

МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ
Тернопільський національний економічний університет
Факультет комп'ютерних інформаційних технологій
Кафедра комп'ютерних наук

ЗАЄЦЬ Владислав Олександрович

**Веб-орієнтована система підтримки діяльності
Оржівського житлово-комунального
господарства/ Web-based system for activity
support of Orzhiv housing and communal Services**

напрямок підготовки: 6.050103 - Програмна інженерія
фахове спрямування - Програмне забезпечення систем

Бакалаврська дипломна робота

Виконав студент групи ПЗС-42
В. О. Заєць

Науковий керівник:
к.т.н., старший викладач
ВОЙТЮК І.Ф.

Бакалаврську дипломну роботу
допущено до захисту:

"__" _____ 20__ р.

Завідувач кафедри
_____ **А. В. Пукас**

РЕЗЮМЕ

Дипломна робота містить 106 сторінок, 16 таблиць, 52 рисунка, список використаних джерел із 17 найменувань.

Метою дипломної роботи є розробка Веб-орієнтованої системи для підтримки діяльності Оржівського житлово-комунального господарства.

Об'єктом дослідження є процес оплати комунального платежу онлайн.

Предметом дослідження є програмна система для оплати комунальних послуг користувачами.

Методи розробки базуються на технології ASP.NET.

Одержані результати полягають в розробці web-орієнтованої системи для підтримки діяльності Оржівського житлово комунального господарства.

Ключові слова: оплата, онлайн, комунальні платежі, веб-сайт, чек.

RESUME

Thesis contains 106 page, 16 tables, 52 figures, list of references with 17 titles.

The aim of the thesis is the development of Web-based system for activity support of Orzhiv housing and communal Service.

The object of research is the process of payment.

The subject of the study is the web-site for payment for communal payments.

The methods of making technology based on ASP.NET.

The results are in the development of Web-based system for activity support of Orzhiv housing and communal Service.

Keywords: payment, online, web-system, communal payments, check.

Зміст

ВСТУП	8
РОЗДІЛ 1 АНАЛІЗ ПРЕДМЕТНОЇ ОБЛАСТІ діяльності Оржівського житлово-комунального господарства.....	10
1.1. Коротка характеристика об'єкту управління.....	10
1.2. Опис предметної області	10
1.3. Огляд і аналіз існуючих аналогів, що реалізують функції предметної області	15
1.4. Специфікація вимог до системи.....	22
Розділ 2 ПРОЕКТУВАННЯ СИСТЕМИ УПРАВЛІННЯ БІЗНЕС-ПРОЦЕСАМИ ФУНКЦІОНУВАННЯ сервісу по оплаті комунальних платежів	32
2.1. Розроблення архітектури програмної системи.....	32
2.2. Проектування структури бази даних	36
Розділ 3 ПРОГРАМНА РЕАЛІЗАЦІЯ Інтернет сервісу для оплати комунальних послуг	40
3.1. Програмна реалізація системи	40
3.2. Програмна реалізація бази даних	50
Розділ 4 ТЕСТУВАННЯ ТА ДОСЛІДНА ЕКСПЛУАТАЦІЯ	58
4.1. Тестування.....	58
4.2. Розгортання програмного продукту	61
4.3. Інструкція користувача	63
ВИСНОВКИ.....	70
СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ	71
ДОДАТОК А.....	73
ДОДАТОК Б.....	74
ДОДАТОК В	97

ВСТУП

У дипломній роботі описано систему для оплати комунальних послуг із використанням глобальної мережі інтернет.

Актуальність створеного ресурсу пояснюється тим, що розроблюваний продукт повинен:

- Задовільнити користувачів усіх вікових категорій, легким доступом до даних пов'язаних з комунальними платежами та їх оплату;
- Гарантувати клієнтам безпеку при проведенні платіжних операцій;
- Забезпечувати безпеку при оплаті комунальних платежів;

На сьогодні існуючі ресурси не дозволяють безпечно оплачувати комунальні послуги тому актуально на даний момент, створити систему яка буде забезпечувати безпеку даних користувачів, та здійснювати безкоштовну оплату.

Ефективність розроблюваної системи забезпечується:

- легким доступом до даних з комунальними платежами та їх оплату;
- розширенням аудиторії, яка може оплачувати комунальні платежі у інтернеті;
- розширенням функцій для тих моментів, коли потрібно забезпечити функціонал одночасно декільком ЖКГ.

Система забезпечить звичайному користувачу велику економію часу, та повноцінну інформацію на рахунок оплати комунальних послуг.

Також за допомогою інтернет-ресурсу можливо отримати повну інформацію про Оржівське ЖКГ, його послуги. Інтернет – ресурс має складну функціональність, може бути призначений для внутрішнього використання або відкритий для відвідувача. Також користувачам надається можливість перегляду транзакцій через інтернет.

Продукт розроблений згідно вимог, для цього було використано мову C# і технологію ASP.Net в програмному середовищі Microsoft Visual Studio 2013. Важливою перевагою даного порталу є наглядність та простота доступу до

потрібних користувачеві ресурсів, також можливість зв'язку з власником даного ресурсу.

Дана веб-сторінка сумісна з всіма сучасними веб-браузерами, такими як GoogleChrome, Mozilla Firefox, Opera, Internet Explorer.

Практичне значення: дана система може використовуватись для людей усіх категорій населення. На сервісі організовано повний склад допомоги користувачу здійснення оплати за комунальні послуги.

РОЗДІЛ 1 АНАЛІЗ ПРЕДМЕТНОЇ ОБЛАСТІ ДІЯЛЬНОСТІ ОРЖІВСЬКОГО ЖИТЛОВО-КОМУНАЛЬНОГО ГОСПОДАРСТВА

1.1. Коротка характеристика об'єкту управління

Об'єктом управління є Оржівське ЖКГ. Ця організація виконує такі функції: оплата комунальних послуг, представлення тарифів користувачам послуг, збереження даних про оплачені платежі. Таким чином формується історія платежів, що залежить від того, які на даний момент ціни на послуги, чи ще їх називають тарифами. Дані про платежі зберігаються в базі даних. Їх зберігають для подальшого перегляду і для формування певних звітів по виконаним транзакціям для користувачів послугами ЖКГ. Доступ до цієї бази даних має тільки підприємство. Мета дипломної роботи полягає в тому, щоб реалізувати доступ до інформації про платежі не тільки для працівників підприємства, а й клієнтів ЖКГ. Так як користувачі зберігають інформацію про їх сплати на папері, це не дає їм можливості переглянути свої борги, оновлену інформацію стосовно пільгових категорій, субсидій. Це означає, що буде створена веб-орієнтована система для підтримки діяльності Оржівського ЖКГ.

Основною метою системи є забезпечення способу довготривалого і централізованого зберігання в електронному вигляді різного роду даних і платіжної інформації, а також забезпечення легкого доступу до платіжних систем, щоб оплатити комунальні послуги.

На даний момент у системі використовується застаріле програмне забезпечення 1с 4 версії та паперова документація, що на даний момент не є актуальним для роботи на підприємстві.

