматизувати облік клієнтів, котрі надіслали заявку на консультацію. ПЗ вебсайту буде отримувати надіслану клієнтом заявку на надання консультації та передавати її співробітнику.

Метою кваліфікаційної роботи є розробка програмного забезпечення вебсайту системи для консультацій у банківській сфері «Legalist».

Виділимо наступні завдання, які необхідні для успішної розробки:

- Проаналізувати предметну область консультацій у банківській сфері, з метою виявлення вимог для розроблюваного ПЗ.
- Розробити специфікацію модулів програмного забезпечення вебсайту системи для консультацій у банківській сфері «Legalist».
- Розробити технічне завдання на розробку програмного забезпечення вебсайту системи для консультацій у банківській сфері «Legalist».
- Розробити концептуальну модель програмного забезпечення вебсайту системи для консультацій у банківській сфері «Legalist» з метою виявлення відношень між сутностями бази даних.
- Виходячи з вимог до ПЗ, обрати найбільш вдалу комбінацію системи керування базами даних (СКБД) та програмних засобів.
- Розробити логічну та фізичну моделі програмного забезпечення вебсайту системи для консультацій у банківській сфері «Legalist».

- Виконати кодування програмного забезпечення вебсайту системи для консультацій у банківській сфері «Legalist» відповідно до розроблених моделей та з використанням обраних програмних засобів.
- Виконати тестування програмного забезпечення вебсайту системи для консультацій у банківській сфері «Legalist».
- Розглянути питання з охорони праці.

1 АНАЛІЗ ПРЕДМЕТНОЇ ОБЛАСТІ ТА МЕТОДІВ СТВОРЕННЯ ПРОГРАМНОГО ЗАБЕЗПЕЧЕННЯ ДИНАМІЧНИХ ВЕБСАЙТІВ ДЛЯ КОНСУЛЬТАЦІЙ У БАНКІВСЬКІЙ СФЕРІ. ПОСТАНОВКА ЗАДАЧІ НА РОЗРОБКУ ПРОГРАМНОГО ЗАБЕЗПЕЧЕННЯ ВЕБСАЙТУ СИСТЕМИ ДЛЯ КОНСУЛЬТАЦІЙ У БАНКІВСЬКІЙ СФЕРІ «LEGALIST»

1.1 Аналіз предметної області та методів створення програмного забезпечення динамічних вебсайтів для консультацій у банківській сфері

З розвитком технологій людство почало швидкий рух до автоматизації різних процесів, у тому числі створюючи банківські системи. Послугою, яку надає банківська система «Legalist» є консультації у банківській сфері.

Завдяки програмному забезпеченню вебсайту системи для консультацій у банківській сфері «Legalist» клієнт зможе отримати консультацію щодо послуг із банківської сфери, таких як інвестиції, умови отримання кредиту, фінансове регулювання, фінансування бізнесу, розміщення депозиту та інші. Для того, щоб отримати консультацію про потрібну йому послугу клієнт на вебсайті заповнює заявку та очікує дзвінка від оператора. Оператор отримує заявку від клієнта та обробляє її після чого з клієнтом зв'язуються для надання консультації щодо вказаної послуги у банківській сфері.

Заявка містить в собі наступну інформацію: EMAIL, контактний номер телефону для зв'язку з оператором, ПІБ клієнта, оператор який буде надавати послугу, питання за яким він звертається, коментар який він залишив, та дату надходження заявки.

Інформація про клієнтів містить в собі наступну інформацію: ПІБ клієнта, EMAIL, контактний номер телефону, паспортні дані.

Інформація про працівників містить в собі наступну інформацію: ПІБ працівника, дата народження, дата приймання на роботу, посада, номер телефону та номер паспорта.

Інформація про послуги містить в собі наступну інформацію: ID працівника,

вид послуги, дата початку роботи, деталі щодо замовлення, та ID клієнта. База даних призначена для використання лише працівниками та не доступна для перегляду клієнтам.

Розглянемо більш детально функції оператора. Оператор відповідальний за обробку та видалення заявок, створення послуги додавання даних про клієнтів.

Розглянемо більш детально функції спеціаліста із банківської сфери. Спеціаліст відповідальний за роботу з клієнтами надання їм кваліфікованої допомоги, консультацій та редагування даних про клієнтів.

Розглянемо більш детально функції адміністратора. Адміністратор відповідальний за внесення, коректування і видалення даних про працівників.

Конкретизуємо інформацію, яка необхідна для роботи ПЗ. Для роботи ПЗ необхідною є інформація про заявки, клієнтів, послуги та працівників.

1) Заявки можна описати наступною інформацією:

- ПІБ
- EMAIL;
- Номер телефону;
- Необхідна послуга;
- Коментар клієнта;
- Оператор;
- Дата надходження заявки.

2) Клієнта можна описати наступною інформацією:

- ПІБ;
- EMAIL;
- Номер телефону;
- Номер паспорта.

3) Працівника можна описати наступною інформацією:

- ПІБ;
- Номер телефону;
- Дата народження;
- Паспортні дані
- Посада;

- Дата приймання на роботу.
- 4) Послуги можна описати наступною інформацією:
 - Послуга за якою звертається клієнт;
 - Клієнт якому надають послугу
 - Деталі щодо замовлення;
 - Дата початку роботи;
 - Працівник який надає послугу.

Після отримання даних працівником він вносить їх до БД де вони зберігаються та оброблюються після чого клієнту надаються усі необхідні для нього послуги.

Існує декілька методів розробки динамічних вебсайтів, які підходять для області консультацій у банківській сфері. Виділимо три основні методи:

- 1) Використання системи управління вмістом (CMS);
 - 2) Створення вебсайтів із використанням конструкторів;
 - 3) Створення динамічних вебсайтів із використанням мови програмування.
- 1) Використання системи управління вмістом (CMS).

CMS - це система яка використовується для управління вмістом, вона також застосовується для створення, редагування та управління документами. Для встановлення CMS системи використовують уже обраний хостинг.

Головними функціями сучасних CMS є:

- Створення вебсторінок за запитом користувача;
- Управління графічними та текстовими блоками які містяться на сайті, та надають користувачу комфортний інтерфейс для взаємодії з контентом;
- Забезпечення зручними інструментами для зберігання та публікування інформації яка міститься на сайті;
- Автоматизація процесів розміщення інформації яка знаходиться в базі даних та публікування її у виді коду HTML;
- Демонстрація вебсторінок які використовуються для редагування розробником;
- Забезпечення необхідних інструментів для видалення вмісту з сайту.

CMS системи зазвичай використовується для створення таких вебсайтів як візитки, прості сторінки, так звані сайти односторінки дизайн яких орієнтований

на користувачів які майже не мають досвіду у програмуванні.

2) Створення вебсайтів із використанням конструкторів

Під час сучасної розробки вебсайтів є велика кількість рішень для швидкої та комфортної розробки вебсайтів. Вони надають можливість розроблювати HTML код, створити вебсайт у візуальному режимі, а також надають розробникам додаткові можливості для створення вебсайтів.

Виділимо основні інструментальні системи під час розробки на HTML:

- Програми які містять візуальні редактори (design-based editor) – завдяки цим програмам необхідний код HTML створюється автоматично.

- Програми редактори (code-based editors) – це програми як надають редактор та додаткові засоби для створення та редагування коду.

Розберемо популярні design-based редактори:

Adobe DreamWeaver - один із найзручніших візуальних редакторів які створюють HTML код. Цей редактор дозволяє працювати одночасно, напряду з HTML кодом або редагуючи сайт у візуальному режимі. Основним недоліком цього редактору є те, що він додає у код велику кількість непотрібних елементів. Це ускладнює редагування HTML коду [1].

Microsoft FrontPage - це простий в засвоєнні та комфортний вебредактор для створення проєкту, підготовки та публікації вебсайтів. Через інтеграцію із сімейством продуктів MS Office, програма має звичний для користувача інтерфейс, велику кількість шаблонів та легкість в освоєнні для користувачів які знайомі з базовими поняттями роботи в MS Word [1.]

Розберемо популярні code-based редактори:

- Adobe HomeSites - це редактор HTML який у своєму складі має велику кількість необхідних функцій та підпрограм. Також в редакторі присутній дистрибутив TopStyle для редагування таблиць CSS, орфографічну перевірку та багато інших корисних функцій [1].

- HotDog – це професійний редактор в який вбудована підтримка великого набору інструментів які використовуються під час створення вебдизайну. Перевірка синтаксису у цьому редакторі налаштовується в досить широких межах.

3) Створення динамічних вебсайтів із використанням мови програмування

Створення динамічного вебсайту із нуля – це досить трудомісткий та складний процес, але він дає можливість завдяки додатковим мовам програмування, таким як PHP, створити сайт який задовольняє усі потреби як користувача, так і розробника. Під час розробки сайту із нуля ми маємо можливість краще зрозуміти процес створення веб додатка та покращити результат розробки. Для створення сучасних вебсайтів використовується:

- HTML для створення скелета сайту;
- CSS для додавання стилів;
- JavaScript для додавання функціонала на стороні клієнта;
- PHP або інші серверні мови програмування для додавання функціонала на стороні вебсервера, включаючи доступ до бази даних;

Для розробки програмного забезпечення вебсайту системи для консультацій у банківській сфері «Legalist» обрано метод, створення динамічних вебсайтів із використанням мови програмування оскільки він дає більше можливостей для створення необхідного функціоналу у порівнянні з іншими способами.

1.2 Аналіз існуючого аналогічного програмного забезпечення

На сьогоднішній день існує багато вебсайтів для надання консультацій у банківській сфері. Ми розглянемо та проаналізуємо декілька з них та визначимо їхні плюси та мінуси, щоб прийти до висновку щодо доцільності розробки програмного забезпечення вебсайту системи для консультацій у банківській сфері «Legalist».

«u-grand.com.ua» – сайт який надає можливість отримати консультацію з питань банківської сфери. Основними питаннями для консультації є отримання кредитів, передача майна, погашення кредитних зобов'язань. Головними недоліками є некоректна робота сайту та незручний дизайн. Титульна сторінка сайту представлена на рисунку 1.1.

Консультації з питань банківської сфери

Надання консультацій з питань, що стосуються банківської сфери:

- отримання кредитів, розрахунки за кредитними зобов'язаннями;
- передача майна в заставу (іпотеку);
- реалізація (протидія реалізації) заставленого майна;
- погашення кредитних зобов'язань;
- аспекти процедури складного банкрутства

Наші послуги

Судово-правовий напрямок

- Фізичним особам
- Юридичним особам

Позасудово-консультаційний напрямок

- Представництво в державних та недержавних установах
- Консультації з питань банківської сфери
- Корпоративні питання
- Правове супроводження операцій з нерухомістю

Правовий супровід бізнес-проектів

Супроводження інвестиційних проектів

Правовий супровід громадян України за кордоном

Юридичний супровід іноземців в

Рисунок 1.1 - Титульна сторінка сайту “u-grand.com.ua” [2]

enomi.pl – сайт який надає можливість отримати консультацію з питань банківської сфери. Сайт надає можливість отримати консультацію для розміщення депозиту, отримання кредиту та виконання валютних операцій. Серед переваг сайту можна назвати невелику вартість послуг. Головними недоліками є відсутність української мови та повільна швидкість роботи. Титульна сторінка сайту представлена на рисунку 1.2.



БАНКОВСКИЙ КОНСАЛТИНГ В УКРАИНЕ

Направленный на предоставление исчерпывающих консультаций и помощи юридическим и физическим лицам, физическим лицам-предпринимателям в проведении каких либо банковских операций на территории Украины. Мы предлагаем Вам новый профессиональный уровень в сотрудничестве с банками. Основные наши возможности:

1. Консультирование и сопровождение при кредитных и гарантийных операциях:

- определение необходимого пакета документов;
- калькуляция всех возможных затрат;
- определение возможности получения кредита, овердрафта, гарантии;
- для физических лиц ознакомления с действующими кредитными программами (кредит на приобретение автомобиля, недвижимости, потребительские кредиты под залог).

2. Консультирование и сопровождение вкладных операций, размещения средств в банках на депозитные счета (в т.ч. инвестиционные депозиты):

- анализ надёжности банков;
- предоставление полной информации о видах депозитов и действующих депозитных ставках;
- управление депозитными счетами;
- организация системы получения активного и пассивного дохода от свободных средств, от средств на текущих счетах (безналичных и наличных);
- анализ депозитных договоров.

3. Консультирование и сопровождение сделок связанных с кредитами от нерезидентов:

- формирование кредитных договоров в соответствии с действующим законодательством;

Рисунок 1.2 – Титульна сторінка сайту “enomi.pl” [3]

Зробивши аналіз аналогічних існуючих вебсайтів для автоматизації надання консультацій для клієнтів, можна виділити наступні недоліки вже існуючих рішень:

- існуючі вебсайти не надають можливості отримати консультацію безкоштовно;
- мала кількість послуг, щодо яких можливо отримати консультацію;
- висока ціна у порівнянні з розробкою ПЗ в межах цієї роботи;
- в існуючих сервісах не має деяких необхідних можливостей;
- більш об’ємне ПЗ з точки зору функцій, що в свою чергу збільшує час на освоєння такого ПЗ людьми, що будуть з ним працювати.

Виходячи із цих недоліків можна зробити висновок, що розробка проєкту програмного забезпечення вебсайту системи для консультацій у банківській сфері «Legalist» та його реалізація мають практичну цінність.

1.3 Постановка задачі на розробку програмного забезпечення вебсайту системи для консультацій у банківській сфері “Legalists”

Провівши аналіз предметної області та методів створення програмного забезпечення вебсайту системи для консультацій у банківській сфері “Legalists”, було показано доцільність розробки відповідного вебсайту.

Функціональним призначенням програмного забезпечення вебсайту для консультацій банківської системи "Legalists" є автоматизація процесів:

- отримання та обробка заявок від клієнтів;
- облік надання консультацій клієнту;
- зберігання та використання інформації про клієнтів;
- зберігання та використання інформації про працівників;

Під час роботи оператор повинен мати можливість виконувати наступні задачі:

- авторизуватись;
- переглядати список заявок;
- призначати оператора заявці;

- видаляти заявки;
- створювати послуги;
- додавати клієнтів.

При роботі спеціаліст повинен мати можливість виконувати наступні задачі:

- авторизуватись;
- перегляд списку клієнтів;
- корегувати дані про клієнтів;
- перегляд списку послуг;
- призначення спеціаліста послугі.

При роботі адміністратор повинен мати можливість виконувати наступні задачі:

- авторизуватись;
- вносити дані про працівників;
- корегувати дані про працівників;
- видаляти дані про працівників.

Авторизація користувачів в програмному забезпеченні необхідна для надання їм необхідних функціональних можливостей. Обов'язкові поля для авторизації є логін та пароль.

2 ПРОЄКТ ПРОГРАМНОГО ЗАБЕЗПЕЧЕННЯ ВЕБСАЙТУ СИСТЕМИ ДЛЯ КОНСУЛЬТАЦІЙ У БАНКІВСЬКІЙ СФЕРІ «LEGALIST»

2.1 Ескізний проєкт програмного забезпечення вебсайту системи для консультацій у банківській сфері «Legalist»

2.1.1 Вибір мови моделювання

Для розробки програмного забезпечення вебсайту системи для консультацій у банківській сфері «Legalist» було обрано мову моделювання UML. Уніфікована мова моделювання (UML) – це графічна мова яка використовується для візуалізації, визначення, проєктування та документування систем, у яких програмне забезпечення відіграє важливу роль. Використовуючи UML, можливо розробити детальний план створеної системи, включаючи не тільки її концептуальні елементи, такі як системні функції та бізнес-процеси, але й конкретні функції, такі як класи, написані спеціальними мовами програмування, схеми баз даних і програмне забезпечення для багаторазового використання. Мова моделювання UML проста та зрозуміла вона підходить для створення моделювання будь-якого програмного забезпечення [4].

2.1.2 Побудова моделі варіантів використання

Використання діаграми варіантів використання є першим кроком від розробки загальної абстрактної концептуальної моделі програмного забезпечення до логічної та фізичної моделі, яка містить важливі деталі реалізації. Використання діаграми варіантів використання потрібне для демонстрації функціональних вимог програмного забезпечення зрозумілим не лише для програмних інженерів.

Після проведеного аналізу та постановки задачі до розробки програмного забезпечення вебсайту системи для консультацій у банківській сфері «Legalist», можемо виявити акторів та варіанти використання ПЗ, що розроблюється.

На рисунку 2.1 представлені основні кандидати в актори системи.

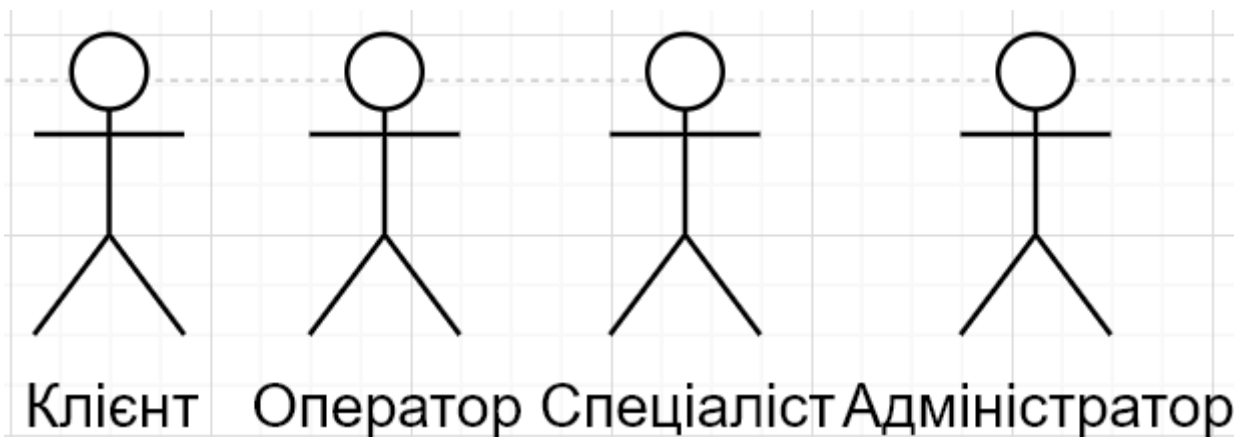


Рисунок 2.1 – Аналіз акторів системи

Короткий опис виявлених акторів представлено в таблиці 2.1.

Таблиця 2.1 – Виявлення акторів

Актор	Короткий опис
Оператор	Особа яка відповідає за обробку заявок, внесення даних про клієнтів та створення послуги
Спеціаліст	Особа яка відповідає за надання послуги та редагування даних про клієнтів
Клієнт	Особа яка відповідає за подання заявки
Адміністратор	Особа яка відповідає за внесення редагування та видалення даних про працівників

Виявлені варіанти використання ПЗ представлені у таблиці 2.2

Таблиця 2.2 – Виявлення варіантів використання

Код	Основний актор	Назва	Формулювання
1	2	3	4
K1	Клієнт	Надсилати заявку	Цей варіант використання дозволяє клієнту надсилати заявку
O1	Оператор	Перегляд списку заявок	Цей варіант використання дозволяє оператору переглядати список усіх доступних заявок
O2	Оператор	Призначення оператора заявці	Цей варіант використання дозволяє оператору розпочати роботу із заявками

Продовження таблиці 2.2

O3	Оператор	Додавання клієнта	Цей варіант використання дозволяє оператору додавати дані про клієнтів у базу даних
O4	Оператор	Видалення заявки	Цей варіант використання дозволяє оператору видалити заявку із БД
O5	Оператор	Створення послуги	Цей варіант використання дозволяє оператору створити послугу
O6	Оператор	Перегляд списку клієнтів	Цей варіант використання дозволяє спеціалісту переглядати список клієнтів
C1	Спеціаліст	Перегляд списку послуг	Цей варіант використання дозволяє спеціалісту переглядати список послуг
C2	Спеціаліст	Перегляд списку клієнтів	Цей варіант використання дозволяє спеціалісту переглядати список клієнтів
C3	Спеціаліст	Призначення спеціаліста послугі	Цей варіант використання надає можливість спеціалісту обрати послугу.
C4	Спеціаліст	Редагування даних клієнта	Цей варіант використання дозволяє спеціалісту редагувати дані про клієнтів у БД
A1	Адміністратор	Додавання працівника	Цей варіант використання дозволяє адміністратору додавати дані про працівника у БД
A2	Адміністратор	Редагування даних працівника	Цей варіант використання дозволяє адміністратору редагувати дані про працівника у БД
A3	Адміністратор	Видалення працівника	Цей варіант використання дозволяє адміністратору видаляти дані про працівника у БД
A4	Адміністратор	Авторизуватися	Цей варіант використання дозволяє Адміністратору авторизуватися
O6	Оператор	Авторизуватися	Цей варіант використання дозволяє Оператору авторизуватися
C5	Спеціаліст	Авторизуватися	Цей варіант використання дозволяє Спеціалісту авторизуватися

На основі виявлених акторів та варіантів використання системи побудуємо модель варіантів використання. Модель варіантів використання представлена на рисунку 2.2.

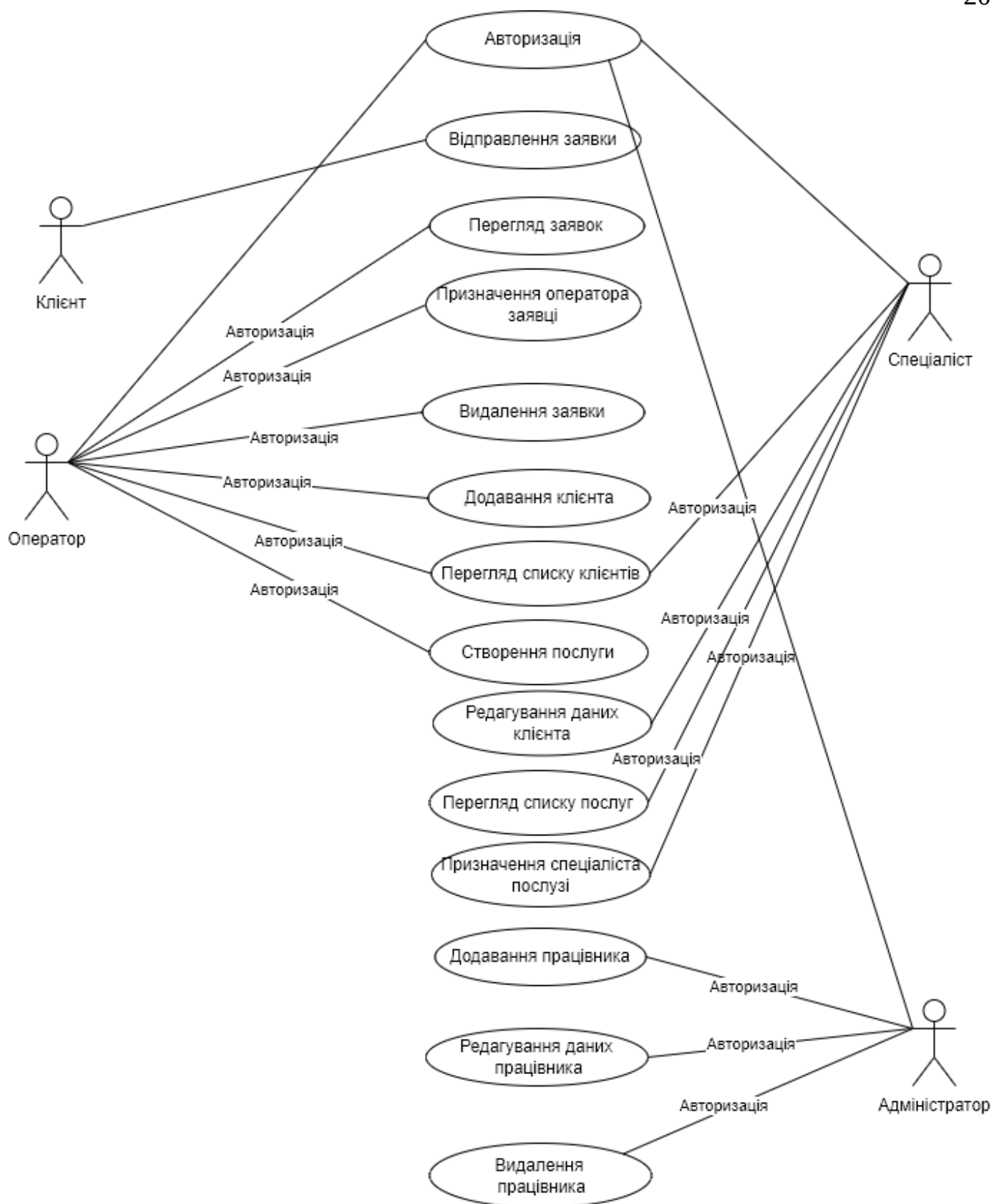


Рисунок 2.2 – Модель варіантів використання

2.1.3 Специфікації варіантів використання

1) Авторизація

Назва варіанту використання: «Авторизація»

Короткий опис: варіант використання ініціюється активним суб'єктом «Оператор», «Спеціаліст» або «Адміністратор» і дозволяє працівникам авторизуватися.

Потоки подій

Основний потік

Працівник знаходячись на головній сторінці вебсайту натискає «Вхід». Вебсайт відображає вікно з полями вводу. Працівник заповнює всі поля даними (у випадку помилки введення даних – виконується альтернативний потік) і підтверджує свій вибір.

Альтернативні потоки

Помилка у форматі даних - буде виведено повідомлення про помилку з поясненням помилки та запит на повторне введення даних користувачем.

Спеціальні умови: спеціальні умови невизначені.

Передумови: передумови не визначені.

Постумови: постумови не визначені.

Додаткові зауваження: додаткових зауважень немає.

2)Відправлення заявки

Назва варіанту використання: «Відправлення заявки»

Короткий опис: варіант використання ініціюється активним суб'єктом «Клієнт» і дозволяє клієнту відправити заявку.

Потоки подій

Основний потік

Клієнт знаходячись на головній сторінці вебсайту вводить усі необхідні дані для заявки та натискає відправити (у випадку помилки введення даних – виконується альтернативний потік).

Альтернативні потоки

Помилка у форматі даних - буде виведено повідомлення про помилку з поясненням помилки та запит на повторне введення даних користувачем.

Спеціальні умови: спеціальні умови невизначені.

Передумови: передумови не визначені.

Постумови: постумови не визначені.

Додаткові зауваження: додаткових зауважень немає.

3)Перегляд списку заявок

Назва варіанту використання: «Перегляд списку заявок»

Короткий опис: варіант використання ініціюється активним суб'єктом «Оператор» і дозволяє оператору переглядати список заявок.

Потоки подій

Основний потік

Оператор обирає в головному меню пункт «Список заявок» після чого вебсайт відображає на екран список з усіма заявками.

Альтернативні потоки

Не знайдено жодних записів у даній вкладці - буде виведено повідомлення про те, що жодних записів в даній вкладці знайдено не було.

Спеціальні умови: спеціальні умови невизначені.

Передумови: передумови не визначені.

Постумови: постумови не визначені.

Додаткові зауваження: додаткових зауважень немає.

4)Призначення оператора заявці

Назва варіанту використання: «Призначення оператора заявці»

Короткий опис: варіант використання ініціюється активним суб'єктом «Оператор» і дозволяє оператору призначити себе до вільної заявки.

Потоки подій

Основний потік

Оператор обирає в головному меню пункт “Список заявок” після чого обирає зі списку вільну заявку і натискає “Прийняти заявку”.

Альтернативні потоки

Помилка у форматі даних - буде виведено повідомлення що заявка уже зайнята іншим оператором.

Спеціальні умови: спеціальні умови невизначені.

Передумови: перед обиранням заявки потрібно обрати в меню пункт “Список заявок”.

Постумови: постумови не визначені.

Додаткові зауваження: додаткових зауважень немає.

5) Додавання клієнта

Назва варіанту використання: «Додавання клієнта»

Короткий опис: варіант використання ініціюється активним суб'єктом «Оператор» і дозволяє оператору додавати клієнта у БД.

Потоки подій

Основний потік

Оператор обирає в головному меню пункт «Додати клієнта». Вебсайт відображає вікно з полями вводу. Оператор заповнює всі поля даними (у випадку помилки введення даних – виконується альтернативний потік) і підтверджує свій вибір. Вебсайт додає клієнта у БД.

Альтернативні потоки

Помилка у форматі даних - буде виведено повідомлення про помилку з поясненням помилки та запит на повторне введення даних користувачем.

Спеціальні умови: спеціальні умови невизначені.

Передумови: передумови не визначені.

Постумови: постумови не визначені.

Додаткові зауваження: додаткових зауважень немає.

6) Перегляд списку клієнтів

Назва варіанту використання: «Перегляд списку клієнтів»

Короткий опис: варіант використання ініціюється активним суб'єктом "Оператор" та "Спеціаліст" і дозволяє їм переглядати список клієнтів.

Потоки подій

Основний потік

Оператор або спеціаліст обирає в головному меню пункт «Список клієнтів» після чого вебсайт відображає на екран список з усіма клієнтами.

Альтернативні потоки

Не знайдено жодних записів у даній вкладці - буде виведено повідомлення про те, що жодних записів в даній вкладці знайдено не було.

Спеціальні умови: спеціальні умови невизначені.

Передумови: передумови не визначені.

Постумови: постумови не визначені.

Додаткові зауваження: додаткових зауважень немає.

7) Редагування даних клієнтів

Назва варіанту використання: «Редагування даних клієнтів»

Короткий опис: варіант використання ініціюється активним суб'єктом «Спеціаліст» і дозволяє спеціалісту редагувати дані про клієнтів.

Потоки подій

Основний потік

Спеціаліст обирає в головному меню пункт “Список клієнтів” далі обирає необхідного клієнта і натискає “Редагувати дані” після чого спеціаліст заповнює усі необхідні поля вводу і підтверджує свій вибір.

Альтернативні потоки

Помилка у форматі даних - буде виведено повідомлення про помилку з поясненням помилки та запит на повторне введення даних користувачем.

Спеціальні умови: спеціальні умови невизначені.

Передумови: перед активізацією варіанта використання потрібно обрати у меню пункт «Список клієнтів».

Постумови: постумови не визначені.

Додаткові зауваження: додаткових зауважень немає.

8) Створення послуги

Назва варіанту використання: «Створення послуги»

Короткий опис: варіант використання ініціюється активним суб'єктом «Оператор» і дозволяє оператору створити послугу.

Потоки подій

Основний потік

Оператор обирає в головному меню пункт “Список клієнтів” і обирає потрібного клієнта після чого натискає “Створити послугу” далі оператор заповнює всі поля даними (у випадку помилки введення даних – виконується альтернативний потік) і підтверджує свій вибір. Вебсайт додає послугу у БД.

Альтернативні потоки

Помилка у форматі даних - буде виведено повідомлення про помилку з

поясненням помилки та запит на повторне введення даних користувачем.

Спеціальні умови: спеціальні умови невизначені.

Передумови: перед активізацією варіанта використання потрібно обрати у меню пункт “Список клієнтів”.

Постумови: постумови не визначені.

Додаткові зауваження: додаткових зауважень немає.

9)Перегляд списку послуг

Назва варіанту використання: «Перегляд списку послуг»

Короткий опис: варіант використання ініціюється активним суб’єктом «Спеціаліст» і дозволяє спеціалісту переглядати список послуг.

Потоки подій

Основний потік

Спеціаліст обирає в головному меню пункт “Список послуг” після чого вебсайт відображає на екран список з усіма послугами.

Альтернативні потоки

Не знайдено жодних записів у даній вкладці - буде виведено повідомлення про те, що жодних записів в даній вкладці знайдено не було.

Спеціальні умови: спеціальні умови невизначені.

Передумови: передумови невизначені.

Постумови: постумови не визначені.

Додаткові зауваження: додаткових зауважень немає.

10)Призначення спеціаліста послугі

Назва варіанту використання: «Призначення спеціаліста послугі»

Короткий опис: варіант використання ініціюється активним суб’єктом «Спеціаліст» і дозволяє спеціалісту призначити себе до послуги.

Потоки подій

Основний потік

Спеціаліст обирає в головному меню пункт “Список послуг” після чого обирає необхідну послугу і натискає «Обрати послугу» після чого вебсайт призначає спеціаліста до послуги.

Альтернативні потоки

Помилка у форматі даних - буде виведено повідомлення, що послуга уже зайнята іншим спеціалістом.

Спеціальні умови: спеціальні умови невизначені.

Передумови: перед активізацією варіанта використання потрібно обрати пункт у меню «Список послуг».

«Переглядати записи з бази даних».

Постумови: постумови не визначені.

Додаткові зауваження: додаткових зауважень немає.

11)Видалення заявки

Назва варіанту використання: «Видалення заявки»

Короткий опис: варіант використання ініціюється активним суб'єктом «Оператор» і дозволяє оператору видалити заявку з бази даних.

Потоки подій

Основний потік

Оператор обирає в головному меню пункт “Список заявок” після чого натискає «Видалити заявку». Вебсайт відображає вікно відображає вікно з опціями підтвердження та скасування вибору. Оператор підтверджує свій вибір і видаляє заявку із БД(у випадку помилки введення даних – виконується альтернативний потік).

Альтернативні потоки

Користувач скасував операцію видалення заявки.

Спеціальні умови: спеціальні умови невизначені.

Передумови: перед активізацією варіанта використання потрібно обрати потрібний пункт у меню “Список заявок».

Постумови: постумови не визначені.

Додаткові зауваження: додаткових зауважень немає.

12)Додавання працівника

Назва варіанту використання: «Додавання працівника»

Короткий опис: варіант використання ініціюється активним суб'єктом

«Адміністратор» і дозволяє адміністратору додавати працівника до бази даних.

Потоки подій

Основний потік

Адміністратор обирає в головному меню пункт «Додати працівника» після чого вебсайт відображає вікно з полями вводу. Адміністратор заповнює всі поля даними (у випадку помилки введення даних виконується альтернативний потік) і підтверджує свій вибір. Вебсайт додає працівника у БД.

Альтернативні потоки

Помилка у форматі даних - буде виведено повідомлення про помилку з поясненням помилки та запит на повторне введення даних користувачем.

Спеціальні умови: спеціальні умови невизначені.

Передумови: передумови не визначені.

Постумови: постумови не визначені.

Додаткові зауваження: додаткових зауважень немає.

13)Редагування даних працівника

Назва варіанту використання: «Редагувати дані працівника»

Короткий опис: варіант використання ініціюється активним суб'єктом «Адміністратор» і дозволяє адміністратору редагувати дані про працівників.

Потоки подій

Основний потік

Адміністратор обирає в головному меню пункт «Список працівників» і обирає опцію «Редагувати дані». Вебсайт відображає вікно з полями вводу. Адміністратор заповнює всі поля даними (у випадку помилки введення даних виконується альтернативний потік) і підтверджує свій вибір. Вебсайт додає даний запис у БД.

Альтернативні потоки

Помилка у форматі даних - буде виведено повідомлення про помилку з поясненням помилки та запит на повторне введення даних користувачем.

Спеціальні умови: спеціальні умови невизначені.

Передумови: перед активізацією варіанта використання потрібно обрати у меню пункт «Список працівників».

Постумови: постумови не визначені.

Додаткові зауваження: додаткових зауважень немає.

14)Видалення працівника

Назва варіанту використання: «Видалення працівника»

Короткий опис: варіант використання ініціюється активним суб'єктом «Адміністратор» і дозволяє адміністратору видаляти дані про працівників.

Потоки подій

Основний потік

Адміністратор обирає в головному меню пункт «Список працівників» і обирає опцію «Видалення працівника». Вебсайт відображає вікно з опціями підтвердження та скасування вибору. Якщо адміністратор підтверджує свій вибір, то вебсайт видаляє працівника з БД. Якщо адміністратор не підтверджує свій вибір, то вебсайт повертає адміністратора у вікно зі списком всіх працівників.

Альтернативні потоки

Помилка у форматі даних - буде виведено повідомлення про помилку з поясненням помилки та запит на повторне введення даних користувачем.

Спеціальні умови: спеціальні умови невизначені.

Передумови: перед активізацією варіанта використання потрібно обрати у меню пункт «Список працівників».

Постумови: постумови не визначені.

Додаткові зауваження: додаткових зауважень немає.