1.2. Опис предметної області

Проаналізувавши роботу організації Оржівського ЖКГ, та дослідивши функції та процеси, які відбуваються всередині організації, потік інформації та документообіг, інформаційна система, що підлягає автоматизації, це відділ

бухгалтерії який займається обліком усіх платежів в системі, що виконуються. Також вона відповідає за розрахунок зарплат, витрачених матеріалів на виконання ремонту.

Система обліку платежів в житлово-комунальному господарстві розроблена для полегшення виконання платежів клієнтами. Це автоматизована система ведення платіжної документації та створення звітності по платежам, відповідно до вимог міністерства юстиції України.

Для належної організації роботи системи необхідне виконання таких основних бізнес-процесів (рисунок 3.1):

- Ведення інформації.
- Оплата послуг.
- Перегляд історії транзакцій.
- Перегляд тарифів на послуги .

Розглянемо детальніше кожен з вище представлених бізнес-процесів. На рисунку 1.1 бачимо діаграму функцій процесу авторизації ведення інформації.



Рис. 1.1. Діаграма функцій процесу додавання нових показників

Для того, щоб ввести покази з лічильників користувач вводить свою інформацію, а адміністратор її перевіряє, тільки після цього дані передаються до БД.

Характеристику бізнес-процесу авторизації користувача в системі наведено в таблиці 1.1.

Таблиця 1.1

Характеристика бізнес-процесу авторизації користувача в системі

Назва характеристики	Значення характеристики
Назва бізнес-процесу	Введення інформації
Основні учасники	Клієнт, адміністратор
Вхідна подія	Перехід на додавання даних
Вхідні документи	Дані з лічильників
Вихідна подія	Створений кортеж у БД
Вихідні документи	Сформована таблиця з даними, які ввів клієнт.
Клієнт бізнес-процесу	Користувач

На рисунку 1.2 побудована діаграма функцій процесу оплати послуг.

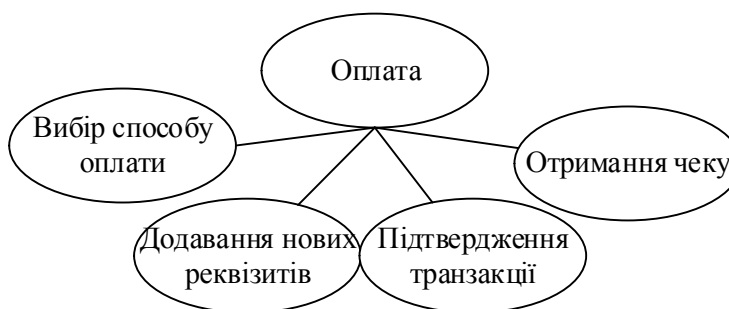


Рис. 1.2. Діаграма функцій процесу оплати послуг

Функція «оплата послуг» дає можливість користувачу оплатити комунальні послуги, шляхом вибору способу оплати, перевести гроші с картки, чи перейти до безпечного платежу з атестатом безпеки через банк. Додавання нових платіжних даних таких як новій покази лічильників з комунальними послугами, підтвердження транзакції та отримання чеку про виконану операцію.

Характеристика бізнес-процесу оплати комунальних послуг наведена в таблиці 1.2.

Таблиця 1.2

Характеристика бізнес-процесу оплати комунальних послуг

Назва характеристики	Значення характеристики
Назва бізнес-процесу	Оплата комунальних послуг
Вхідна подія	Перехід на форму оплати
Основні учасники	Користувач
Вхідні документи	Покази лічильників
Вихідні документи	Чек
Вихідна подія	Послугу сплачено
Клієнт бізнес-процесу	Клієнт

На рисунку 1.3 зображено діаграму функцій процесу ознайомлення із тарифами на послуги.



Рис. 1.3. Діаграма функцій ознайомлення із тарифами на послуги

Функція «ознайомлення із тарифами на послуги» дає змогу користувачеві переглянути вибраний тариф, та провести сортування для зручного користування системою. Тарифи існують для того, щоб можна було обрахувати вартість комунальних послуг за допомогою тарифу та об'єму використаної послуги, система буде автоматично вираховувати ціну за послуги.

Характеристику бізнес-процесу ознайомлення з тарифами на послуги наведено в таблиці 1.3.

Таблиця 1.3

Характеристика бізнес-процесу ознайомлення з тарифами на послуги

Назва характеристики	Значення характеристики
Назва бізнес-процесу	Перегляд тарифів на послуги
Вхідна подія	Вибір потрібного тарифу
Основні учасники	Користувач
Вхідна подія	Вибраний користувачем тариф
Вихідна подія	Сформована форма з дійсними тарифами
Вихідні документи	Сформована структурна таблиця з обраними даними
Клієнт бізнес-процесу	Користувач

На рисунку 1.4 зображено діаграму функцій процесу перегляду історії транзакцій.



Рисунок 1.4. Діаграма функцій процесу перегляду історії транзакцій

Функція «перегляду історії транзакцій» дає змогу користувачеві знайти необхідну транзакцію, вибравши відповідну категорію та вказавши його назву послуги.

Характеристику бізнес-процесу пошуку потрібної транзакції в таблиці 1.4.

Таблиця 1.4

Характеристика бізнес-процесу пошуку потрібної транзакції

Назва характеристики	Значення характеристики
Назва бізнес-процесу	Пошук потрібної транзакції
Основні учасники	Користувач
Вхідна подія	Вибір категорії та введення ключового слова
Вхідні документи	Значення текстового поля фільтрації
Вихідна подія	Знайдено необхідну інформацію
Вихідні документи	Сформована таблиця по транзакціям
Клієнт бізнес-процесу	Користувач

1.3. Огляд і аналіз існуючих аналогів, що реалізують функції предметної області

Основною метою проведення аналізу існуючих аналогів є виявлення недоліків, порівняння можливостей та інтерфейсу.

До аналогів інтернет-ресурсів оплати комунальних послуг можна віднести: mega-billing.com, ipay.ua, megabank.net. Головна сторінка системи mega-billing.com показана на рисунку 1.5.