2.1.4 Розробка концептуальної моделі

Концептуальне проєктування – це створення змістовної моделі предметної галузі. Для створення концептуальної моделі використано діаграму «Сутність-Зв'язок». Концептуальна модель включає в себе опис понять предметної галузі та зв'язків між ними [6]. Інформаційна модель даних програмного забезпечення має на меті відобразити дані, які буде використовувати програма, тобто моделювання сутностей предметної галузі з відображенням зв'язків між ними. Для цих цілей слушно використати діаграму «Сутність-Зв'язок». Діаграма сутність зв'язок наведена на рисунку 2.3:

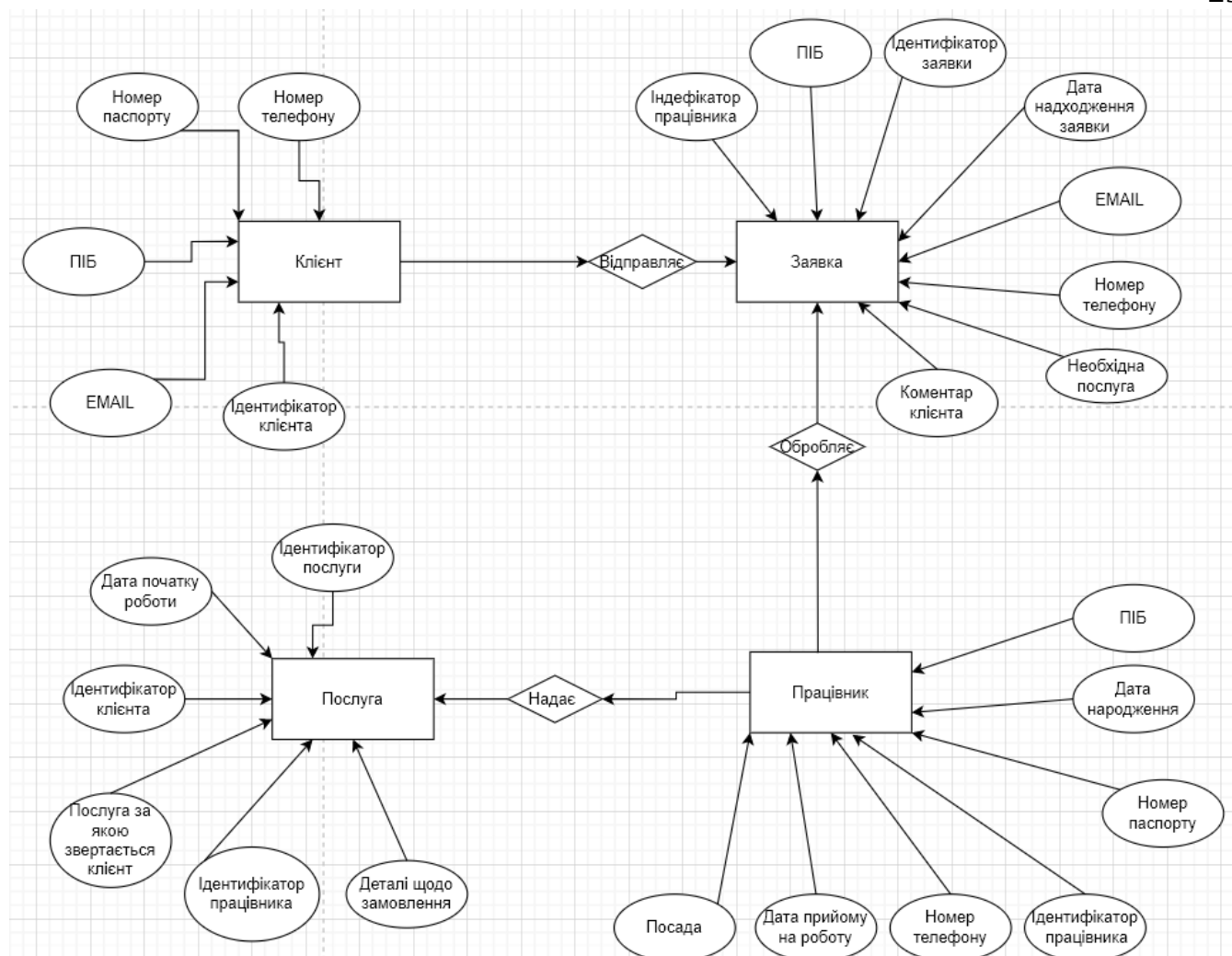


Рисунок 2.3 – Діаграма «Сутність-Зв'язок»

2.1.5 Розробка ескізу інтерфейсу

Проектування користувацького інтерфейсу надзвичайно важливий етап під час розробки програмного забезпечення. Саме на цьому етапі залежить фактична зручність та практичність у використанні розробленого програмного забезпечення. Тому дуже важливо, створюючи інтерфейс програмного забезпечення вебсайту системи для консультацій у банківській сфері «Legalist», задовольнити такі основні властивості інтерфейсу, як:

- ясність;
- послідовність;
- відповідність
- дружність;

Ураховуючи усі попередні властивості було створено ескізи інтерфейсу

програмного забезпечення вебсайту системи для консультацій у банківській сфері «Legalist». Ескізи інтерфейсу ПЗ представлені на рисунках 2.4 – 2.15.

The sketch shows a rectangular form with a header bar at the top. The header bar contains three elements: a box labeled 'Логотип' (Logo) on the left, the word 'Авторизація' (Authorization) in the center, and a box labeled 'Вийти' (Logout) on the right. Below the header, the form contains two input fields stacked vertically. The first is labeled 'Логін' (Login) and the second is labeled 'Пароль' (Password). Below these fields is a button labeled 'Увійти' (Login).

Рисунок 2.4 – Ескіз форми авторизації користувача

The sketch shows a rectangular form with a header bar at the top. The header bar contains three elements: a box labeled 'Логотип' (Logo) on the left, the word 'Вхід' (Entry) in the center, and a box labeled 'Вхід' (Entry) on the right. Below the header, the form contains three input fields arranged in a grid. The first row has two fields: 'ПІБ' (Full Name) and 'Номер телефону' (Phone Number). The second row has two fields: 'EMAIL' and 'Коментар' (Comment). The third row has one field labeled 'Послуга' (Service). Below these fields is a button labeled 'Відправити' (Send).

Рисунок 2.5 – Ескіз форми відправлення заявки

Рисунок 2.6 – Ескіз форми меню оператора

Рисунок 2.7 – Ескіз форми меню спеціаліста

Рисунок 2.8 – Ескіз форми меню адміністратора

ID заявки	Номер телефону	ПІБ	EMAIL	Необхідна послуга	Коментар клієнта	Дата надходження		
ID заявки	Номер телефону	ПІБ	EMAIL	Необхідна послуга	Коментар клієнта	Дата надходження	Прийняти заявку	Видалити заявку
ID заявки	Номер телефону	ПІБ	EMAIL	Необхідна послуга	Коментар клієнта	Дата надходження	Прийняти заявку	Видалити заявку
ID заявки	Номер телефону	ПІБ	EMAIL	Необхідна послуга	Коментар клієнта	Дата надходження	ID працівника	

Рисунок 2.9 – Ескіз форми списку заявок

Логотип					Меню		Вийти	
Список Клієнтів								
ID клієнта	ПІБ	EMAIL	Номер телефону	Номер паспорту	Редагувати дані про клієнта	Створити послугу		
ID клієнта	ПІБ	EMAIL	Номер телефону	Номер паспорту	Редагувати дані про клієнта	Створити послугу		
ID клієнта	ПІБ	EMAIL	Номер телефону	Номер паспорту	Редагувати дані про клієнта	Створити послугу		

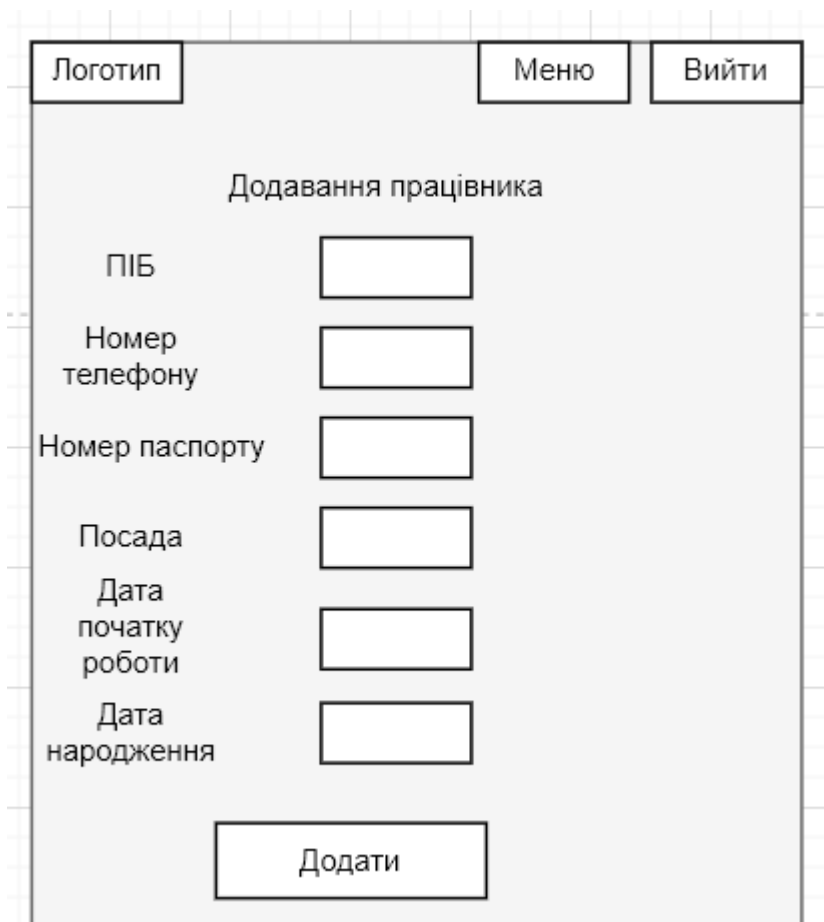
Рисунок 2.10 – Ескіз форми списку клієнтів

Логотип					Меню		Вийти	
Список Послуг								
ID послуги	ID клієнта	Послуга за якою звертається	Дата початку роботи	Деталі щодо замовлення	Обрати послугу			
ID послуги	ID клієнта	Послуга за якою звертається	Дата початку роботи	Деталі щодо замовлення	Обрати послугу			
ID послуги	ID клієнта	Послуга за якою звертається	Дата початку роботи	Деталі щодо замовлення	ID працівника			

Рисунок 2.11 – Ескіз форми списку послуг

Логотип							Меню		Вийти	
Список працівників										
ID працівника	ПІБ	Номер телефону	Номер паспорту	Посада	Дата народження	Дата прийому на роботу	Редагувати дані	Видалити працівника		
ID працівника	ПІБ	Номер телефону	Номер паспорту	Посада	Дата народження	Дата прийому на роботу	Редагувати дані	Видалити працівника		
ID працівника	ПІБ	Номер телефону	Номер паспорту	Посада	Дата народження	Дата прийому на роботу	Редагувати дані	Видалити працівника		
ID працівника	ПІБ	Номер телефону	Номер паспорту	Посада	Дата народження	Дата прийому на роботу	Редагувати дані	Видалити працівника		

Рисунок 2.12 – Ескіз форми списку працівників



Логотип Меню Вийти

Додавання працівника

ПІБ

Номер телефону

Номер паспорту

Посада

Дата початку роботи

Дата народження

Додати

Рисунок 2.13 – Ескіз форми додавання працівника



Логотип Меню Вийти

Додавання клієнта

ПІБ

EMAIL

Номер паспорта

Номер телефону

Додати

Рисунок 2.14 – Ескіз форми додавання клієнта

The sketch shows a web form titled "Створення послуги" (Service Creation). At the top, there are three buttons: "Логотип" (Logo), "Меню" (Menu), and "Вийти" (Exit). The main form area contains three input fields with labels: "Послуга" (Service), "Дата початку роботи" (Start date of work), and "Деталі щодо замовлення" (Details regarding the order). Below these fields is a large "Додати" (Add) button.

Рисунок 2.15 – Ескіз форми створення послуги

2.2 Технічний проєкт програмного забезпечення програмного забезпечення вебсайту системи для консультацій у банківській сфері «Legalist»

2.2.1 Логічна модель бази даних

Створивши логічну модель БД це створить більш зрозуміле представлення даних та вимог до розробки програмного забезпечення. Логічна модель бази даних ПЗ вебсайту системи для консультацій “Legalists” представлена на рисунку 2.16.

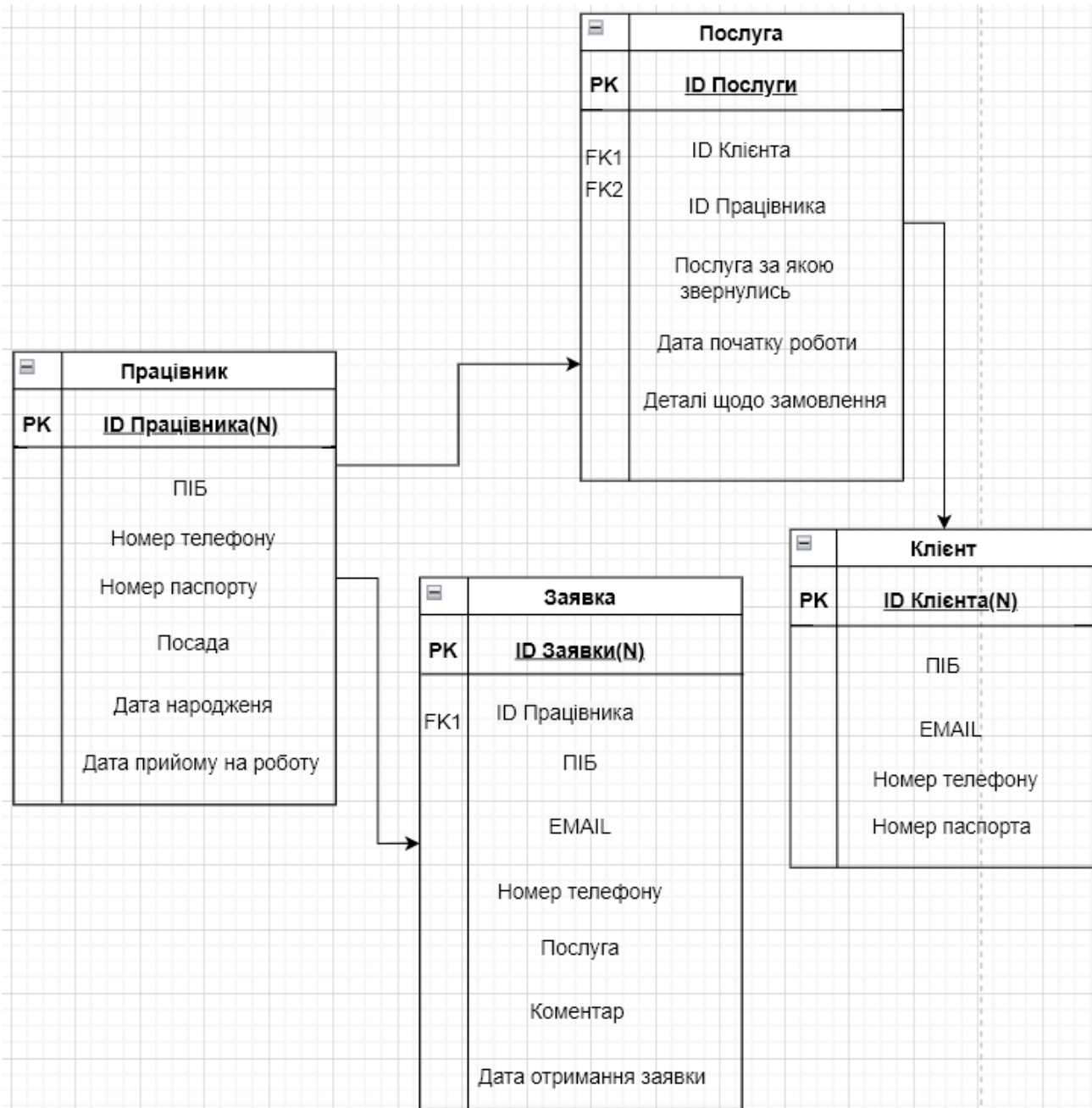


Рисунок 2.16 – Логічна модель БД

Атрибути таблиць та тип даних атрибутів наведено у таблицях 2.3–2.6

Таблиця 2.3 – Опис атрибутів таблиці «Заявки»

Атрибут	Тип даних
Ідентифікатор заявки	Число
Ідентифікатор працівника	Число
ПІБ	Текст
EMAIL	Текст
Номер телефону	Число
Необхідна послуга	Текст
Коментар клієнта	Текст
Дата надходження заявки	Дата

Таблиця 2.4 – Опис атрибутів таблиці «Клієнти»

Атрибут	Тип даних
Ідентифікатор клієнта	Число
ПІБ	Текст
EMAIL	Текст
Номер телефону	Число
Номер паспорта	Текст

Таблиця 2.5 – Опис атрибутів таблиці «Працівники»

Атрибут	Тип даних
Ідентифікатор працівника	Число
ПІБ	Текст
Номер телефону	Число
Номер паспорта	Текст
Посада	Текст
Дата народження	Дата
Дата прийому на роботу	Дата

Таблиця 2.6 – Опис атрибутів таблиці «Послуги»

Атрибут	Тип даних
Ідентифікатор послуги	Число
Ідентифікатор клієнта	Число
Ідентифікатор працівника	Число
Послуга за якою звертається клієнт	Текст
Дата початку роботи.	Дата
Деталі щодо замовлення	Текст

2.2.2 Діаграма класів програмного забезпечення “Legalist”

Діаграма класів програмного забезпечення описує внутрішню структуру класів, що моделюються. Діаграма класів є головним засобом для моделювання статичного представлення системи [6]. Діаграму класів програмного забезпечення «Legalist» наведено на рисунку 2.17.

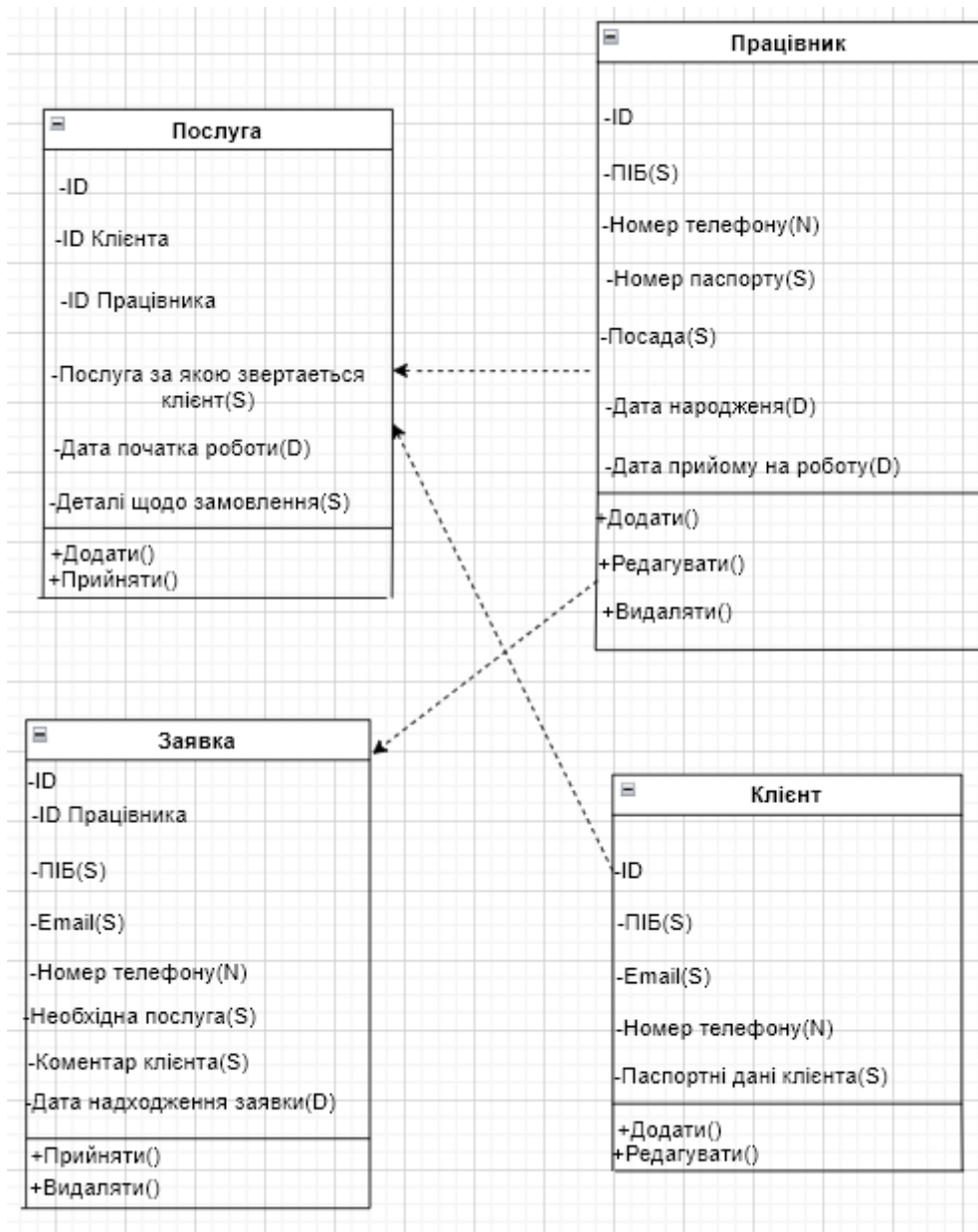


Рисунок 2.17 – Діаграма класів ПЗ

2.2.3 Динамічна модель програмного забезпечення

Динамічні моделі забезпечують демонстрацію поведінки системи. «Динамізм» цих моделей полягає в тому, що вони відображають зміни стану під час роботи системи [6].