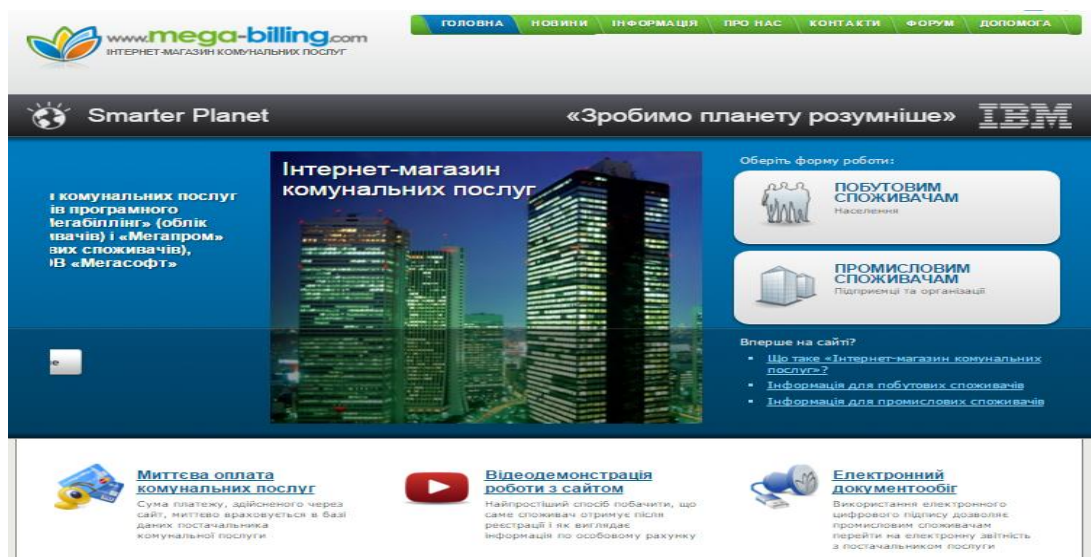


Рис. 1.5. Головна сторінка системи Mega-billing

На рисунку 1.5, видно що інтерфейс системи Mega-billing є інтуїтивно зрозумілим. Навіть якщо ви щойно почали користуватись даною системою, то труднощів при пошуку певної функції не буде. У правій частині рисунка 1.5 можна побачити поле для пошуку та категорії, за якими буде проводитись пошук. Завдання системи — створення між постачальниками комунальних послуг та споживачами чесних та прозорих відносин. Другим завданням системи є виключення можливих свідомих перекручувань і маніпуляцій даними, як з боку споживача, так і з боку постачальника. На рисунку 1.6 відтворено процес реєстрації на сайті.

The screenshot displays the registration interface for the Mega-billing system. The title is "Ідентифікація споживача послуг". The process is divided into three steps: 1. Ідентифікація споживача (highlighted in blue), 2. Створення облікового запису, and 3. Активація облікового запису.

The main section is titled "Оберіть постачальника:" and includes the following elements:

- Region:** A dropdown menu with "Закарпатська область" selected.
- Settlement:** A dropdown menu with "(Все)" selected. A note states: "Якщо в списку немає Вашого населеного пункту, спробуйте обрати найближчий більш великий населений пункт."
- Providers:** Two buttons with logos: "ПАТ «Закарпаттяобленерго»" (with a lightning bolt logo) and "КП «Водоканал м.Ужгород»" (with a water wave logo).
- Sub-division:** A dropdown menu with a red warning message: "Оберіть підрозділ постачальника, що обслуговує Ваш населений пункт".
- Personal account number:** An input field with a note: "Без додаткових символів ('№', 'о/р' і т.д.) Набір символів має точно відповідати кодуванню Вашого особового рахунку постачальника послуг".
- Last payment amount:** An input field with a note: "Формат суми: 0.00 Надходження грошових коштів на особовий рахунок відбувається із запізненням, у зв'язку з затримкою перерахування грошей фінансовими установами (Банки, пошта)".
- Payment status:** A checkbox labeled "Платежів ще не було".

Рис. 1.6. Реєстрація на сайті Mega-billing

Під час реєстрації, користувач обирає постачальника і вводить дані свого рахунку, після чого історія платежів даного споживача буде збережено. Якщо під час реєстрації утворюються проблеми, то сайт надає можливість вирішення питань стосовно реєстрації.

Крім того, на головній сторінці системи розташований блок новин, де користувач може переглянути перелік нових надходжень. Новини зберігаються в архіві, який розраховано на п'ять років зберігання інформації. На рисунку 1.7 зображено новини сайту Mega-billing.



Рисунок 1.7 – Новини сайту Mega-billing

Результати огляду та аналізу можливостей системи Mega-billing показали: система має зручний та зрозумілий вигляд та є простою у користуванні; контент є «добре» по групуваням, що значно спрощує процес пошуку матеріалів. Крім того, система характеризується високою швидкістю. Однак, було виявлено і певні недоліки: зазначений програмний продукт не надає можливості переглядати усі регіони та міста, щоб здійснити реєстрацію.

Веб-сайт Ірау надає можливість оплачувати різноманітні види послуг, не тільки комунальних (Рисунок 1.8). Сервіс іРау.ua - це сучасна Інтернет-платформа, за допомогою якої можна здійснювати онлайн - платежі або грошові перекази. За роки роботи сервіс перетворився на лідера ринку онлайн-платежів. Місія іРау.ua: впроваджувати нові методи захисту карткових даних, створюючи максимальний захист клієнтів під час інтернет-платежів; розробляти власні методи захисту від інтернет-шахрайства та пропонувати їх ринку з метою убезпечити клієнтів не тільки іРау.ua, але й інших платіжних

сервісів і банків. На рисунку 1.9 проілюстровано загальний вигляд головної сторінки сайту.