Діаграма діяльності є особливою формою кінцевого автомата, що демонструє процес розрахунку та робочого процесу в ній. Діаграми діяльності програмного забезпечення «Legalist» наведено на рисунках 2.18-2.27.

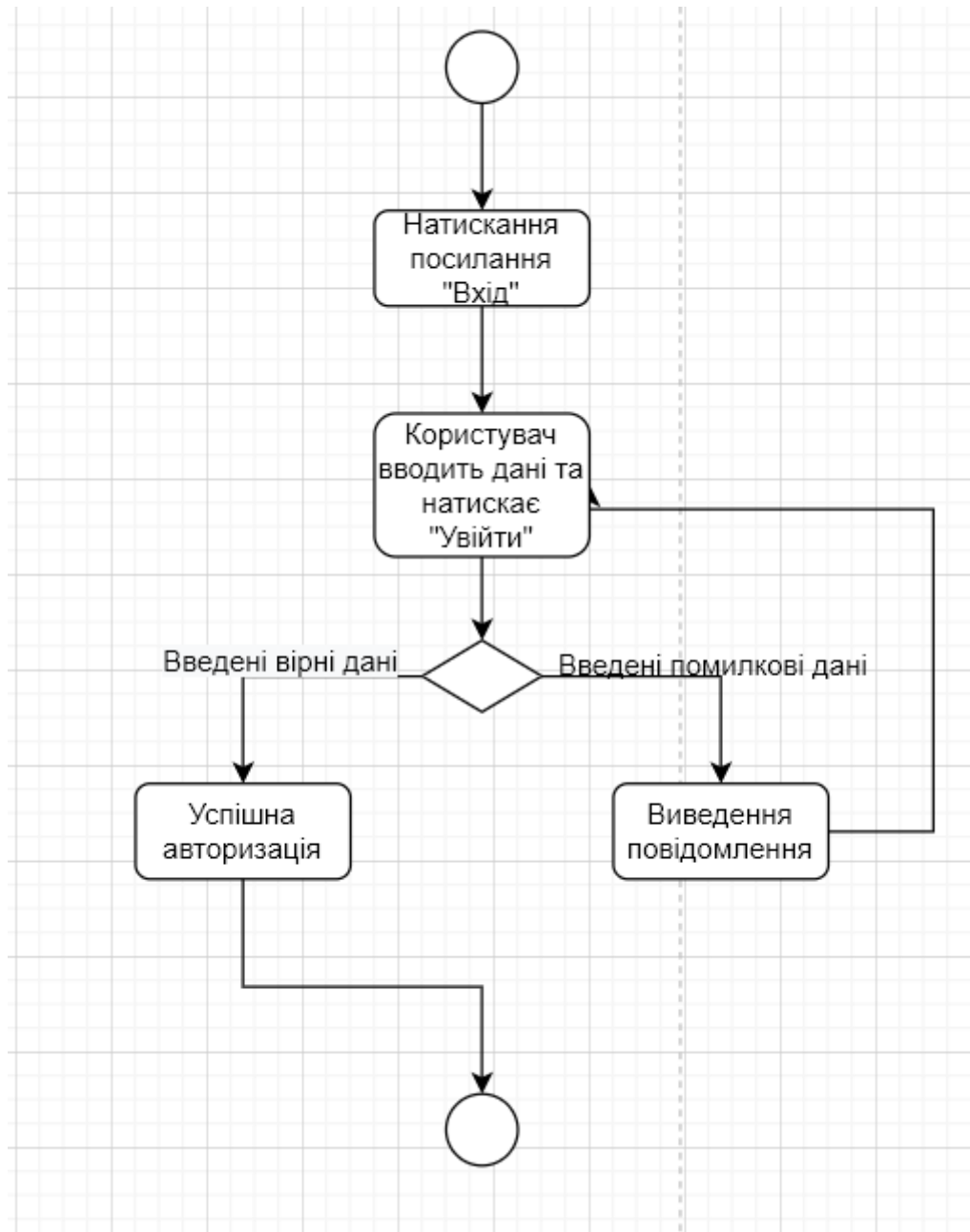


Рисунок 2.18 – Діаграма діяльності для варіанту використання “Авторизація”

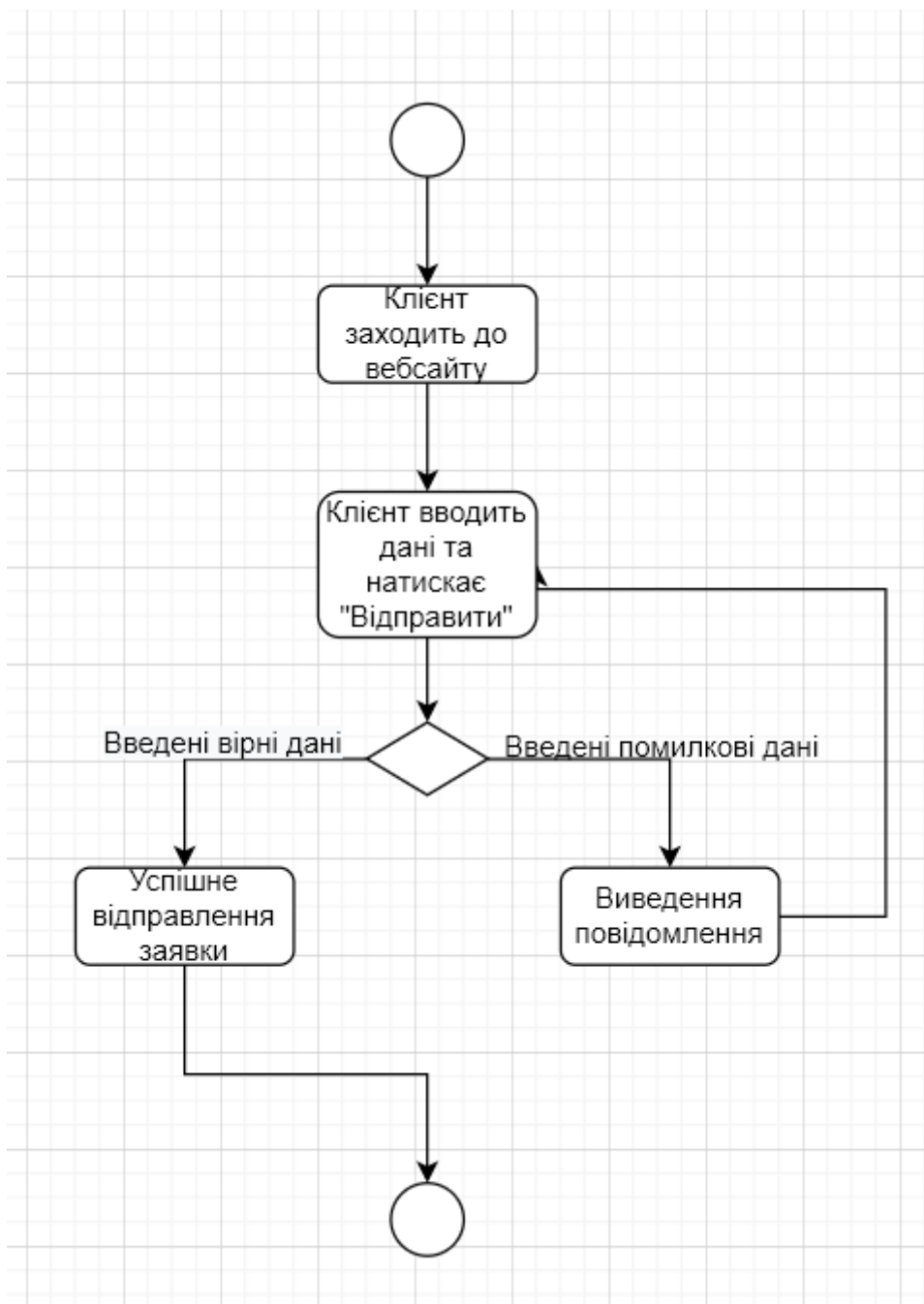


Рисунок 2.19 – Діаграма діяльності для варіанту використання “Відправлення заявки”



Рисунок 2.20 – Діаграма діяльності для варіанту використання “Перегляд списку заявок”



Рисунок 2.21 – Діаграма діяльності для варіанту використання “Призначення оператора заявці”



Рисунок 2.22 – Діаграма діяльності для варіанту використання “Видалення заявки”

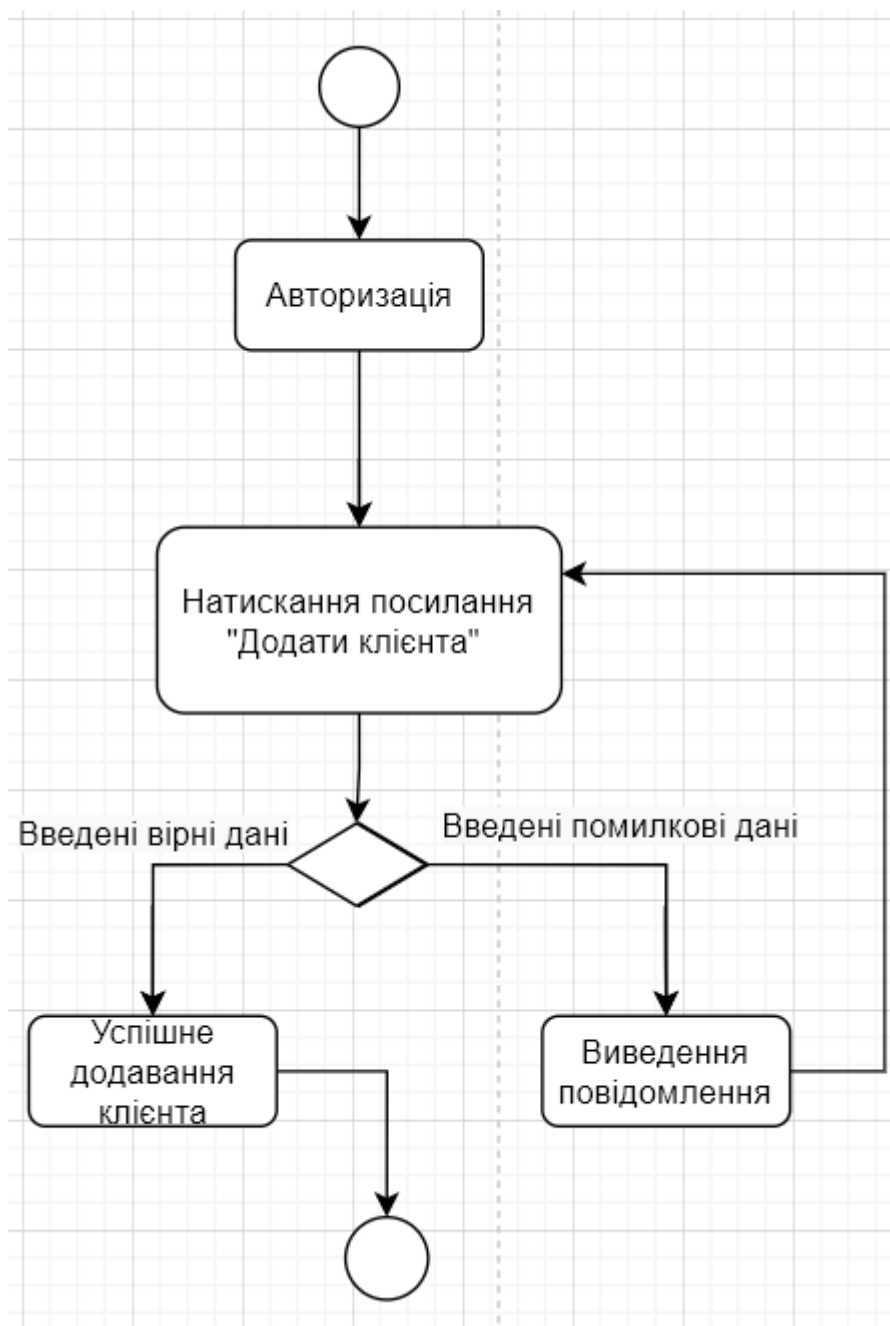


Рисунок 2.23 – Діаграма діяльності для варіанту використання “Додавання клієнта”



Рисунок 2.24 – Діаграма діяльності для варіанту використання “Перегляд списку клієнтів”



Рисунок 2.25 – Діаграма діяльності для варіанту використання “Створення послуги”

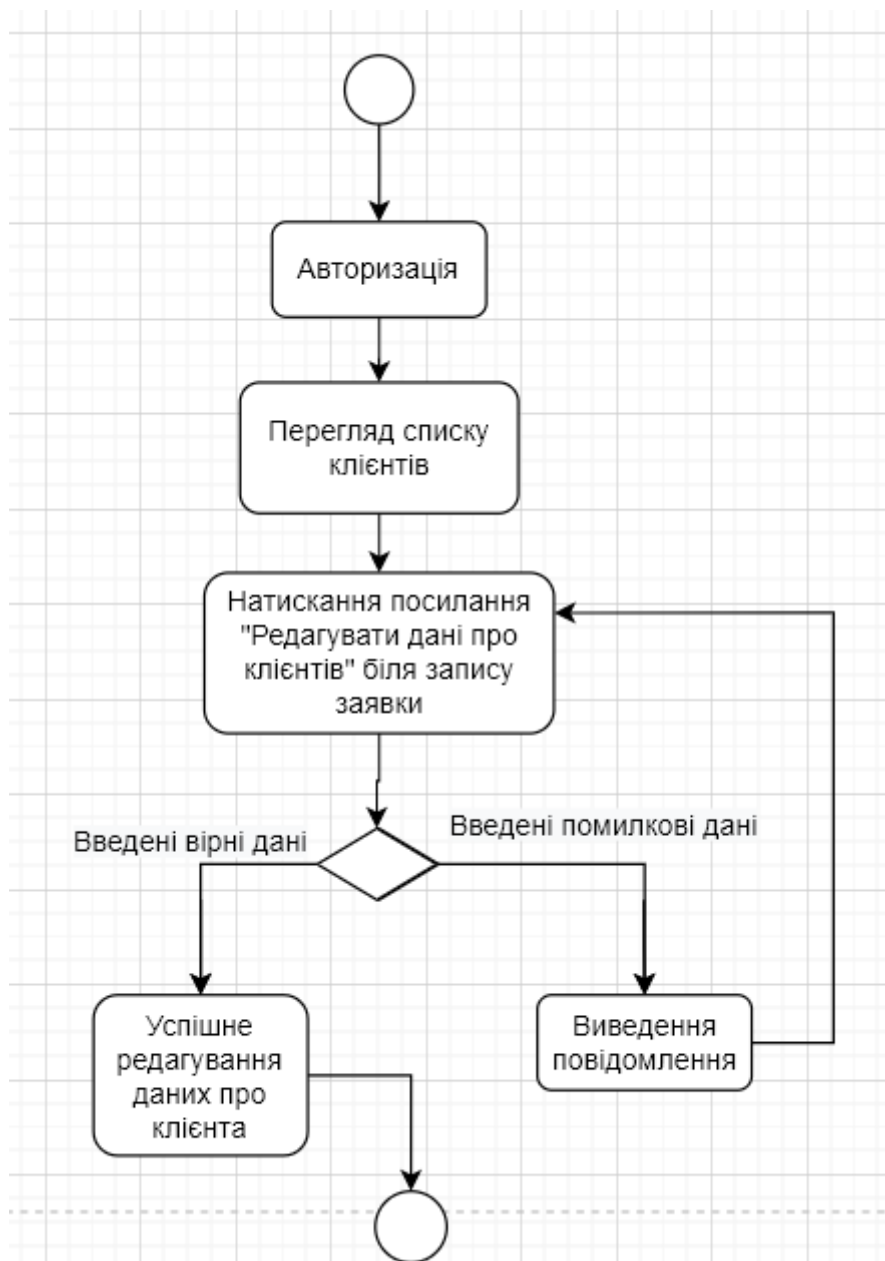


Рисунок 2.26 – Діаграма діяльності для варіанту використання “Редагувати дані про клієнтів”



Рисунок 2.27 – Діаграма діяльності для варіанту використання “Перегляд списку послуг”



Рисунок 2.28 – Діаграма діяльності для варіанту використання “Перегляд списку послуг”

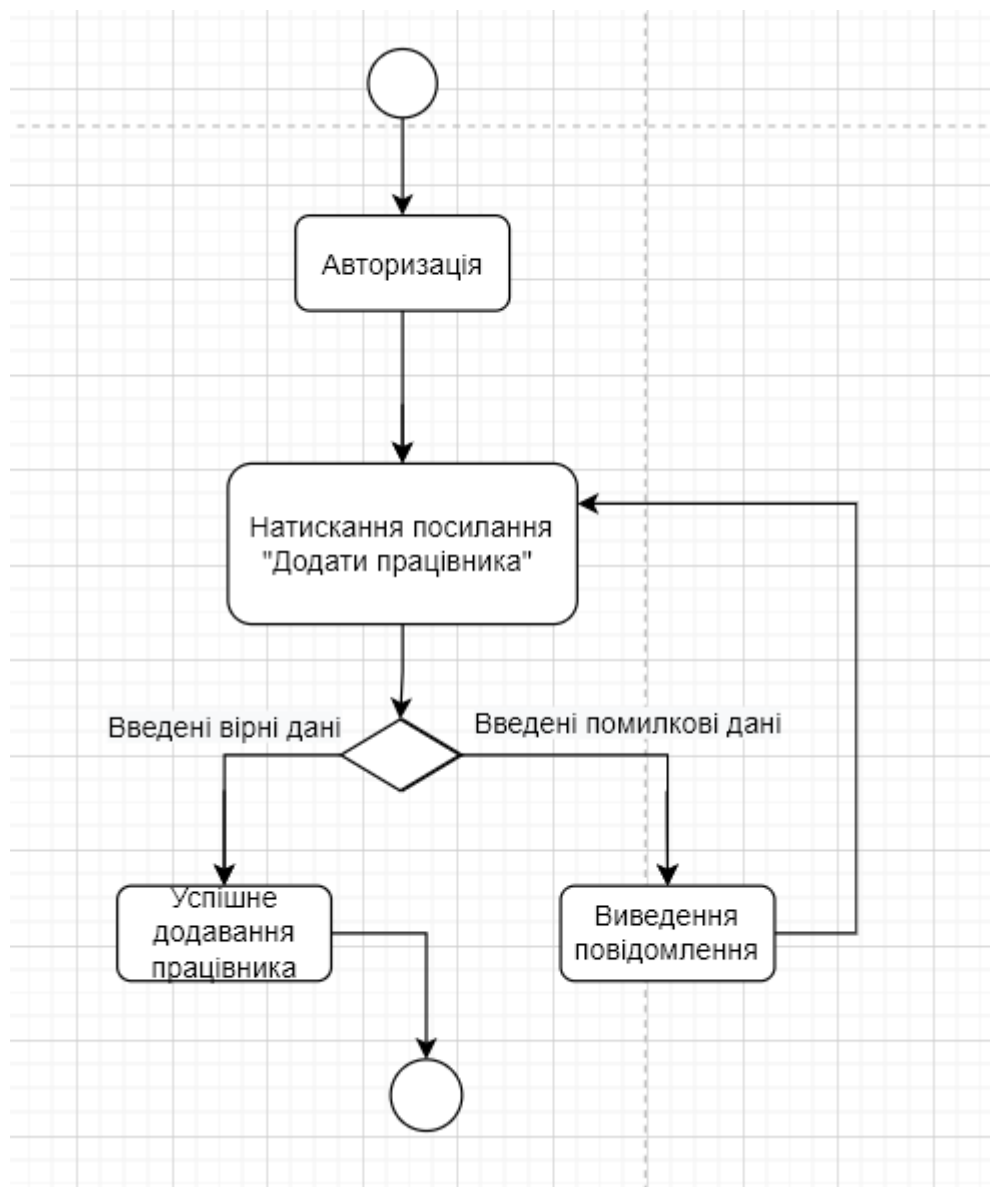


Рисунок 2.29 – Діаграма діяльності для варіанту використання “Додавання працівника”



Рисунок 2.30 – Діаграма діяльності для варіанту використання “Редагування даних працівника”



Рисунок 2.31 – Діаграма діяльності для варіанту використання “Видалення працівника”

Спроектвані діаграми на рисунках 2.18 – 2.31 є динамічними моделями ПЗ, яке розроблюється в даній кваліфікаційній роботі, і забезпечують представлення поведінки системи програмного забезпечення вебсайту системи для консультацій у банківській сфері «Legalist».