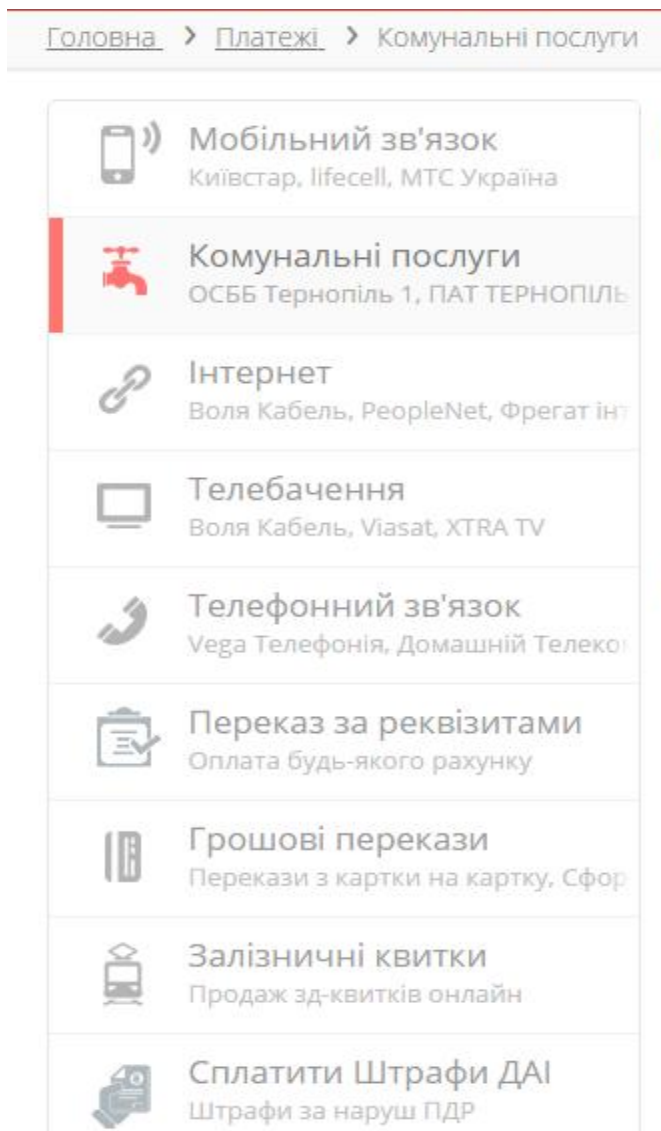


Рисунок 1.8 – Платежі системи Iray

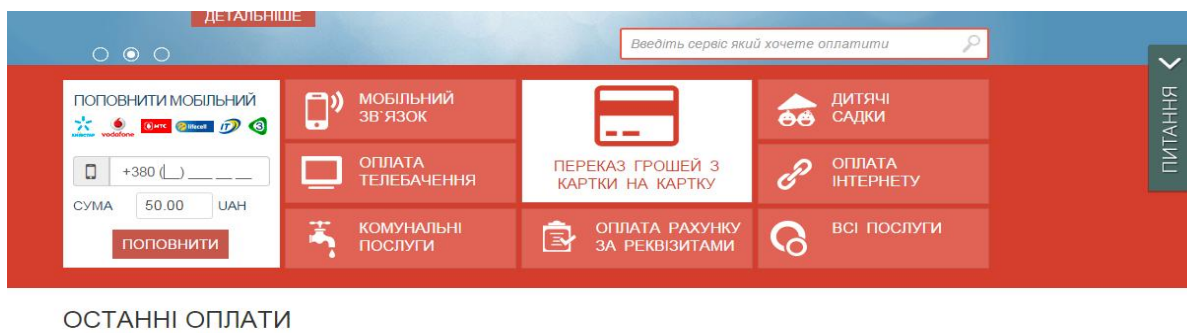


Рисунок 1.9 – Головна сторінка сайту Iray

Веб-сайт дозволяє економити час адже - час оплати рахунку займає не більше 10 хвилин. Є також переваги і в грошах - в багатьох містах України діє знижка до 10% на сплату комунальних послуг, якщо оплата здійснюється до 20 числа (включно). Онлайн оплата комунальних послуг на сайті iPay.ua тепер доступна в наступних областях і містах:

Київська обл., Харківська обл., Луганська обл., Донецька обл., Дніпропетровська обл., Одеська обл., Запорізька обл.

Під час користування сайтом, користувач надає йому доступ до інформації щодо місця проживання, для швидкого доступу до оплати за певним містом.

Веб-сайт Megabank дозволяє сплачувати комунальні послуги. При цьому має не самий зручний інтерфейс, через те що усі посилання на сторінки створеними простими методами, які не потребують значних знань, та не виділяють сайт серед інших. У зв'язку з цим, процес оплати значно ускладнюється. Під час реєстрації акаунта можлива функція сайту яка покращує безпеку користування, так як необхідне обов'язкове підтвердження на пошті, яку ви вказали. Головна сторінка сайту Megabank зображена на рисунку 1.10.

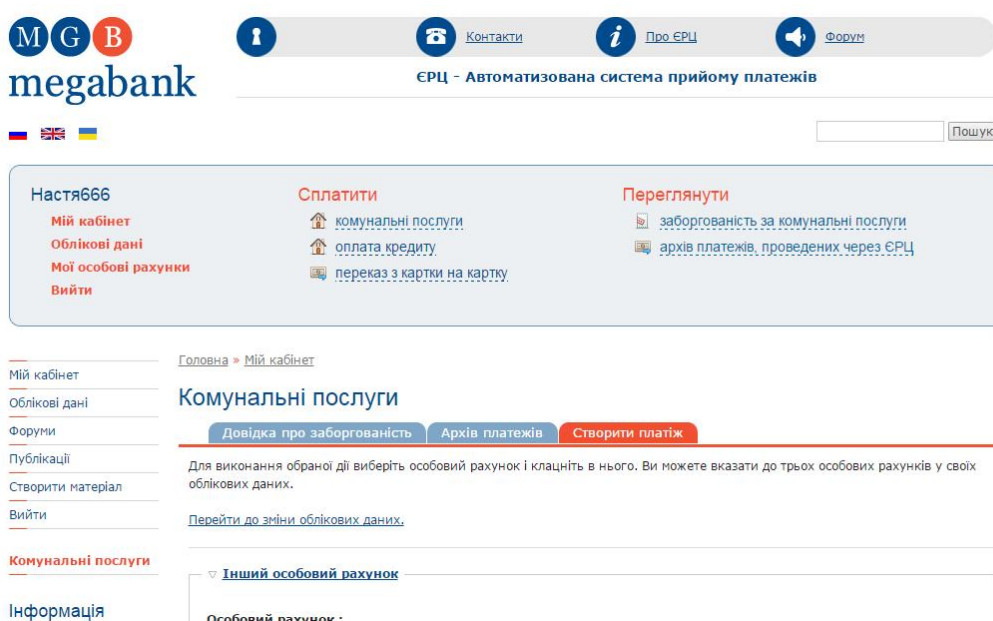


Рисунок 1.10 – Головна сторінка сайту Megabank

Ресурс створювався на єдиній системі оплати з метою забезпечення зручності оплати спожитих житлово – комунальних послуг та енергоносіїв для

населення. Мешканцям замість 5-6 платіжних документів, надається єдина квитанція для оплати всіх видів житлово-комунальних послуг та енергоносіїв. Єдині квитанції приймаються до оплати всіма відділеннями Ощадбанку, комерційних банків і пошти. Крім того, "Мегабанк" на додаток до діючих пунктів прийому платежів, розвиває власну мережу приймальних пунктів.

У результаті обміну інформацією з підприємствами у своїй базі даних "Мегабанк" зберігає відомості про нарахування та заборгованості всіх платників комунальних послуг, які передаються в кожний приймальний пункт ВАТ "Мегабанк". Це означає, що клієнти можуть перейти на посилення та одержати необхідну інформацію про стан своїх платежів, а саме про нарахування, попередні оплати та заборгованості і прийняти рішення стосовно наступної транзакції. Можливість прийому платежів-онлайн була створена пізніше, проте одразу стала популярною серед користувачів.

Провівши аналіз аналогових програмних продуктів можна сказати, що подібні сайти спрощують процес оплати комунальних послуг, є зручними у користуванні та мають інтуїтивно-зрозумілий інтерфейс, але – будь-який користувацький інтерфейс завжди можна покращити, чи додати інноваційні графічні функції, які будуть спрощувати користувачу навігацію, чи користування сервісом.

В таблиці 1.5 наведено порівняльну характеристику вищеописаних продуктів.

Наступні сервіси були повністю проаналізовані та оцінені функції які в них присутні для користування клієнтів послуг ЖКГ, далі наведено повну порівняльну характеристику цих продуктів, для подальшого аналізу та проектування власної системи.

Таблиця 1.5

Порівняльна характеристика продуктів

Фірма розробник	Mega-billing	Ірау	Megabank
Назва програмного продукту	Інтернет-магазин комунальних послуг	Сайт Всеукраїнського сервісу онлайн платежів	Єдиний розрахунковий центр
Функціональність	Пошук потрібних матеріалів Оплата послуг	Пошук потрібних матеріалів Оплата послуг	Пошук потрібних матеріалів Оплата послуг
Інтерфейс користувача	Простий, зручний у користуванні	Простий, зручний у користуванні	Простий, не зручний у користуванні
Допомога користувачу	Загальні питання по сайту	Немає	Немає

Далі наведенні недоліки існуючих аналогів для їх аналізу та покращення створюваної системи для комунальних платежів.

Недоліки системи Mega-billing:

- Платежі обробляються не автоматично, тому час на них затрачений досягає семи днів.;
- Застарілий дизайн;
- Комісія сайту при оплаті 1% від суми платежу;

Недоліки системи Ірау:

- Комісія сайту 2%+5 грн;

- Оплата тільки через картку, немає можливості використовувати такі способи оплати як: Web-money, Yandex-money і через смс-платежі;
- Безпека на низькому рівні;
- Обмежена кількість областей для оплати;

Недоліки системи Megabank:

- Не зручний інтерфейс;
- Платежі виконуються на сторонньому сервісі;
- Обов'язкова реєстрація;

Тому під час розробки веб-орієнтованої системи підтримки діяльності Оржівського Житлово-комунального господарства важливо враховувати та реалізувати усі переваги розглянутих аналогів, а також уникнути недоліків виявлених у існуючих аналогах Ця система буде забезпечувати:

- швидку роботу при оплаті;
- захист інформаційних даних користувачів;
- безпеку при переказі коштів;
- безкоштовна оплата послуг для користувачів системи.

1.4. Специфікація вимог до системи.

Глосарій – це словник основних використовуваних термінів. Глосарій проекту знаходиться у додатку А.

На рисунку 1.11 і 1.12 проілюстровано діаграми варіантів використання системи.

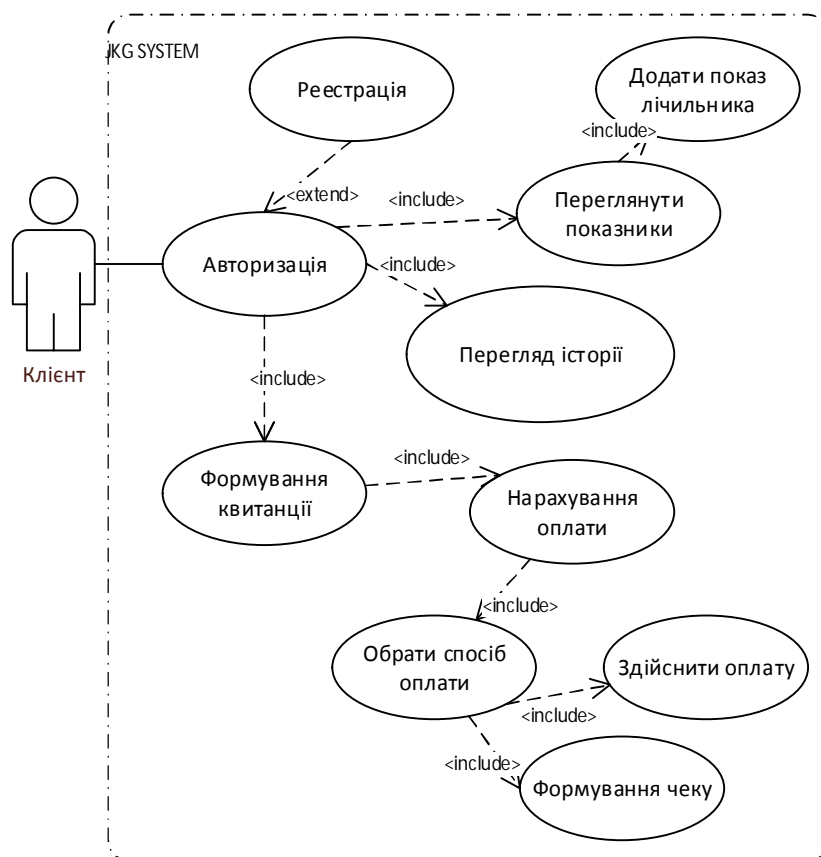


Рис.1.11. Діаграма використання для клієнту системи

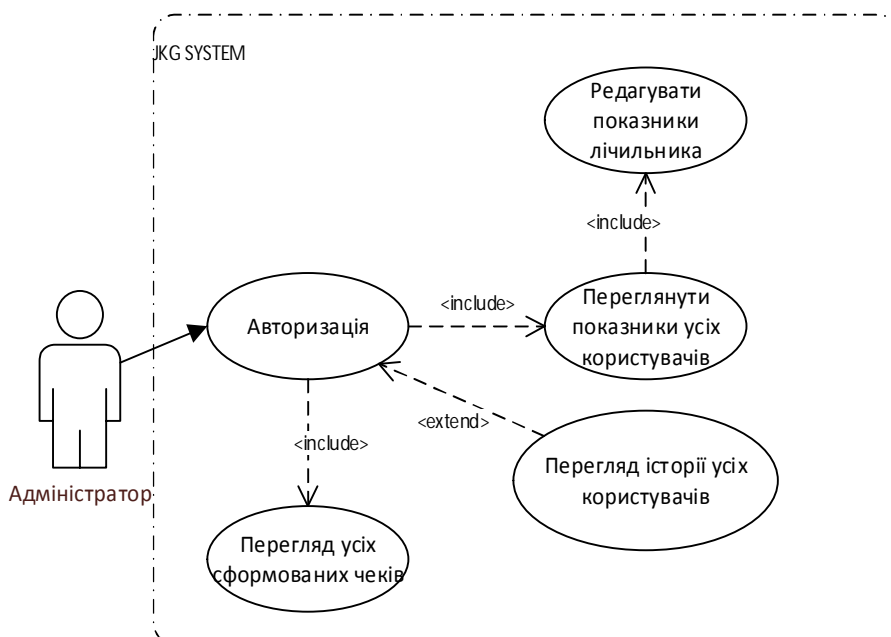


Рис.1.12. Діаграма використання для адміністратора системи

Опис варіантів використання поданий у таблицях 1.6-1.14.

Таблиця 1.6

Варіант використання «Авторизація»

Контекст використання	Авторизація в системі
Дійові особи	Клієнт, адміністратор
Передумова	Реєстрація в системі
Тригер	Відкрита авторизаційна сторінка
Сценарій	1. Перехід на сторінку авторизації; 2. Заповнення полів форми; 3. Підтвердження.
Пост-умова	Для Клієнта: перенаправлено в особистий кабінет Для адміністратора: перенаправлено в адмін-панель.

На рисунку 1.13. показано Розкадровка варіанту використання «Авторизація».

Логін

Пароль:

Рисунок 1.13. Розкадровка варіанту використання «Авторизація»

Таблиця 1.7

Варіант використання «Оплата комунальних послуг»

Контекст використання	Додавання нового матеріалу
Дійові особи	Клієнт
Передумова	Авторизація в системі
Тригер	Відкрита форма оплати
Сценарій	1. Перехід на форму оплати; 2. Вибір послуги, яку необхідно оплатити; 3. Здійснення операції. 4. Перерахувати кошти
Пост-умова	Матеріал додано в історію оплати, та сформовано чек.

На рисунку 1.14. можна переглянути розкадровку варіанту використання «Оплата».

Рисунок 1.14. Розкадровка варіанту використання «Оплата»

Таблиця 1.8

Варіант використання «Перегляд транзакцій»

Контекст використання	Перегляд транзакцій зроблених в системі
Дійові особи	Клієнт, адміністратор.
Передумова	Авторизація в системі.
Тригер	Знайдено матеріал який відповідає критеріям пошуку.
Сценарій	1. Перегляд транзакцій за певний період;
Пост-умова	Користувач переглянув свої транзакції. Адміністратор переглянув усі транзакції.

На рисунку 1.15. можна переглянути розкадровку варіанту використання «Перегляд транзакцій».