2.3 Робочий проєкт програмного забезпечення вебсайту системи для консультацій у банківській сфері «Legalist»

2.3.1 Обґрунтування вибору мови програмування та СКБД

Під час створення програмного забезпечення вебсайту системи для консультацій у банківській сфері «Legalist» іде розробка спочатку frontend частини, а потім backend. Frontend – це клієнтська сторона інтерфейсу яку користувач бачить під час роботи із програмною частиною сервісу. Основою для фронтенду є мова гіперрозмітки HTML та CSS яка буде використовуватись для створення стилів. Також під час розробки вебсайту було використано JavaScript для створення анімацій та деяких функціональних можливостей. Backend – це програмно-апаратна частина яка відповідає за функціонування внутрішньої частини сервісу. Саме під час розробки Backend частини буде створена авторизація користувачів, взаємодія з базою даних.

Обираючи мову програмування для створення вебсайту я зупинився на мові програмування PHP. Завдяки PHP швидкість роботи вебсайту буде високою, адже PHP створювався для розробки динамічних веб сторінок. Також дуже важливим пунктом є те, що код PHP дуже легко вбудовується у HTML. Саме тому я обрав мову PHP для створення ПЗ вебсайту для консультацій “Legalist”.

Під час розробки ПЗ необхідно було обрати СКБД для роботи із базами даних. СКБД буде надавати безпеку та зберігання цілісності даних. Я обрав СКБД MySQL оскільки вона надійна, швидка, проста у використанні та має обширний функціонал.

2.3.2 Фізична модель бази даних

Фізична модель бази даних розробляється з огляду на логічну модель. Фізична модель має на меті перейти від формальних імен сутностей та їх атрибутів, що використовувались на етапах концептуального та логічного проєктування, до фактичних імен сутностей та атрибутів, що будуть використані при розробці програмного забезпечення. Фізичну модель бази даних програмного забезпечення

вебсайту системи для консультацій у банківській сфері «Legalist» представлено на рисунку 2.32.

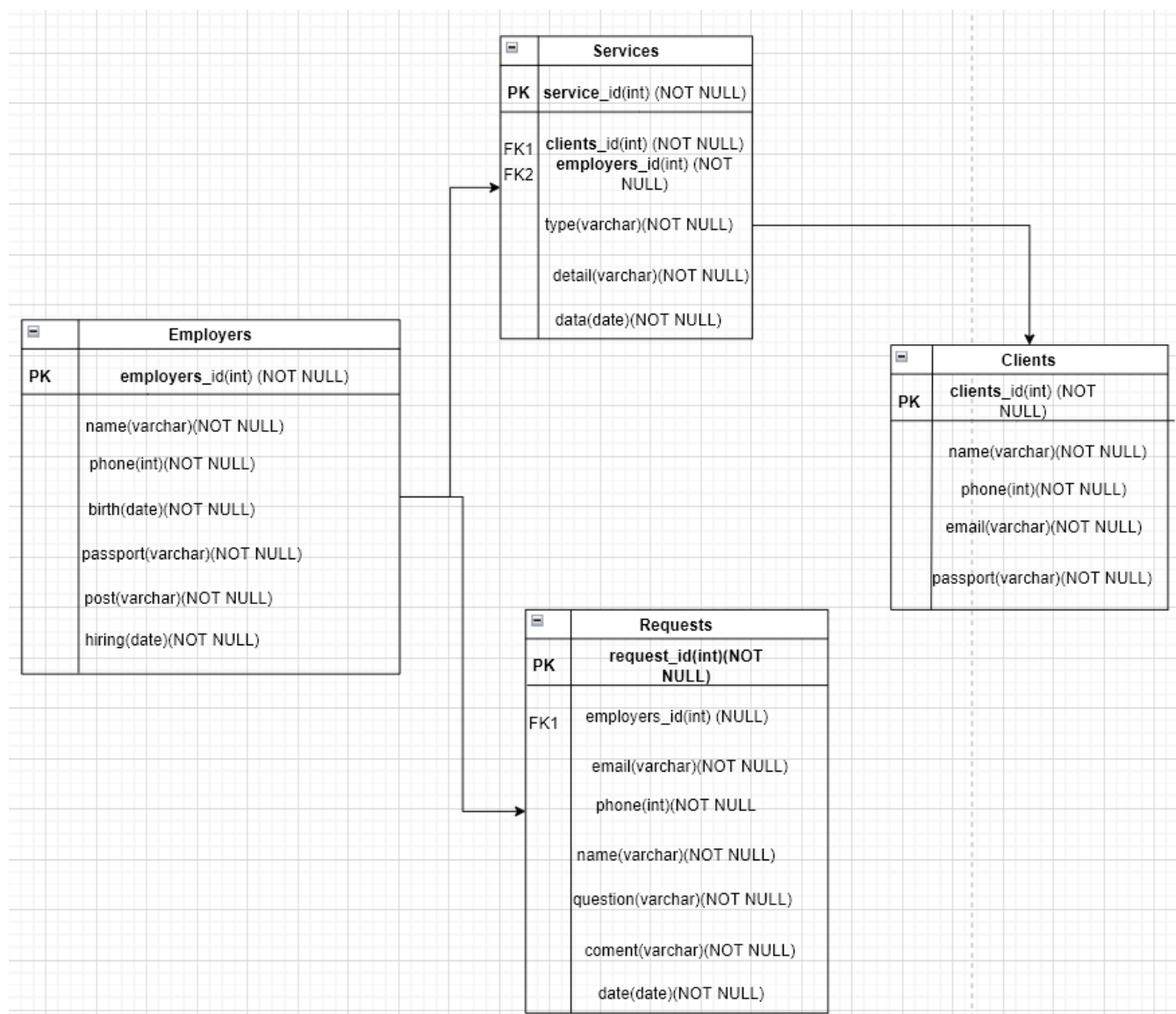


Рисунок 2.32 – Фізична модель бази даних

Побудувавши таблиці відповідності логічної та фізичної моделей БД ми усунемо будь-які розбіжності в іменуванні сутностей та їх атрибутів програмного забезпечення вебсайту системи для консультацій у банківській сфері «Legalist». Таблиці відповідності логічної та фізичної моделей бази даних представлені у таблицях 2.7–2.10.

Таблиця 2.7 – Таблиця відповідності логічної та фізичної моделей бази даних для заявок

Ім'я атрибуту в фізичній моделі	Ім'я атрибуту в логічній моделі	Тип даних	Розмір	Ключ	Обов'язковість поля
1	2	3	4	5	6
request_id	ID заявки	int	4	PK	NOT NULL
clients_id	ID працівника	int	4	FK1	NULL
email	EMAIL	varchar	255		NOT NULL
phone	Номер телефону	int	4		NOT NULL
name	ПІБ	varchar	255		NOT NULL
question	Питання за яким звернувся клієнт	varchar	255		NOT NULL
coment	Коментар клієнта	varchar	255		NOT NULL
date	Дата надходження заявки	date	3		NOT NULL

Таблиця 2.8 – Таблиця відповідності логічної та фізичної моделей бази даних для клієнтів

clients_id	ID клієнта	int	4	PK	NOT NULL
name	ПІБ	varchar	255		NOT NULL
phone	Номер телефону	int	4		NOT NULL
email	EMAIL	varchar	255		NOT NULL
passport	Номер паспорта	varchar	255		NOT NULL

Таблиця 2.9 – Таблиця відповідності логічної та фізичної моделей бази даних для послуг

service_id	ID послуги	int	4	PK	NOT NULL
clients_id	ID клієнта	int	4	FK1	NOT NULL
clients_id	ID працівника	int	4	FK2	NULL
type	Вид послуги	varchar	255		NOT NULL
detail	Деталі щодо замовлення	varchar	255		NOT NULL
data	Дата початку роботи	date	3		NOT NULL

Таблиця 2.10 – Таблиця відповідності логічної та фізичної моделей бази даних для працівників

clients_id	ID працівника	int	4	PK	NOT NULL
name	ПІБ	varchar	255		NOT NULL
phone	Номер телефону	int	4		NOT NULL
birth	Дата народження	date	3		NOT NULL
passport	Номер паспорта	varchar	255		NOT NULL
Post	Посада	varchar	255		NOT NULL
hiring	Дата прийому на роботу	date	3		NOT NULL

2.3.3 Тестування основних класів еквівалентності ПЗ ‘Legalist’

Тестування – це процес технічного дослідження для перевірки фактичної роботи у порівнянні з очікуваною. Тестування передбачає спеціальний підбір вхідних даних з метою проведення повної кількості тестів.

Під час розробки тестів у цій роботі було використано метод на основі аналізу специфікації метод еквівалентного розбиття. Еквівалентне розбиття передбачає визначення такої множини вхідних даних, відносно якої поведінка програми буде однаковою. Класи еквівалентності поділяються на правильні (дозволені) та неправильні (недозволені).

Тестування функції «Додати працівника»

Класи еквівалентності наведені в таблиці 2.11.

Таблиця 2.11 – Класи еквівалентності для функції ”Додати працівника”

Вхідні дані	Правильні класи	Неправильні класи
ПІБ	1.Символи кирилиці	2. Символи, окрім кирилиці
Номер телефону	3. Будь-які числа	4. Символи, окрім чисел
Номер паспорта	5. Будь-які числа	6. Символи, окрім чисел
Дата початку роботи	7. Дата (рядок у форматі «дд.мм.рррр»)	8. Символи або числа у будь-якому форматі, окрім зазначеного
Дата народження	9. Дата (рядок у форматі «дд.мм.рррр»)	10. Символи або числа у будь-якому форматі, окрім зазначеного

Таблиця 2.12 – Тести з правильних класів еквівалентності

№ Тесту	№ покритого класу еквівалентності	Вхідні дані	Результат тестування
1	1	Жмишенко Валерій Альбертович	В результаті тестування працівника успішно додано до БД.
	3	380930651076	
	5	00567984	
	7	08.02.2021	
	9	05.07.1975	

Тести з неправильних класів еквівалентності наведено в таблиці 2.13.

Таблиця 2.13 – Тести з неправильних класів еквівалентності

№ Тесту	№ покритого класу еквівалентності	Вхідні дані	Результат
1	2	Rtetri34	«Помилка у форматі даних»
2	4	eeee	«Помилка у форматі даних»
3	6	апрнн	«Помилка у форматі даних»
4	8	1996,45,21 1	«Помилка у форматі даних»
5	10	10151996	«Помилка у форматі даних»

Тестування функції «Додати клієнта»

Класи еквівалентності наведені в таблиці 2.14.

Таблиця 2.14 – Класи еквівалентності для функції ”Додати клієнта”

Вхідні дані	Правильні класи	Неправильні класи
ПІБ	1. Символи кирилиці	2. Символи, окрім кирилиці
EMAIL	3. Користувач@домен	4. Строка без символу @
Номер паспорта	5. Будь-які числа	6. Символи, окрім чисел
Номер телефону	7. Будь-які числа	8. Символи, окрім чисел

Таблиця 2.15 – Тести з правильних класів еквівалентності

№ Тесту	№ покритого класу еквівалентності	Вхідні дані	Результат тестування
1	1	Головенко Андрій Кирилович	В результаті тестування клієнта успішно додано до БД.
	3	golovenkooo@gmail.com	
	5	056386592	
	7	380951045687	

Тести з неправильних класів еквівалентності наведено в таблиці 2.16.

Таблиця 2.16 – Тести з неправильних класів еквівалентності

№ Тесту	№ покритого класу еквівалентності	Вхідні дані	Результат
1	2	Андрій23	«Помилка у форматі даних»
2	4	Golovenko gmail.com	«Помилка у форматі даних»
3	6	Нпппп	«Помилка у форматі даних»
4	8	івамсч	«Помилка у форматі даних»

Тестування функції «Створити послугу»

Класи еквівалентності наведені в таблиці 2.17.

Таблиця 2.17 – Класи еквівалентності для функції «Створити послугу»

Вхідні дані	Правильні класи	Неправильні класи
Послуга	1.Символи кирилиці	2. Символи, окрім кирилиці
Дата початку роботи	3. Дата (рядок у форматі «дд.мм.рррр»)	4 Символи або числа у будь-якому форматі, окрім зазначеного
Деталі щодо замовлення	5.Символи кирилиці	6. Символи, окрім кирилиці

Таблиця 2.18 – Тести з правильних класів еквівалентності

№ Тесту	№ покритого класу еквівалентності	Вхідні дані	Результат тестування
1	1	Консультація для отримання кредиту	В результаті тестування послугу успішно додано до БД.
	3	08.02.2022	
	5	Допомога для бізнесу	

Тести з неправильних класів еквівалентності наведено в таблиці 2.19.

Таблиця 2.19 – Тести з неправильних класів еквівалентності

№ Тесту	№ покритого класу еквівалентності	Вхідні дані	Результат
1	2	555	«Помилка у форматі даних»
2	4	Ппр42	«Помилка у форматі даних»
3	6	Інвестиції2 !	«Помилка у форматі даних»

2.3.4 Випробування програмного забезпечення "Legalist"

Відповідність програмної документації вимогам до програмного забезпечення перевіряє представник служби, відповідальний за експлуатацію. При перевірці склад і повнота наданої розробником документації на програмне забезпечення порівнюється з переліком програмної документації наведеним нижче:

- 1) Технічне завдання.
- 2) Текст програми.
- 3) Опис програми.
- 4) Програма та методика випробувань.
- 5) Інструкція користувача [7].

Тест вважається завершеним у випадку якщо склад та комплексність відповідає програмній документації яка була представлена розробником, переліком програмної документації, наведеному у зазначеному вище переліку [7].

Перевірка працездатності програми виконується згідно до вимог функціональних характеристик ПЗ. Тест вважається виконаним у разі відповідності складу та послідовності виконаних дій до функціональних характеристик. Під час процесу тестування була перевірена функціональність за темою «Розробка програмного забезпечення вебсайту для банківської системи "Legalist"». У таблиці 2.20 вказано перелік випробувань основних функціональних можливостей.

Таблиця 2.20 - Методика випробування

№	Дія	Результат який повинен бути	Результат перевірки	Зауваження
1	Перевірка правильності Введення даних при авторизації	Програма повинна перевірити введені дані з записами у БД	Виконано	
2	Перевірка правильності додавання нового працівника	Програма повинна перевірити коректність внесених даних	Виконано	
3	Перевірка правильності додавання нового клієнта	Програма повинна перевірити коректність внесених даних	Виконано	
4	Перевірка правильності додавання нової послуги	Програма повинна перевірити коректність внесених даних	Виконано	
5	Перевірка правильності внесених даних при редагуванні даних про клієнтів або працівників в БД	Програма повинна перевірити коректність внесених даних	Виконано	
6	Перевірка правильності видалення записів із БД	Програма повинна перевірити введені дані з записами у БД	Виконано	

3 РЕЗУЛЬТАТИ РОЗРОБКИ

За результатами виконання кваліфікаційної роботи було розроблено програмне забезпечення вебсайту системи для консультацій у банківській сфері "Legalist". Завдяки аналізу аналогів ПЗ були виявлені їх недоліки та переваги, що допомогло створити більш досконале ПЗ. Програмне забезпечення вебсайту системи для консультацій у банківській сфері «Legalist» дозволяє клієнту відправляти заявку для її подальшої обробки. Оператору дозволяє авторизуватись, переглядати, приймати, видаляти заявки, а також додавати клієнтів та створювати послуги. Спеціалісту дозволяє авторизуватись, редагувати дані про клієнтів, переглядати список послуг та призначати себе до послуг. Адміністратор може авторизуватись, переглядати список працівників додавати, редагувати та видаляти дані про працівників.

Результати аналізу предметної області були формалізовані завдяки створенню діаграми варіантів використання та специфікації варіантів використання. Також було виконано моделювання за допомогою діаграми діяльності для основних варіантів використання.

Під час виконання роботи було здійснено постановку задачі та створено технічне завдання для розробки програмного забезпечення вебсайту системи для консультацій у банківській сфері "Legalist", що наведене в додатку А «Технічне завдання».

Під час розробки кваліфікаційної роботи було створено ескізний, технічний та робочий проєкти.

Під час розробки ескізного проєкту було обрано мову моделювання, за допомогою обраної мови моделювання було створено концептуальну модель ПЗ та моделі варіантів використання, а також створено специфікацію для кожного варіанту використання і розроблено ескіз інтерфейсу користувача.

Під час розробки технічного проєкту була створена логічна модель бази даних, діаграма класів для статичного представлення системи та діаграми діяльності та станів системи для динамічного представлення системи.

Під час розробки робочого проєкту було обґрунтовано вибір інструментів розробки та СКБД, створено фізичну модель бази даних, а також було проведено тестування та випробування програмного забезпечення, яке показало повну працездатність та відповідність всім вимогам, які були представлені у технічному завданні для розробки програмного забезпечення вебсайту системи для консультацій у банківській сфері "Legalist. Для експлуатації програмного забезпечення також було розроблено інструкцію користувача.

Текст програми наведено у додатку Б – «Текст програми»

Опис програми наведено у додатку В – «Опис програми».

Інструкцію користувача наведено в додатку Г – «Інструкція користувача».