Дата оплати	Сума оплати	Метод оплати	Назва послуги	Коефіцієнт оплати
20.04.2016	200	PrivatBank	Електроенергія	0,99

Рис. 1.15. Розкадровка варіанту використання «Перегляд транзакцій»

Таблиця 1.9

Варіант використання «Перегляд показників»

Контекст використання	Перегляд показників
Дійові особи	Клієнт
Передумова	Авторизація в системі
Тригер	Відкрите вікно пошуку
Сценарій	1. Вибір послуги; 2. Введення даних по вибраному типу.
Пост-умова	Знайдено потрібний матеріал, та сформовано таблицю для його виведення

На рисунку 1.16. Зображено Розкадровка варіанту використання «Перегляд показників»



Рис. 1.16. Розкадровка варіанту використання «Перегляд показників»

Таблиця 1.10

Варіант використання «Реєстрація»

Контекст використання	Реєстрація в системі нового користувача
Дійові особи	Клієнт
Передумова	Користувач натиснув кнопку «Реєстрація»
Тригер	Відкрите вікно реєстрації
Сценарій	1) Введено данні клієнта 2) Перевірена правильність
Пост-умова	Клієнт створений у системі у вигляді кортежу

На рисунку 1.17. можна переглянути Розкадровка варіанту використання «Реєстрація».

Ім'я	<input type="text"/>
Логін	<input type="text"/>
Пароль	<input type="text"/>
Підтвердіть пароль	<input type="text"/>
Адреса	<input type="text"/>
Номер квартири	<input type="text"/>
@mail	<input type="text"/>
	<input type="button" value="Зареєструватись"/>

Рис. 1.17. Розкадровка варіанту використання «Реєстрація»

Таблиця 1.11

Варіант використання «Перегляд усіх сформованих чеків»

Контекст використання	Перегляд чеків
Дійові особи	Адміністратор
Передумова	Адміністратор авторизувався
Тригер	Натиснути на кнопку «Перегляд чеків»
Сценарій	1) Введено данні усіх клієнтів 2) Перевірена правильність
Пост-умова	Перевірена правильність введених даних клієнтами

На рисунку 1.18. можна переглянути Розкадровка варіанту використання «Перегляд усіх сформованих чеків».

Дата оплати	Сума оплати	Номер переводу
<input type="text"/>	<input type="text"/>	<input type="text"/>
<input type="text"/>	<input type="text"/>	<input type="text"/>
<input type="text"/>	<input type="text"/>	<input type="text"/>

Рис. 1.18. Розкадровка варіанту використання «Перегляд усіх сформованих чеків»

Таблиця 1.12

Варіант використання «Перегляд історії платежів усіх користувачів»

Контекст використання	Пошук історії платежів усіх користувачів
Дійові особи	Адміністратор
Передумова	Адміністратор авторизувався
Тригер	Натиснути на кнопку «Перегляд платежів»
Сценарій	1) Введено данні усіх клієнтів 2) Перевірена правильність
Пост-умова	Перевірена правильність введених даних клієнтами

На рисунку 1.19. можна переглянути Розкадровка варіанту використання «Перегляд історії платежів усіх користувачів».

Дата оплати	Сума оплати	Метод оплати	Назва послуги	Коіфіцієнт оплати	Id користувача
20.04.2016	200	PrivatBank	Електроенергія	0,99	

Рис. 1.19. Розкадровка варіанту використання «Перегляд історії платежів усіх користувачів»

Таблиця 1.13

Варіант використання «Перегляд усіх показників користувачів»

Контекст використання	Пошук потрібного матеріалу
Дійові особи	Адміністратор
Передумова	Адміністратор авторизувався
Тригер	Натиснути на кнопку «Перегляд показників»
Сценарій	1) Введено данні усіх клієнтів 2) Перевірена правильність
Пост-умова	Перевірена правильність введених даних клієнтами

На рисунку 1.20. можна переглянути Розкадровка варіанту використання «Перегляд усіх показників користувачів»

Назва послуги	Показник	ID користувача
Електроенергія ▾	98.146 ▾	

Рис. 1.20. Розкадровка варіанту використання «Перегляд усіх показників користувачів»

Далі, було сформовано специфікацію функціональних вимог, які наведені у таблиці 1.14.

Таблиця 1.14

Специфікація функціональних вимог

Ідент. вимоги	Назва вимоги (варіанту використання)	Атрибути вимоги	
		Пріоритет	Складність
1	Авторизація	Обов'язкова	Середня
2	Додавання показників	Обов'язкова	Висока
3	Оплата послуг	Обов'язкова	Висока
4	Перегляд загального списку оплати	Рекомендована	Середня
5	Перегляд загального списку чеків	Рекомендована	Середня
6	Реєстрація	Обов'язкова	Висока
7	Перегляд транзакцій	Обов'язкова	Низька
8	Перегляд показників	Рекомендована	Середня

Специфікація не функціональних вимог зображена у таблиці 1.15.

Таблиця 1.15

Специфікація не функціональних вимог

Ідент. вимоги	Назва вимоги	Характеристики
1.	Застосовність	
1.1	Навчання користувача	Не більше 15хв
1.2	Застосовність створеної системи відносно інших	Не повинна суттєво відрізнятися від відомих
2	Надійність	
2.1	Середній час безвідмовної роботи	5 год.
2.2	Середній час роботи без ремонту	6 міс
3	Робочі характеристики	
3.1	Швидкодія	Середній час транзакції оплати 1с (час оновлення браузера і запиту до сервера)
3.2	Продуктивність	Утримання 100 користувачів без помилок.
3.3	Місткість	База даних може утримувати до 1тб інформації
3.4	Режим зниженої продуктивності	Режим зниженої продуктивності відсутній.
3.5	Використання ресурсів	Інтернет (100мб канал), 2гб оперативної пам'яті, 2ядра 2 ghz+
4	Експлуатаційна придатність	
4.1	Стандарти надійності	Використовувати основний функціонал тільки з функцій та процедур для збільшення надійності та швидкості виправлення помилок

Таблиця 1.15 продовження

5	Проектні обмеження	
5.1	Система	Обов'язково має утримуватись на системі Windows Server x64 будь -якої версії
5.2	Програмне забезпечення	На системі обов'язково має бути встановлений .net Framework 4+ версії
6	Вимоги до документації	
6.1	Наявність документації користування	Наявність зрозумілої інструкції для користувача, яка в повному обсязі демонструє усі функції системи

Висновки до першого розділу:

1. Наведено короткий опис об'єкту управління «Оржівське ЖКГ» та виділено основні критерії майбутньої системи.
2. Описано всі функції, які виконуються в предметній області.
3. Проведено аналіз відомих програмних систем, описано недоліки, розроблена порівняльна характеристика основних аналогів в предметній області, та виділені їх недоліки.
4. Наведено специфікацію вимог до системи, сформовано діаграми варіантів використання та повну розкадровку варіантів використання. Створено специфікацію функціональних та не функціональних вимог до розроблюваної системи.

РОЗДІЛ 2 ПРОЕКТУВАННЯ СИСТЕМИ УПРАВЛІННЯ БІЗНЕС-ПРОЦЕСАМИ ФУНКЦІОНУВАННЯ СЕРВІСУ ПО ОПЛАТІ КОМУНАЛЬНИХ ПЛАТЕЖІВ

2.1. Розроблення архітектури програмної системи

Для функціональної частини web-сайту, яка забезпечує обробку запитів було взято клієнт-серверну архітектуру. Дана технологія складається з наступних алгоритмів:

1. До програми, вірніше – до бази даних серверу, посилається запит, який формується користувачем системи;
2. Програмою проводяться маніпуляції з базами даних, що знаходяться на сервері. В цей момент формується відповідь на запит і передається клієнту
3. Результат роботи із базами даних, тобто відповідь на запит клієнта відображається на його дисплеї. На протязі всього часу, коли клієнт користується сервером, цей цикл буде повторюватись.

На рисунку 2.1 процес передачі інформації в архітектурі web-додатку зображено схематично.

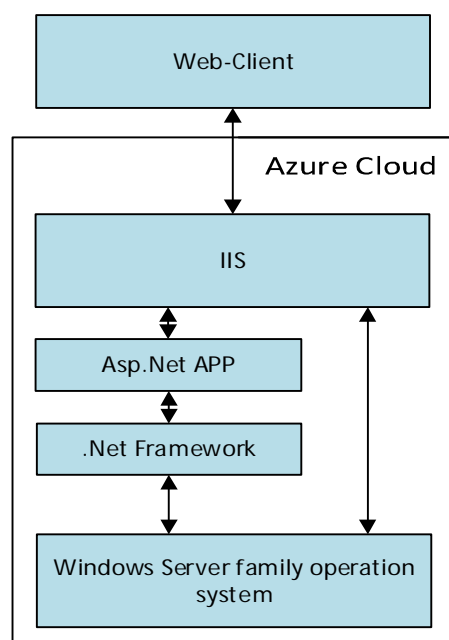


Рис. 2.1. Процес передачі інформації в архітектурі

Для відображення бізнес-логіки розроблюваної системи спроектовано UML - діаграму класів. Діаграма класів зображена на рисунку 4.2 і відображає співвідношення між класами спроектованої системи.

На рисунку 2.2 наведено діаграму основних класів.

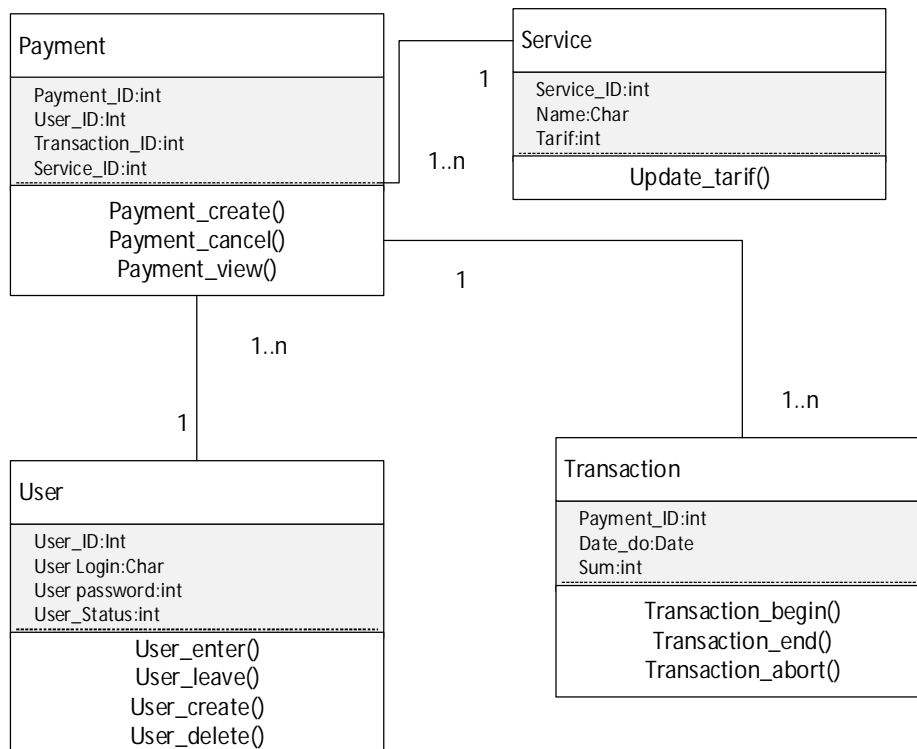


Рис. 2.2. Діаграма класів

Діаграма класів дозволяє відтворювати логічне уявлення, а на його основі створюється вихідний код описаних класів. Відображення ієрархії систем, взаємозв'язків класів та інтерфейсів реалізується за допомогою символів у діаграмі. Система яка проектується складається з наступних класів:

- User – забезпечує авторизацію, реєстрацію та сесію для користувача,;
- Transaction – надає можливість оплачувати одночасно до 3 послуг;
- Payment – забезпечує користувача елементами для оплати комунальних послуг;
- Service – Виділяє для користувача і актуальні тарифи та послуги ;

Усі об'єкти у програмуванні відповідають одній логіці – вони завжди змінюють стан, для того щоб забезпечити роботу, тому основну логіку роботи

можна переглянути на діаграмі станів. Її вигляд для прецедента «Активність користувача» зображено на рисунк 2.3.



Рис.2.3. Діаграма станів для прецедента «Активність користувача»

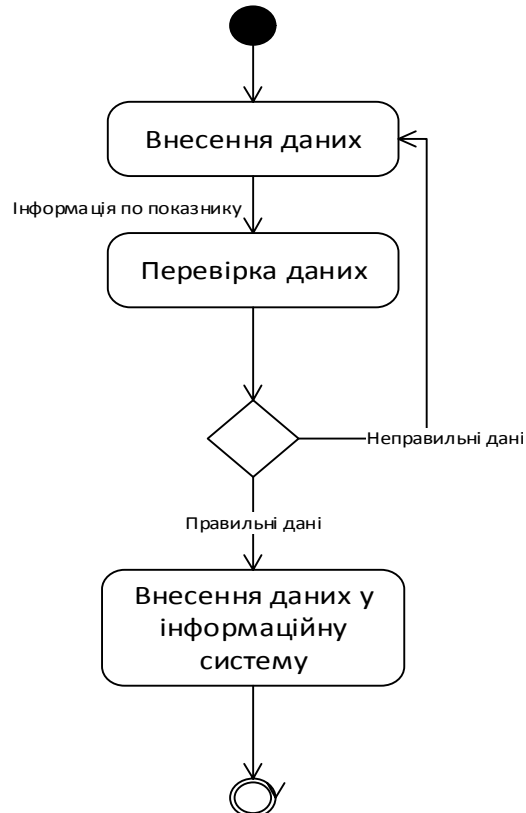


Рис. 2.4. Діаграма активності для прецедента «Модуль введення та перевірки даних»