Програма та методика випробувань програмного забезпечення наведена в додатку Д – «Програма та методика випробувань ПЗ».

4 ОХОРОНА ПРАЦІ

Охорона праці це система правових, соціально-економічних, організаційно-технічних, санітарно-гігієнічних і лікувально-профілактичних заходів та засобів, спрямованих на збереження життя, здоров'я і працездатності людини у процесі трудової діяльності [8].

Мета охорони праці - забезпечення безпечних, нешкідливих і сприятливих умов праці через вирішення багатьох складних завдань, основними з яких є:

- проектування підприємств, технологічних процесів і конструювання обладнання з обов'язковим виконанням вимог охорони праці;
- знаходження оптимальних співвідношень між різними факторами виробничого середовища, що дозволяє забезпечити мінімум несприятливого впливу їх на здоров'я працівників;
- встановлення, законодавче оформлення визначених норм кожного з несприятливих або небезпечних факторів, систематичний контроль за їх застосуванням;
- розробка конкретних заходів щодо покращення умов праці та забезпечення її безпеки на основі застосування у виробництві новітніх досягнень науки і техніки;
- застосування раціональних засобів захисту працівників від впливу несприятливих факторів виробничого середовища, а також втілення організаційних заходів, які нейтралізують або послаблюють ступінь їх впливу на організм людини;
- розробка та застосування методів і засобів оцінки ефективності заходів з охорони праці, що плануються і здійснюються [8].

Основним законом в області охорони праці є Закон України «Про охорону праці». Дія Закону України «Про охорону праці» поширюється на всіх юридичних та фізичних осіб, які відповідно до законодавства використовують найману працю, та на всіх працюючих [9].

4.1 Аналіз шкідливих та небезпечних факторів на робочому місці

Під час укладання трудових договорів роботодавець повинен поінформувати Оператора під розписку про умови праці та про наявність на його робочому місці небезпечних і шкідливих виробничих факторів, які ще не усунуто, можливі наслідки їх впливу на здоров'я та про права Оператора на пільги та компенсації за роботу в таких умовах відповідно до законодавства і колективного договору [9].

Небезпечний виробничий фактор середовища і трудового процесу, що може бути причиною гострого захворювання (отруєння), раптового різкого погіршення здоров'я або смерті [10].

Шкідливий виробничий фактор середовища або трудового процесу, вплив якого на Оператора за певних умов (інтенсивність, тривалість дії тощо) може спричинити професійне або виробниче обумовлене захворювання, тимчасове або стійке зниження працездатності, підвищення частоти соматичних та інфекційних захворювань, призвести до порушення здоров'я як Оператора, так і його нащадків [10].

В таблиці 4.1 перераховані шкідливі та небезпечні виробничі фактори, з якими можуть зіткнутися під час роботи працівники банківської системи [11].

Таблиця 4.1 – Шкідливі та небезпечні виробничі фактори [11]

Найменування фактору	Можливі джерела його виникнення	Характер дії
1	2	3
Небезпека ураження електричним струмом	Мережа живлення	Небезпечний
Пожежонебезпечність приміщень	Наявність матеріалів, що загоряються і джерел запалення (електроапаратура)	Небезпечний та шкідливий
Електромагнітне випромінювання в тому числі й рентгенівське	ЕПТ (Дисплей є джерелом рентгенівського, радіочастотного, ультрафіолетового й інфрачервоного випромінювання та випромінювання звукового діапазону)	Шкідливий
Несприятлива освітленість	Недостатнє штучне і природне освітлення	Шкідливий
Психофізіологічні напруження	Монотонність праці, перенапруженість зорових аналізаторів, розумова напруженість, незручність і статичність пози	Шкідливий

Тривала й інтенсивна робота на комп'ютері може стати джерелом важких професійних та звичайних захворювань таких як:

- неправильна постава це призводить до подальшого розвитку викривлення хребта сколіозу, лордозу, кіфозу, а як результат головні болі, болі в області шиї та всього хребта, болі в області таза. Неправильно положення ніг може привести до артриту (запалення суглобів), артрозу (деформації);
- тунельний синдром найвідоміше захворювання людей працюють за комп'ютером. Воно ж синдром зап'ястного каналу. Тунельний синдром проявляється після кількох годин напруженої роботи за комп'ютером;
- геморой - хвороба аналогічна варикозного розширення вен. Причиною хвороби може бути застій крові у венах;
- розвиток короткозорості через те, що екран монітора за контрастом вище, ніж навколишні об'єкти розвивається короткозорість;
- порушення фокусування наслідком напруженої роботи за монітором є порушення фокусування, яка може бути викликана перенапруженням очних м'язів;
- сухість очей через рефлекс, заснованого на тому, що при погляді на джерело світла очей починає менше моргати виникає «обсушування» рогівки ока яке призводить до очних болів [11].

4.2 Заходи щодо зменшення впливу шкідливих факторів

Працюючи за комп'ютером, рекомендуємо дотримуватися правил тривалості роботи, правильної постави, розміру шрифтів та зображень, вимог до приміщення тощо.

Принципи правильної роботи за комп'ютером:

- у робочому приміщенні (кімнаті), де встановлені комп'ютери, щодня потрібно виконувати вологе прибирання;
- приміщення, у якому знаходяться комп'ютери, потрібно провітрювати щогодини;
- після кожного часу роботи рекомендується робити десяти хвилинну

перерву, яку зручно суміщати з провітрюванням. За будь-яких умов безперервна робота за комп'ютером для дорослої людини не повинна перевищувати двох годин. Під час перерви не варто читати або дивитися телевізор;

- необхідно постійно слідкувати за станом екрана монітора: він має бути чистим, без плям та пилу;

- потрібно слідкувати за поставою: ноги повинні твердо стояти на підлозі чи на спеціальній підставці; стегна повинні бути розташовані під прямим кутом до тулуба, а гомілки під прямим кутом до стегон; сидіти потрібно прямо або злегка нахилившись вперед; відстань від очей до екрана монітора не менше 55-60 см; центр екрана має знаходитися на рівні очей чи трохи нижче; рекомендується хоча б раз на день виконувати гімнастику для очей;

- щоб попередити „синдром сухого ока”, потрібно моргати кожні 3-5 секунд;

- не використовувати звичайний телевізор замість монітора;

- у процесі роботи рекомендується періодично (приблизно раз на 20-30 хвилин) переводити погляд з екрана на найбільш віддалений предмет у кімнаті, а ще краще – на віддалений об'єкт за вікном;

- якщо з'явилося відчуття втоми, напруження, сонливості, тяжкості в очах, потрібно припинити роботу та хоча б трохи відпочити [12].

Рекомендовані умови для роботи за комп'ютером:

- твердий стілець. край стільця не повинен тиснути на судини під колінами;

- відстань до монітора повинна бути 50-70 см;

- перерву в сидячій роботі, 15-20 хвилин кожні 1-2 години;

- правильне освітлення робочого місця. при слабкому світлі очі напружуються і болять;

- потрібно закривати очі для відпочинку. час від часу відводите очі вбік, щоб дати відпочити своєму зору [12].

ВИСНОВКИ

У кваліфікаційній роботі було розв’язано практичну задачу з інженерії програмного забезпечення, що характеризується комплексністю та невизначеністю умов: розроблене програмне забезпечення вебсайту системи для консультацій у банківській сфері «Legalist».

За результатами виконання кваліфікаційної роботи було виконано аналіз та постановку задачі на розробку програмного забезпечення вебсайту системи для консультацій у банківській сфері «Legalist», розроблено проєкт програмного забезпечення, а також представлені результати розробки, тестування та спеціальний розділ з охорони праці.

Також у даній кваліфікаційній роботі було розроблено технічне завдання, опис програми, програму та методику випробувань програмного продукту та інструкцію користувача за темою кваліфікаційної роботи “Розробка програмного забезпечення вебсайту для банківської системи "Legalist".

СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ

1. Методи створення розробки вебсайтів URL: <https://sites.google.com/site/tz5103voinovakateryna/metodi-rozrobki-web-sajtiv>.
2. Інформаційний сервіс надання консультацій у банківській сфері місті Києві «u-grand.com.ua». URL: <http://u-grand.com.ua> (дата звернення: 04.03.2022).
3. Інформаційний сервіс надання консультацій у банківській сфері у Польщі та Україні «enomi.pl». URL: <https://enomi.pl/ru/home> (дата звернення: 04.03.2022).
4. Буч Г., Рамбо Д., Якобсон І. Введення в UML від творців, видання. ДМК-Прес 2010 р.
5. Карпова Т. С. Базы данных: модели, разработка, реализация : уч. пособ. СПб : Питер, 2001. 304 с.
6. Орлов С. А. Технологии разработки программного обеспечения. СПб.: Питер. 2002. 464 с.
7. Методика проведення перевірки комплектності програмної документації. URL: <https://ukrbukva.net/page,11,108340-Baza-dannyh-tehnologicheskogo-oborudovaniya-dlya-proizvodstva-pechatnyh-plat.html> (дата звернення: 04.05.2022).
8. Про охорону праці : Закон України від 21.11.2002 р. № 229-IV. Дата оновлення: 05.02.2022. URL: <https://zakon.rada.gov.ua/laws/main/2694-12#Text>. (дата звернення: 12.03.2022).
9. Основи законодавства з охорони праці в області. URL: https://studopedia.su/7_23399_ponyattya-meta-i-zavdannya-ohoroni-pratsi.html (дата звернення: 12.03.2022).
10. Гігієнічна класифікація праці за показниками шкідливості та небезпечності факторів виробничого середовища, важкості та напруженості трудового процесу : Державні санітарні норми та правила від 08.04.2014 р. №248. Дата оновлення: 08.04.2014. URL: <https://zakon.rada.gov.ua/laws/show/z0472-14#n14>. (дата звернення: 12.03.2022).
11. Охорона праці та навколишнього середовища. URL: <https://smekni.com/a/298437-2/okhorona-prats-navkolishnogo-seredovishcha-2>. (дата

звернення: 12.03.2022).

12. Правила безпечної роботи на комп'ютері. URL: <https://pedcollege.kiev.ua/index.php/77-robota-koledzhu/okhorona-pratsi/589-pravyla-bezpechnoi-roboty-na-kompiuteri>. (дата звернення: 12.03.2022).

ДОДАТОК А ТЕХНІЧНЕ ЗАВДАННЯ

ВСТУП

Найменування програмного забезпечення вебсайту системи для консультацій у банківській сфері «Legalist». Область застосування банківська сфера.

1 ПІДСТАВА ДЛЯ РОЗРОБКИ

Підставою для розробки програмного забезпечення є завдання з бакалаврської кваліфікаційної роботи.

1.1 Найменування та умовне позначення теми розробки

Найменування розробки – Розробка програмного забезпечення для консультацій у банківській сфері «Legalists»

2 ПРИЗНАЧЕННЯ РОЗРОБКИ

2.1 Експлуатаційне призначення

Експлуатаційним призначенням програмного забезпечення вебсайту для консультацій банківської системи “Legalists” є автоматизація отримання заявок від клієнта їх зберігання, обробка та надання клієнту необхідної допомоги та послуг.

2.2 Функціональне призначення

Функціональним призначенням програмного забезпечення для консультацій у банківській сфері «Legalists» є автоматизація процесів:

- авторизація користувачів;
- призначення оператора заявці;
- призначення спеціаліста послугі;
- зберігання та використання інформації про клієнтів;
- зберігання та використання інформації про працівників.

3 ВИМОГИ ДО ПРОГРАМНОГО ЗАБЕЗПЕЧЕННЯ

3.1 Вимоги до функціональних характеристик

3.1.1 Вимоги до складу виконуваних функцій

Розроблена програма має надавати можливість виконувати наступні функції.

- Авторизація;
- Відправлення клієнтом заявки;
- Перегляд списку заявок;
- Призначення оператора заявці;
- Видалення заявки;
- Додавання клієнта;
- Перегляд списку клієнтів;
- Створення послуги;
- Редагування даних клієнта;
- Перегляд списку послуг;
- Призначення спеціаліста послугі;
- Додавання працівника;
- Редагування даних працівника;
- Видалення працівника.

Створивши ці функції ми зробимо роботу працівників більш зручною та легкою. Усі дані будуть зберігатись у БД до якої доступ будуть мати лише працівники, це забезпечить надійність та зберігання конфіденційних даних.

3.1.2 Вимоги до організаційних вхідних та вихідних даних

Дані зберігаються в файлі БД з типом .sql.

Вихідні дані: виведення даних здійснюється через пристрої виведення даних таких як монітор або принтер.

Вхідні дані: введення даних здійснюється за допомогою пристроїв введення даних таких як клавіатура та миша. Введення даних здійснюється або вручну або обираючи із представлених у списку.

Вхідні дані, що задіяні у роботі програмного забезпечення представляються у

вигляді наступних таблиць у БД: клієнт, оператор, спеціаліст, послуги. Атрибути таблиць та тип даних атрибутів наведено у таблицях А.1 – А.4.

Таблиця А.1 – Опис атрибутів таблиці «Заявки»

Атрибут	Тип даних
Ідентифікатор заявки	Число
Ідентифікатор працівника	Число
ПІБ	Текст
EMAIL	Текст
Номер телефону	Число
Необхідна послуга	Текст
Коментар клієнта	Текст
Дата надходження заявки	Дата

Таблиця А.2 – Опис атрибутів таблиці «Клієнти»

Атрибут	Тип даних
Ідентифікатор клієнта	Число
ПІБ	Текст
EMAIL	Текст
Номер телефону	Число
Номер паспорта	Текст

Таблиця А.3 – Опис атрибутів таблиці «Працівники»

Атрибут	Тип даних
Ідентифікатор працівника	Число
ПІБ	Текст
Номер телефону	Число
Номер паспорта	Текст
Посада	Текст
Дата народження	Дата
Дата прийому на роботу	Дата

Таблиця А.4 – Опис атрибутів таблиці «Послуги»

Атрибут	Тип даних
Ідентифікатор послуги	Число
Ідентифікатор клієнта	Число
Ідентифікатор працівника	Число
Послуга за якою звертається клієнт	Текст
Дата початку роботи.	Дата
Деталі щодо замовлення	Текст

3.2 Вимоги до надійності

3.2.1 Вимоги до забезпечення надійного функціонування програми

Програмне забезпечення повинно функціонувати без збоїв при умові виконання сукупності організаційно-технічних заходів.

3.2.2 Час відновлення після відмови

Час відновлення після відмови, викликаній несправністю технічних засобів, фатальним збоєм (крахом) операційної системи, не повинен перевищувати часу, необхідного на усунення несправностей технічних засобів і переустановленням програмних засобів.

3.2.3 Відмови з причини не коректних дій працівника

Відмови програми можливі з причин некоректних дій користувача при взаємодії з ПЗ. Для того, щоб уникнути відмов такого роду необхідно надати користувачу лише ті права які необхідні йому для роботи, а також надати користувачу інструкцію по експлуатації програми.

3.3 Вимоги до умов експлуатації

Користувач потрібен мати мінімальні навички в користуванні комп'ютером. Комп'ютер призначений для роботи в закритому опалювальному приміщенні.

3.4 Вимоги до складу та параметрів технічних засобів

Програмне забезпечення «Legalist» буде виконуватися на особистих комп'ютерах або смартфонах наступних характеристик. Системи вимог персонального комп'ютера для користувача представлено у таблиці А.5.

Таблиця А.5 — Системні вимоги для персонального комп'ютера

Комплектуючі	Мінімальні вимоги
Відео карта	1 Gb
Процесор	2.4 GHz
Операційна пам'ять	2 GB RAM
Монітор	HDMI 1920x1080
Місце у пам'яті на HDD / SSD	Відсутнє
Пристрій для взаємодії з користувачем	Клавіатура та комп'ютерна миша
Інтернет	Від 3 МБ/с

Система вимог для мобільних пристроїв для користувача представлено у таблиці А.6.

Таблиця А.6 — Системні вимоги для мобільних пристроїв

Комплектуючі	Мінімальні вимоги
Процесор	1.4 GHz
Операційна пам'ять	1 GB RAM
Місце у пам'яті на HDD / SSD	500 МБ
Пристрій для взаємодії з користувачем	Сенсорна
Інтернет	Від 3 МБ/с

Швидкість та якість роботи програмного забезпечення на конкретному пристрої залежить також від характеристик окремих його комплектуючих.

3.5 Вимоги до інформаційної та програмної сумісності

Для повноцінного функціонування програмного забезпечення необхідно мати комп'ютер або телефон, доступ до інтернету та веббраузер.

3.6 Вимоги до маркування та упаковки

Програма буде розповсюджуватись через веббраузер, тому потрібен лише інтернет, та особистий пристрій: комп'ютер чи смартфон.

3.6.1 Вимоги до маркування

Вимоги до маркування не висуваються.

3.6.2 Вимоги до упаковки

Вимоги до упаковки не висуваються.

3.6.3 Умови пакування

Вимоги до умов пакування не висуваються.

3.6.4 Порядок пакування

Вимоги до порядку пакування не висуваються.

3.7 Вимоги до транспортування та зберігання

Вимоги до транспортування та зберігання не висуваються.

3.7.1 Умови транспортування та зберігання

Вимоги щодо умов транспортування та зберігання не висуваються.

4 ВИМОГИ ДО ПРОГРАМНОЇ ДОКУМЕНТАЦІЇ

Програмна документація повинна містити:

- технічне завдання (ТЗ);
- текст програми
- опис програми;
- інструкцію користувача;
- програму і методику випробувань.

5 СТАДІЇ ТА ЕТАПИ РОЗРОБКИ

Стадії та етапи розробки представлені в таблиці А.7.

Таблиця А.7 – Стадії та етапи розробки програмного забезпечення

Стадії розробки	Етапи робіт	Вміст робіт	Контрольні точки
1	2	3	4
Технічне завдання	Обумовлення необхідності розробки	Постановка задачі. Збір початкових матеріалів. Вибір і обумовлення критеріїв ефективності та якості розроблюваного ПЗ	02.03.2022 -15.03.2022
	Розробка та затвердження ТЗ.	Визначення вимог до ПЗ. Узгодження і затвердження ТЗ.	16.03.2022 -31.03.2022
Ескізний проєкт	Розробка ескізного проєкту	Попередня розробка структури вхідних та вихідних даних. Уточнення методів рішення задачі. Розробка загального опису алгоритму розв'язання задачі	01.04.2022 -15.04.2022
	Затвердження ескізного проєкту	Розробка пояснювальної записки. Узгодження і затвердження ескізного проєкту	16.04.2022 -27.04.2022
Технічний проєкт	Розробка технічного проєкту	Уточнення структури вхідних та вихідних даних. Розробка алгоритму рішення задачі. Розробка структури ПЗ	28.04.2022 -03.05.2022
	Затвердження технічного проєкту	Розробка пояснювальної записки. Узгодження і затвердження технічного проєкту	04.05.2022 - 06.05.2022
Робочий проєкт	Розробка ПЗ	Програмування та налагодження програми	07.05.2022 -10.05.2022
	Розробка програмної документації	Розробка програмної документації відповідно до вимог ГОСТ 19.101-77	11.05.2022 -13.05.2022
	Випробування ПЗ	Розробка, узгодження та затвердження порядку і методики випробувань. Проведення попередніх випробувань. Коригування ПЗ і програмної документації за результатами випробувань.	14.05.2022 -15.05.2022

6 ПОРЯДОК КОНТРОЛЮ ТА ПРИЙМАННЯ

Контроль за кожним етапом розробки проводиться викладачем шляхом надсилання йому завершених розділів кваліфікаційної роботи. Кожна стадія розробки повинна бути представлена в зазначені строки та узгоджена із замовником

(викладачем).

Аналіз та проєктування кожної окремої частини програмного забезпечення контролюється на кожному етапі з урахуванням вимог, зазначених у технічному завданні. Кожен етап розробки має бути представлений в обумовлені терміни та погодженні із замовником.

Приймання здійснюється відповідно до процедур та методів випробувань і документується за допомогою протоколів випробувань. У разі виявлення помилки під час процесу приймання ПЗ оформляється копія порушення, підписується представниками замовника та розробника та затверджується керівником організації – замовника та організації – розробника. Розробник повинен виправити ці помилки та повідомити замовника для повторної перевірки протягом 2 тижнів.

ДОДАТОК Б ТЕКСТ ПРОГРАМИ

Main.js

```
$(function(){

    $('.slider__inner, .news__slider-inner').slick({
        nextArrow: '<button type="button" class="slick-btn slick-next"></button>',
        prevArrow: '<button type="button" class="slick-btn slick-prev"></button>',
        infinite: false
    });

    $('select').styler();

    $('.header__btn-menu').on('click', function(){
        $('.menu ul').slideToggle();
    });

})
```

Baza.sql

```
-- phpMyAdmin SQL
-- version 4.8.1
-- https://www.phpmyadmin.net/
-- Host: 127.0.0.1
-- Generation Time: Sep 06, 2018 at 08:58 AM
-- PHP Version: 7.2.6

SET SQL_MODE = "NO_AUTO_VALUE_ON_ZERO";
SET AUTOCOMMIT = 0;
START TRANSACTION;
SET time_zone = "+00:00";
```

```

/*!40101 SET @OLD_CHARACTER_SET_CLIENT=@@CHARACTER_SET_CLIENT */;
/*!40101 SET @OLD_CHARACTER_SET_RESULTS=@@CHARACTER_SET_RESULTS
*/;

/*!40101 SET @OLD_COLLATION_CONNECTION=@@COLLATION_CONNECTION */;
/*!40101 SET NAMES utf8mb4 */;

--
-- Database: `bank`
--

-- -----

--
-- Table structure for table `Requests`
--

CREATE TABLE `Requests` (
  `clients_id` int(11) NOT NULL,
  `email` varchar(15) NOT NULL,
  `name` int(11) NOT NULL,
  `question` varchar(15) NOT NULL,
  `coment` varchar(50) NOT NULL,
  `date` date(10) NOT NULL,
) ENGINE=InnoDB DEFAULT CHARSET=latin1;

--
-- Dumping data for table `Requests`
--

INSERT INTO `clients` (`clients_id`, `email`, `name`, `question`, `coment`, `date)
VALUES
(1, 'SBI23432310001', 'Savings', 'SBI234323', 'chandan', 'M', '2018-09-06', 'xyz xyz',
'234432234', 20500);

-- -----

--

```

-- Table structure for table `services`

--

```
CREATE TABLE `services` (
  `service_id` int(11) NOT NULL,
  `clients_id` int(11) NOT NULL,
  `type` varchar(255) NOT NULL
  `detail` varchar(255) NOT NULL,
  `date` date(50) NOT NULL

) ENGINE=InnoDB DEFAULT CHARSET=latin1;
```

--

```
INSERT INTO `requests` (`service_id`,`clients_id` type `detail` `date` date);
```

--

-- Table structure for table `clients`

--

```
CREATE TABLE `clients` (
  `clients_id` int(11) NOT NULL,
  `name` varchar(50) NOT NULL,
  `phone` int(11) NOT NULL
  `birth` date(50) NOT NULL,
  `passport` varchar(255) NOT NULL
  `post` varchar(255) NOT NULL,
  `hiring` date(50) NOT NULL

) ENGINE=InnoDB DEFAULT CHARSET=latin1;
```

--

-- Dumping data for table `employers`

--


```

INSERT INTO `employers` (`clients_id`, `name`, `phone` `birth` `passport` `post` `hiring`)
VALUES
(1, 'arun', 'SBI234323');

```

```

-----

```

```

--
-- Table structure for table `clients`
--

```

```

CREATE TABLE `employers` (
  `employers_id` int(11) NOT NULL,
  `name` varchar(255) NOT NULL,
  `phone` int(11) NOT NULL,
  `passport` varchar(255) NOT NULL,
) ENGINE=InnoDB DEFAULT CHARSET=latin1;

```

```

--
-- Dumping data for table `employers`
--

```

```

ALTER TABLE `employers`
  ADD PRIMARY KEY (`id`);

```

```

--
-- Indexes for table `employers`
--

```

```

ALTER TABLE `employers`
  ADD PRIMARY KEY (`id`);

```

```

--
-- Indexes for table `employers`
--

```

```

ALTER TABLE `employers`
  ADD PRIMARY KEY (`id`);

```

```

--

```

```
-- Indexes for table `services`  
  
--  
ALTER TABLE `services`  
  ADD KEY `par_ind` (`TransactionId`);  
  
--  
  
-- Indexes for table `requests`  
  
--  
ALTER TABLE `requests`  
  ADD PRIMARY KEY (`Id`);  
  
--  
  
-- AUTO_INCREMENT for dumped tables  
  
--  
  
--  
  
-- AUTO_INCREMENT for table `clients`  
  
--  
ALTER TABLE `clients`  
  MODIFY `Id` int(11) NOT NULL AUTO_INCREMENT, AUTO_INCREMENT=2;  
  
--  
  
-- AUTO_INCREMENT for table `requests`  
  
--  
ALTER TABLE `requests`  
  MODIFY `Id` int(11) NOT NULL AUTO_INCREMENT, AUTO_INCREMENT=2;  
  
--  
  
-- AUTO_INCREMENT for table `employers`  
  
--  
ALTER TABLE `employers`  
  MODIFY `Id` int(11) NOT NULL AUTO_INCREMENT, AUTO_INCREMENT=2;  
  
--  
  
-- AUTO_INCREMENT for table `clients`  
  
--  
ALTER TABLE `clients`
```

```
MODIFY `Id` int(11) NOT NULL AUTO_INCREMENT, AUTO_INCREMENT=3;
```

```
--
```

```
-- Constraints for dumped tables
```

```
--
```

```
--
```

```
-- Constraints for table `services`
```

```
--
```

```
ALTER TABLE `services`
```

```
    ADD CONSTRAINT `fk_tranTable` FOREIGN KEY (`servicesId`) REFERENCES `services`
(`Id`) ON DELETE CASCADE ON UPDATE CASCADE;
```

```
COMMIT;
```

```
/*!40101 SET CHARACTER_SET_CLIENT=@OLD_CHARACTER_SET_CLIENT */;
```

```
/*!40101 SET CHARACTER_SET_RESULTS=@OLD_CHARACTER_SET_RESULTS */;
```

```
/*!40101 SET COLLATION_CONNECTION=@OLD_COLLATION_CONNECTION */;
```

Userscontroller.php

```
<?php
```

```
    $module="users";
    if(!isset($user_code))
    {
        header("Location: login.php");
    }
}
```

```
/*
```

```
 * Calling the topbar from template
```

```
*/
```

```
if(!isset($_POST['usersSubmitted']))
```

```
{
```

```
    /*
```

```
     * You header file and topbar path here
```

```
    */
```

```
        /*
```

```
         * Sample
```

```
        require_once("../templates/ams/views/header.php");
```

```
        require_once("../templates/ams/views/topbar.php");
```

```
        *
```

```
        */
```

```
    }
```

```
?>
```

```
<body>
```

```
<!-- Start of container and row fluid -->
```

```
<div class="container-fluid-full">
```

```

<div class="row-fluid">
<?php
/*
 * Place the sidebar menu here
 */
    require_once("../templates/ams/views/menu.php");
?>
<!-- start: Content -->
<div id="content" class="span10" >
    <ul class="breadcrumb">
        <li>
            <i class="icon-home"></i>
            <a href="index.php">Home</a>
            <i class="icon-angle-right"></i>
        </li>
        <li><a href="userscontroller.php">users</a></li>
    </ul>

<?php
/*
 * 3 Types of actions in the user
 * 1.View
 * 2.Edit
 * 3.Add
 */

if(isset($_REQUEST['action']))
{
    $action = $_REQUEST['action'];
}
else
{
    $action = 'View';
}

if(strcmp($action, 'View')==0)
{
    require_once("../views/usersview.php");
}
else if(strcmp($action, 'Edit')==0)
{
    $id = $_REQUEST['id'];

    require_once("../views/usersedit.php");
}
else if(strcmp($action, 'Add')==0)
{
    require_once("../views/usersadd.php");
}

?>
</div>
</div>
</div>
<!-- End of container and row fluid-->

```

```

<?php

/*
 * Footer file path here

require_once("../templates/ams/views/footer.php");

*
*/

?>
clients.php

<body onload="hide_buttons();"
<!-- Start of container and row fluid -->
<div class="container-fluid-full">
    <div class="row-fluid">
        <?php
        /*
         * Place the sidebar menu here
         */
        ?>
        <!-- start: Content -->
        <div id="content" class="span10" >
            <ul class="breadcrumb">
                <li>
                    <i class="icon-home"></i>
                    <a href="index.php">Home</a>
                    <i class="icon-angle-right"></i>
                </li>
                <li><a href="customerscontroller.php">Customers</a></li>
            </ul>

        <?php
        /*
         * 3 Types of actions in the user
         * 1.View
         * 2.Edit
         * 3.Add
         */

        if(isset($_REQUEST['action']))
        {
            $action = $_REQUEST['action'];
        }
        else
        {
            $action = 'View';
        }

        if(strcmp($action, 'View')==0)
        {
            require_once("../views/customersview.php");
        }
    
```

```

else if(strcmp($action,'Edit')==0)
{
    $id = $_REQUEST['id'];

    require_once("../views/customersedit.php");
}
else if(strcmp($action,'Add')==0)
{
    require_once("../views/customersadd.php");
}

?>
</div>
</div>
</div>
<!-- End of container and row fluid-->
<?php

/*
* Place Footer Content Here
*/
require_once("../templates/ams/views/footer.php");
?>
User.php
<?php

namespace model;

class user extends Model{

    /*
    * This is a class for taking care of the user.
    * This class defines as set of functions using
    * RRPhpLibrary class and all the other functions.
    */

    private $tbname;

    public $userID;
    public $userLname;
    public $userFname;
    public $userEmail;

    private $RRPhpLib;

    public function __construct() {
        $this->RRPhpLib = new RRPhpLibrary;
    }

    public function __destruct() {

    }

    public function getRecord($id)
    {
        $where = " where userID=".$id;

```

```

        $fieldList = $this->RRPhpLib->pCrud->getFieldList($tbname);
        $result = $this->RRPhpLib->pCrud->SimpleSelectColumnsWhere($this->tbname,$fieldList,$where);
        print_r($result);
    //return $result;
    }
    public function getRecords()
    {
        return $result = $RRPhpLib->$pCrud->SimpleSelectColumns($this->tbname,$fieldlist);
        //Format the response into an array

    }
    public function updateRecord($values,$where)
    {
        $where = " where userID=".$id;
        $fieldList = $this->RRPhpLib->pCrud->getFieldList($this->tbname);
        return $result = $this->RRPhpLib->pCrud->SimpleSelectColumnsWhere($this->tbname,$fieldList,$where);
    }

    public function __get($name) {
        echo "Get:".$name;
    }
    public function __set($name, $value) {
        echo "Set:".$name." value:".$value;
    }

}

?>

```

Userview.php

```

<?php

use model\user as user;
//echo "User testing here";
$userObj = new user;
$result = $userObj->getRecords();

?>
<div class="row-fluid sortable">
<div class="box span12">
<div class="box-header" data-original-title>
<h2><i class="halflings-icon user"></i><span
class="break"></span>Customers</h2><a style="margin: -5px 0 0px
10px;" class="btn btn-small btn-success"
href="../controller/customerscontroller.php?action=Add"></i>Add
Customer</a>
<div class="box-icon">
<a href="#" class="btn-setting"><i class="halflings-icon
wrench"></i></a>

```

```

        <a href="#" class="btn-minimize"><i class="halflings-icon
chevron-up"></i></a>
        <a href="#" class="btn-close"><i class="halflings-icon
remove"></i></a>
            </div>
        </div>
        <div class="box-content">
            <table class="table table-striped table-bordered bootstrap-
datatable datatable">
                <thead>
                    <tr>
                        <th>ID</th>
                        <th>First Name</th>
                        <th>Last Name</th>
                        <th>Email</th>
                        <th>Actions</th>
                    </tr>
                </thead>
                <tbody>
                    <?php

                    $rowcount = mysql_num_rows($result);
                    if($rowcount > 0)
                    {
                        while ($row =
mysql_fetch_assoc($result))
                        {
                            echo '<tr>';
                            for($j=0; $j <
count($fieldlist); $j++)
                            {
                                if(strcmp($fieldlist[$j], 'user_is_active')==0)
                                {
                                    if($row[$fieldlist[$j]] == 1)
                                    {
                                        echo
'<td class="center">
<span class="label label-success">Active</span>
</td>';
                                    }
                                    else
                                    {
                                        echo
'<td class="center">
<span class="label">Inactive</span>
</td>';
                                    }
                                }
                            }
                        }
                    }
                }
            }
        }
    }

```



```

else if(strcmp($fieldlist[$j], 'customer_fye')==0)
{
    $fye_date = explode("/", $row[$fieldlist[$j]]);
    echo "<td>".$fye_date[1]."-".$fye_date[0]."-".$fye_date[2]."</td>";
}

else
{
    echo
    "<td>".$row[$fieldlist[$j]]."</td>";
}

$curr_url =
$next_url =
$base_url =
echo '<td
class="center">
class="btn btn-success"
href="'.$next_url.'&id='.$row['customer_id'].'&action=Edit">
class="halflings-icon white edit"></i>
</a>';

if($row['customer_is_active']==1)
{
    $next_url =
    $curr_url =
    echo '
    <a class="btn
    btn-info"
    href="'.$next_url.'&previous_url='.$curr_url.'&tname=users&id='.$row
    ['user_id'].'&fieldname=user_id&active=user_is_active">
    class="halflings-icon white remove"></i>
    </a>';
}
else
{
    $next_url =
    $curr_url =
    echo '
    <a class="btn
    btn-info"

```

```

href="'. $next_url. '?previous_url=' . $curr_url. '&tbnname=users&id=' . $row
['user_id']. '&fieldname=user_id&active=user_is_active'">
<i
class="halflings-icon white ok"></i>
</a>';
}

$next_url =
$base_url. '/model/delete.php';
$curr_url =
$base_url. '/controller/userscontroller.php';

    if(strcmp($user_type, "Super Admin") == 0 ||
strcmp($user_type, "Staff Admin") == 0)
    {
        echo'
        <a class="btn btn-
danger"
href="'. $next_url. '?previous_url=' . $curr_url. '&tbnname=users&id=' . $row
['user_id']. '&fieldname=user_is_active'">
<i
class="halflings-icon white trash"></i>
</a>';
    }
    echo'</td>

</tr>';
}

?>
</tbody>
</table>
</div>
</div><!--/span-->
</div><!--/row-->

```

ДОДАТОК В ОПИС ПРОГРАМИ

1 Загальні відомості

1.1 Найменування програми

Найменування програмного забезпечення вебсайту для консультацій у банківській сфері «Legalist».

1.2 Програмне забезпечення, необхідне для функціонування програми

Для забезпечення роботи вебсайту необхідно мати інтернет від 1 МБ/с, та сучасний веббраузер.

1.3 Мови програмування

Під час розробки програмного забезпечення використано мову PHP.

2 Функціональне призначення

2.1 Класи розв'язуваних завдань та призначення програми

Функціональним призначенням програмного забезпечення для консультацій у банківській сфері «Legalists» є автоматизація процесів:

- авторизація користувачів;
- призначення оператора заявці;
- призначення спеціаліста послугі;
- зберігання та використання інформації про клієнтів;
- зберігання та використання інформації про працівників.

2.2 Функціональні обмеження

Щоб робота програмного забезпечення було комфортною, логічною, та зрозумілою, було надано такі обмеження:

користувач із роллю «Оператор» обмежений у правах;

- не може редагувати дані про клієнтів;
- не може додавати редагувати та видаляти дані про працівників;

- не може призначати себе до послуги;
- не може переглядати список послуг.

користувач із роллю «Спеціаліст» обмежений у правах;

- не може додавати клієнтів;
- не може переглядати, видаляти та призначати себе до заявки;
- не може створювати послуги;
- не може додавати, редагувати та видаляти дані про працівників.

користувач із роллю «Адміністратор» обмежений у правах;

- не може переглядати, видаляти та призначати себе до заявки;
- не може призначати себе до послуги;
- не може редагувати дані про клієнтів;
- не може додавати клієнтів;
- не може переглядати список клієнтів;
- не може створювати послуги;
- не може переглядати список послуг.

3 Опис логічної структури

3.1 Структура програми

Програмне забезпечення вебсайту «Legalist» має дві складові частини: клієнтська частина - це графічний інтерфейс та база даних - програмне забезпечення, що забезпечує зберігання даних та їх видачу в момент запиту.

3.2 Алгоритм програми

Робота програмного забезпечення вебсайту «Legalist» представляє собою взаємодію між клієнтською частиною та базою даних. Алгоритм роботи програми є наступним:

- 1) Користувач взаємодіє з програмним додатком.
- 2) Програмний додаток звертається до бази даних (якщо це потрібно) та відсилає результат.
- 3) Програмний додаток відображає отриманий результат.

3.3 Використовувані методи

Програмний додаток - «Legalist» розроблено мовою програмування PHP з використанням HTML, CSS та JavaScript та використовуючи СКБД MySQL.

4 Технічні засоби, що використовуються

Програмне забезпечення «Legalist» буде виконуватися на особистих комп'ютерах та смартфонах наступних характеристик.

Смартфон:

- Оперативна пам'ять: 4 Гігабайт;
- Веббраузер Google Chrome 96;
- Смартфон на базі андроїд.

Персональний комп'ютер:

- Процесор: Intel Core 2 Duo;
- Оперативна пам'ять: 4 ГБ;
- Операційна система: Windows 10.

Швидкість та якість роботи програмного забезпечення на конкретному пристрої залежить також від характеристик окремих його комплектуючих.

5 Виклик та завантаження

Відкрити браузер, перейти на вебсайт через посилання. Відкриється головна сторінка сайту.

6 Вхідні дані

Вхідні дані: введення даних здійснюється за допомогою пристроїв введення даних (клавіатура та миша). Дані або вводяться вручну, або обираються із представлених списків.

7 Вихідні дані

Вихідні дані: виведення даних здійснюється за допомогою пристроїв виведення даних (монітор та принтер).

ДОДАТОК Г ІНСТРУКЦІЯ КОРИСТУВАЧА

Дана інструкція призначена для ознайомлення користувачів з функціями ПЗ та облегшення роботи із ним.

Для початку роботи користувач переходить за посиланням на сайт. Після того як користувач перейшов до вебсайту перед ним буде відображено головну сторінку вебсайту приклад на рисунку Г.1.

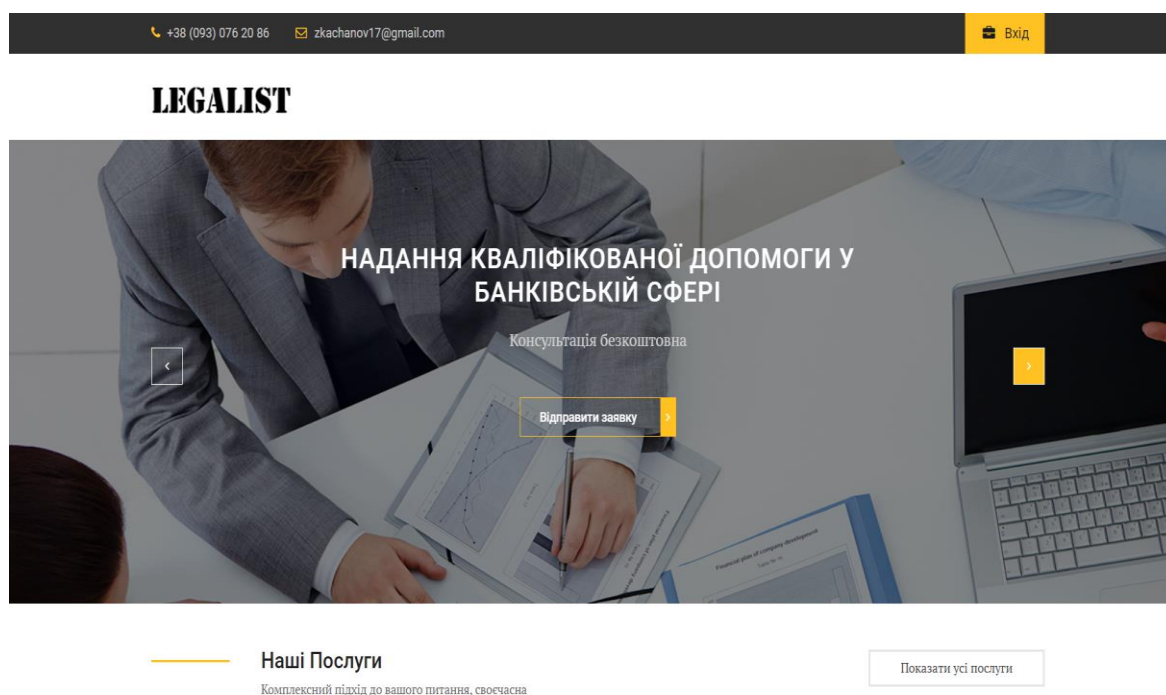


Рисунок Г.1 - Головна сторінка вебсайту

Для роботи клієнту необхідно заповнити інформацію у поля для відправлення заявки після чого натиснути “Відправити” далі заявка буде відправлена. На рисунку Г.2 ви можете побачити форму заявки.

Отримати Консультацію
Кваліфікована допомога для кожного клієнта.

E-mail

Номер телефону

ПІБ

Необхідна послуга

Коментар клієнта

Відправити >

Рисунок Г.2 - Форма відправлення заявки для клієнта

Після того як клієнт відправив заявку йому потрібно лише очікувати дзвінка від оператора.

Для роботи працівникам необхідно натиснути посилання вхід та вести дані для авторизації після чого натиснути “Увійти”. На рисунку Г.3 ви можете побачити приклад меню авторизації.

Логін

Пароль

Увійти

Рисунок Г.3 - Меню авторизації

Після авторизації перед працівниками буде відкрито головне меню. Перед кожною спеціальністю буде відкрито своє меню із різним функціоналом.

1. Оператор

Після того як оператор авторизується буде відкрито головне меню. Приклад із головним меню оператора буде на рисунку Г.4.

LEGALIST

Головне меню

Список заявок

Список клієнтів

Додати клієнта

Вийти

Рисунок Г.4 – Головне меню оператора

Після авторизації в оператора будуть доступні наступні функції:

1) Вийти із головного меню

Якщо оператор закінчив працювати він натискає кнопку вийти для завершення роботи після чого його повертає на головну сторінку.

2) Перегляд списку заявок.

Для того, щоб оператору переглянути список заявок йому необхідно натиснути на “Список заявок” після чого буде відкритий список із необхідними даними про заявки. Приклад зі списком заявок наведено на рисунку Г.5.

LEGALIST

Меню

Вийти

Список Заявок

ID Заявки	Номер телефону	ПІБ	EMAIL	Питання	Коментар	Дата надходження	ID Оператора
1	3809510954534	Жмишенко Валерій Альбертович	zmichenko54@gmail.com	Інвестиції	Збір інвестиційного портфелю	15.03.2022	1
2	3805003478025	Шевченко Анатолій Вікторович	Viktor4543@ukr.net	Консультавання	Потрібна консультація	15.03.2022	1
3	3809374905122	Мельник Ігор Валентинович	melnikkk@gmail.com	Інші послуги	nnn	15.03.2022	
4	3805564234234	nnnn nnn nnn	sraaaaam@gmail.com	Інші послуги	nnnnnnnnnnnnnnnnnn	15.03.2022	

Прийняти заявку

Видалити заявку

Прийняти заявку

Видалити заявку

Рисунок Г.5 – Список заявок

3) Призначення оператора заявці

Після того як оператор перейде до списку заявок у нього буде можливість призначити вільну заявку собі. Для того, щоб прийняти заявку оператору необхідно натиснути “Прийняти заявку” після чого заявка буде прикріплена до оператора.

4) Видалення заявки

Якщо клієнт залишив невірні дані для зв'язку або заявка схожа на спам оператор може видалити заявку натиснувши “Видалити заявку” після чого заявка буде видалена із БД.

5) Повернення в меню

Натиснувши кнопку “Меню” оператор повернеться до головного меню.

6) Додавання клієнта

Знаходячись у меню оператор може додати клієнта у БД для цього оператору необхідно натиснути кнопку “Додати клієнта” після чого з'явиться форма для додавання клієнта. Оператор заповнює усі необхідні дані та натискає “Додати клієнта” після чого запис додається у БД. Якщо оператор вирішив не додавати клієнта він може натиснути меню. Приклад із формою додавання клієнта на рисунку Г.6.

LEGALIST

Додавання клієнта

ПІБ

Email

Номер паспорта

Номер телефону

Додати клієнта

Меню Вийти

Рисунок Г.6 – Форма додавання клієнта

7) Перегляд списку клієнтів

Якщо оператору необхідно переглянути список клієнтів він натискає на “Список клієнтів” після чого буде відкритий список із необхідними даними клієнтів. Приклад зі списком клієнтів наведено на рисунку Г.7.

Список клієнтів

ID Клієнта	ПІБ	EMAIL	Номер телефону	Номер паспорту	
1	Жмищенко Валерій Альбертович	zmichenko54@gmail.com	3809510954534	05067905	Створити послугу
2	Шевченко Анатолій Вікторович	Viktor4543@ukr.net	3805003478025	04576532	Створити послугу
3	Мельник Ігор Валентинович	melnikkkk@gmail.com	3809374905122	05672312	Створити послугу

Рисунок Г.7 – Список клієнтів у оператора

8) Створення послуги

Для того, щоб створити послугу оператору потрібно обрати необхідного клієнта і натиснути “Створити послугу” після чого з’явиться форма для додавання послуги, оператор заповнює усі необхідні поля інформацією та натискає “Створити послугу”. Приклад із формою створення послуги наведено на рисунку Г.8.

Створення послуги

Послуга

Дата початку роботи

Деталі щодо замовлення

Створити послугу

Рисунок Г.8 – Форма для створення послуги

2. Спеціаліст

Після того як спеціаліст авторизується буде відкрито головне меню. Приклад із головним меню оператора буде на рисунку Г.9.

Головне меню

Список клієнтів

Список послуг

Рисунок Г.9 – Головне меню спеціаліста

Після авторизації у спеціаліста будуть наступні функції:

1) Вийти із головного меню

Якщо спеціаліст закінчив працювати він натискає кнопку вийти для завершення роботи після чого його повертає на головну сторінку.

2) Перегляд списку клієнтів

Якщо спеціалісту необхідно переглянути список клієнтів необхідно натиснути на “Список клієнтів” після чого буде відкритий список із необхідними даними клієнтів. Приклад зі списком клієнтів наведено на рисунку Г.10.

Меню

Вийти

Список клієнтів

ID Клієнта	ПІБ	EMAIL	Номер телефону	Номер паспорту	
1	Жмишенко Валерій Альбертович	zmichenko54@gmail.com	3809510954534	05067905	Редагувати дані
2	Шевченко Анатолій Вікторович	Viktor4543@ukr.net	3805003478025	04576532	Редагувати дані
3	Мельник Ігор Валентинович	melnikkkk@gmail.com	3809374905122	05672312	Редагувати дані

Рисунок Г.10 – Список клієнтів у спеціаліста

3) Повернення в меню

Натиснувши кнопку меню спеціаліст повернеться до головного меню.

4) Редагування даних клієнта

Якщо спеціалісту потрібно змінити дані клієнтів він натискає “Редагувати дані” та змінює необхідні дані далі він підтверджує свій вибір натискає “Прийняти редагування”. Приклад редагування даних клієнта наведено на рисунку Г.11

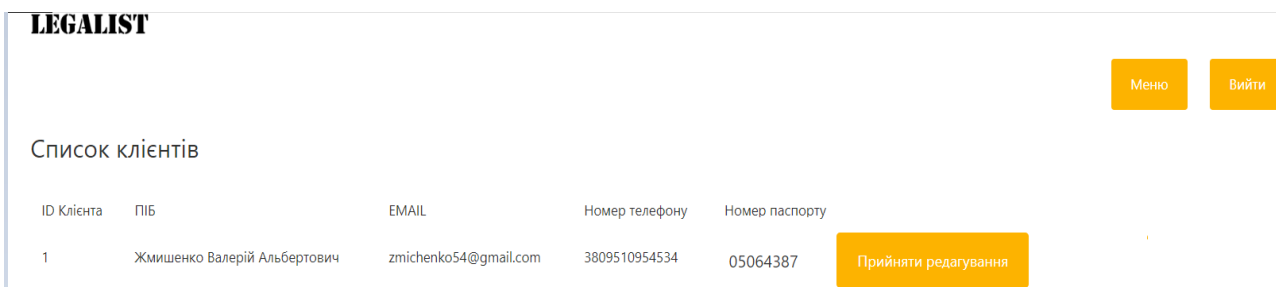


Рисунок Г.11 – Редагування даних клієнта

5) Перегляд списку послуг

Якщо спеціалісту необхідно переглянути список послуг необхідно натиснути на “Список послуг” після чого буде відкритий список із необхідними даними щодо послуг. Приклад зі списком послуг наведено на рисунку Г.12.

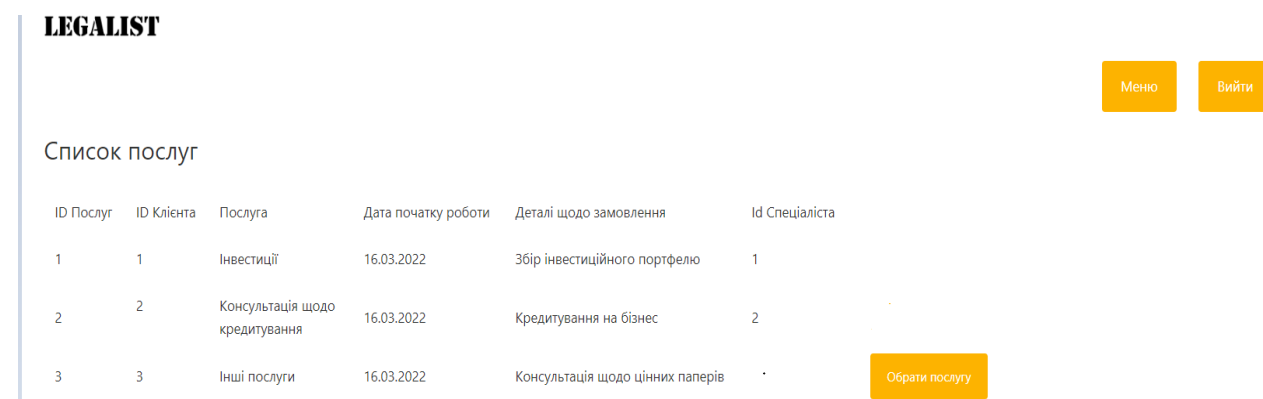


Рисунок Г.12 – Список послуг

6) Призначення спеціаліста послугі.

Після того як спеціаліст перейде до списку послуг у нього буде можливість призначити вільну послугу собі. Для того, щоб прийняти послугу спеціалісту необхідно натиснути “Обрати послугу” після чого послуга буде прикріплена до оператора.

3. Адміністратор

Після того як адміністратор авторизується буде відкрито головне меню. Приклад із головним меню адміністратора буде на рисунку Г.13.

Вийти

Головне меню

Список працівників

Додати працівника

Рисунок Г.13 – Головне меню адміністратора

Після авторизації в адміністратора будуть наступні функції

1) Вийти із головного меню

Якщо адміністратор закінчив працювати він натискає кнопку вийти для завершення роботи після чого його повертає на головну сторінку.

2) Додавання працівників

Знаходячись у меню адміністратор може додати працівника у БД для цього адміністратору необхідно натиснути кнопку “Додати працівника” після чого з’явиться форма для додавання працівника. Адміністратор заповнює усі необхідні дані та натискає “Додати працівника” після чого запис додається у БД. Якщо адміністратор вирішив не додавати клієнта він може натиснути меню. Приклад із формою додавання працівника на рисунку Г.14.

Меню

Вийти

Додавання працівника

ПІБ

Номер телефону

Номер паспорту

Посада

Дата початку роботи

Дата народження

Додати працівника

Рисунок Г.14 – Форма додавання працівника

3) Перегляд списку працівників

Для того, щоб переглянути список працівників адміністратору необхідно натиснути на “Список працівників” після чого буде відкритий список із необхідними даними працівників. Приклад зі списком працівників наведено на рисунку Г.15.

LEGALIST

Меню

Вийти

Список працівників

ID Працівник а	ПІБ	Номер телефону	Номер паспорту	Посада	Дата початку роботи	Дата народження		
1	Кравчук Георгій Ігоревич	38095034786 9	05043423	Адміністра тор	14.03.2022	25.04.1995	Редагувати дані	Видалити працівника
2	Комар Дмитро Іванович	38095443716 2	06043212	Спеціаліст	14.03.2022	27.07.1980	Редагувати дані	Видалити працівника
3	Плужник Василь Геннадійович	38050449143 4	05439212	Оператор	14.03.2022	23.06.2001	Редагувати дані	Видалити працівника

Рисунок Г.15 – Список працівників

4) Редагувати дані працівника

Якщо адміністратору потрібно змінити дані про працівників він натискає “Редагувати дані” та змінює необхідні дані далі він підтверджує свій вибір натиснувши “Прийняти редагування”. Приклад редагування даних працівника наведено на рисунку Г.16.

LEGALIST

Меню

Вийти

Список працівників

ID Працівник а	ПІБ	Номер телефону	Номер паспорту	Посада	Дата початку роботи	Дата народження	
1	Кравчук Георгій Ігоревич	38095034786 9	05043423	Адміністра тор	14.03.2022	14.03.1995	Прийняти редагування

Рисунок Г.16 – Редагування даних клієнта

5) Видалення працівника

Якщо працівник звільняється із місця роботи адміністратор може видалити працівника із БД натиснувши “Видалити працівника” після чого працівник буде видалений із БД.

ДОДАТОК Д ПРОГРАМА ТА МЕТОДИКА ВИПРОБУВАНЬ ПРОГРАМНОГО ЗАБЕЗПЕЧЕННЯ

1 Об'єкт випробувань

Об'єктом випробування є програмне забезпечення вебсайту для консультацій у банківській сфері «Legalist».

2 Мета випробувань

Мета проведення випробування є перевірка працездатності програмного забезпечення та перевірка виконання його головних функцій, а також перевірка відповідності характеристик та вимог розробленої програми.

3 Вимоги до вебсайту

Під час проведення випробувань усі функціональні характеристики перевіряються на відповідність вимогам які описані у технічному завданні.

4 Вимоги до програмної документації

4.1 Програмна документація включає в себе наступні документи

- 1) технічне завдання;
- 2) текст програми;
- 3) програму та методику випробувань;
- 3) інструкцію користувача;
- 4) опис програмного забезпечення.

5 Засоби та порядок випробувань

5.1 Технічні засоби, що використовуються під час випробувань

Склад технічних засобів, що використовувались під час випробувань.

Смартфон:

Samsung Galaxy a30;

Процесор: Samsung Exynos 7904, 8 ядер, 2 GHz;

Оперативна пам'ять: 4 гігабайти;

Накопичувач даних: 64 гігабайти;

Веббраузер Google Chrome 96;

Смартфон на базі андроїд.

Персональний комп'ютер:

Відео карта: GeForce GT 710;

Процесор: Intel Core 2 Duo;

Оперативна пам'ять: 4 ГБ;

Операційна система: Windows 10.

5.2 Порядок проведення випробувань

Порядок проведення випробувань складається з перевірки вимог до функціональних характеристик програмного продукту та перевірки вимог до програмної документації.

6 Методи випробувань

6.1 Методика проведення перевірки вимог до програмної документації

Перевірка дотримання вимог програмної документації на програмний продукт виконується візуально представником служби, відповідальної за експлуатацію. У ході перевірки зіставляється склад і комплектність програмної документації, представленої розробником, з переліком програмної документації, наведеним у пункті «Склад програмної документації, пропонованої на випробування» цього документа.

Перевірка вважається завершеною у випадку відповідності складу та комплектності програмної документації, представленої розробником, переліком програмної документації, наведеному у зазначеному вище пункті.

6.2 Методика проведення перевірки вимог до функціональних характеристик програмного продукту

Перевірка працездатності програми виконується згідно з пунктом «Вимоги до функціональних характеристик» Додатку А Технічне завдання. Перевірка вважається завершеною у разі відповідності складу і послідовності виконаних дій пункту «Вимоги до функціональних характеристик». В процесі тестування була перевірена функціональність за темою «Розробка програмного забезпечення вебсайту для банківської системи "Legalist"». У таблиці Д.1, що наведена нижче, вказано перелік випробувань основних функціональних можливостей.

Таблиця Д.1- Методика випробування

№	Дія	Результат який повинен бути	Результат перевірки	Зауваження
1	Перевірка правильності Введення даних при авторизації	Програма повинна перевірити введені дані з записами у БД		
2	Перевірка правильності додавання нового працівника	Програма повинна перевірити коректність введених даних		
3	Перевірка правильності додавання нового клієнта	Програма повинна перевірити коректність введених даних		
4	Перевірка правильності додавання нової послуги	Програма повинна перевірити коректність введених даних		
5	Перевірка правильності введених даних при редагуванні даних про клієнтів або працівників в БД	Програма повинна перевірити коректність введених даних		
6	Перевірка правильності видалення записів із БД	Програма повинна перевірити введені дані з записами у БД		