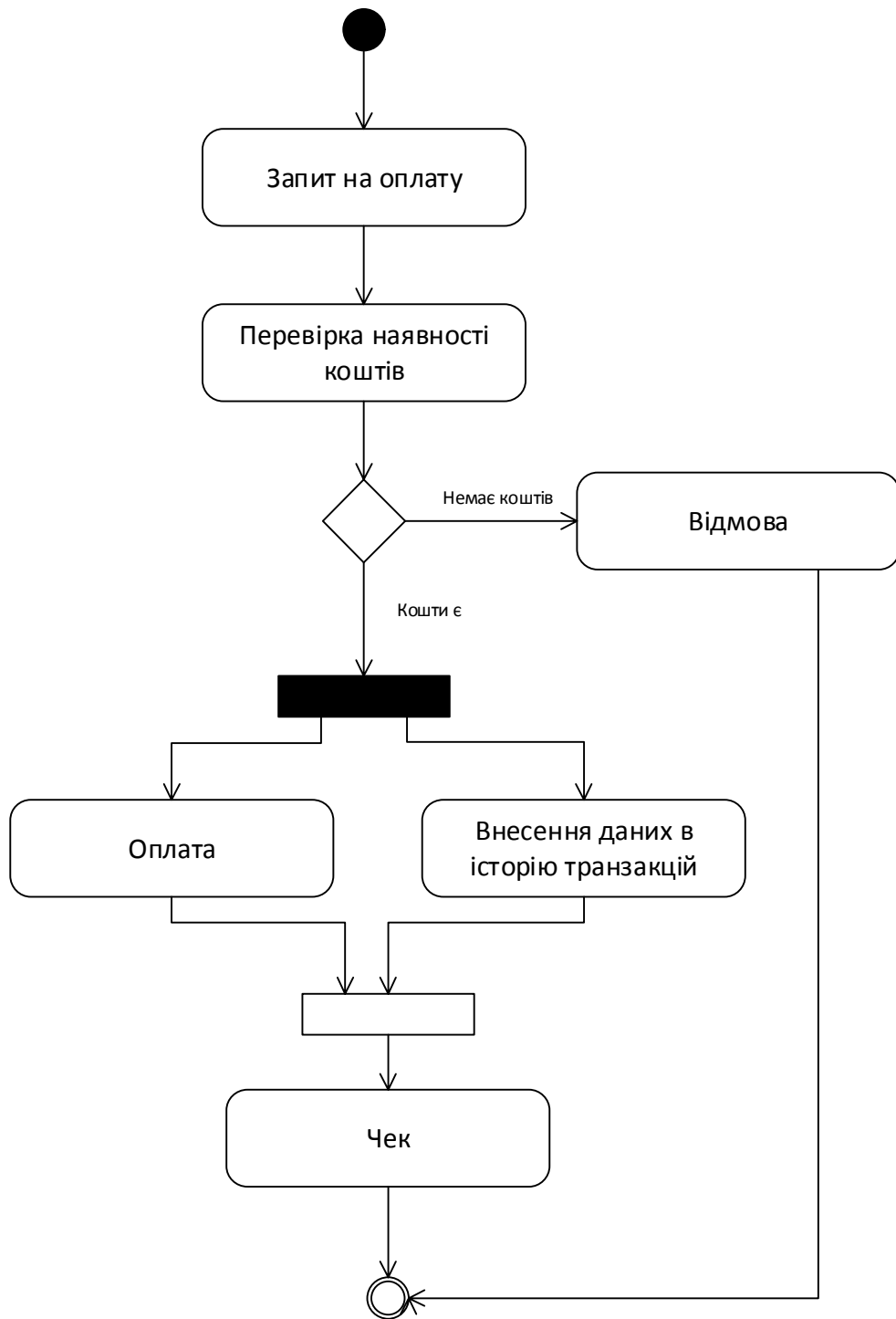


Рис 2.5. Діаграма активності для прецедента «Модуль оплати комунальних платежів»

На рисунку 2.4. та 2.5 зображено діаграму активності для модулів оплати комунальних платежів та перевірки запиту із введеними даними для клієнта. Активність систем під час того як клієнт виконує транзакцію комунальних

послуг через сайт відображається за допомогою модуля оплати. Модуль перевірки даних відображає чи всі дані, які ввів користувач в системі, логічно вірні, наприклад:

- Неможливо буде ввести показники лічильника нижче ніж вони є на даний момент, щоб система не помилилась в обрахунках при оплаті комунальних послуг;
- Неможливо буде ввести інформацію, яка є неповною для системи, пов'язану з картковою оплатою, через те, що там обов'язково має бути фіксована кількість цифр в найменуванні картки;
- Неможливо буде додати до системи двох однакових користувачів.

Проектування глобальної дата логічної моделі даних зображено на рисунку 2.6.

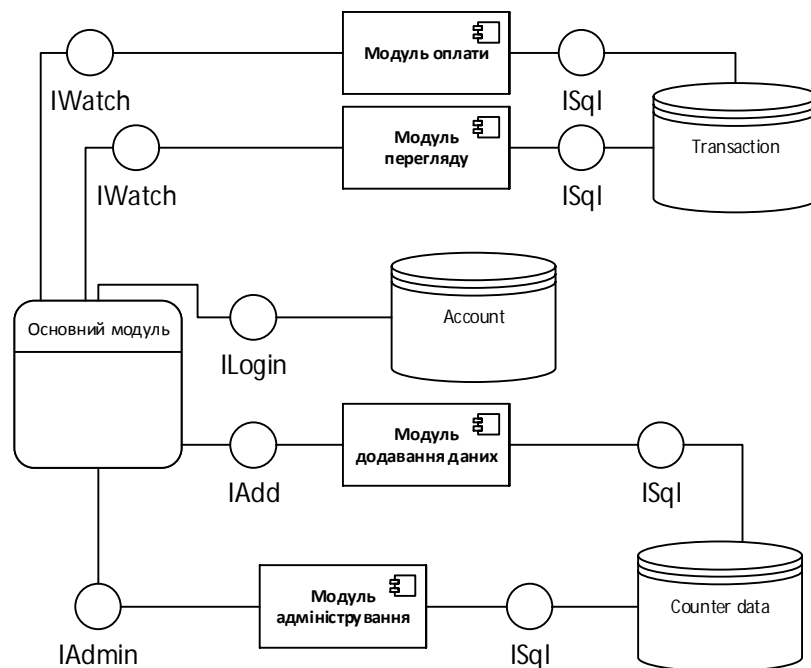


Рис. 2.6. Логічна модель системи

2.2 Проектування структури бази даних

Для розробки моделі бази даних обрана реляційна модель даних. На сьогоднішній день вона являється загальноприйнятим стандартом. Більша частина сучасних комерційних систем керування базами даних орієнтується на

реляційну модель. На відміну від мережевої та ієрархічної моделі, у реляційній моделі досягається більш високий рівень абстракції даних. До переваг цієї моделі можна віднести:

1. Використовувана інформаційна конструкція виконана у «табличному» форматі. Тому для клієнта вона достатньо зрозуміла, так як забезпечена простота та доступність.
2. Суворі правила проектування, які базуються на математичному апараті.
3. Повна незалежність даних. Незначні зміни в прикладній програмі при зміні реляційної БД. Для написання прикладного ПЗ та створення запитів у зовнішній пам'яті відсутня необхідність знати певну організацію БД.

Для опису документообігу і обробки інформації в системі використовується діаграма потоків даних, яка зображена на рисунку 2.7. На рисунках 2.8. – 2.11 зображена її декомпозиція.

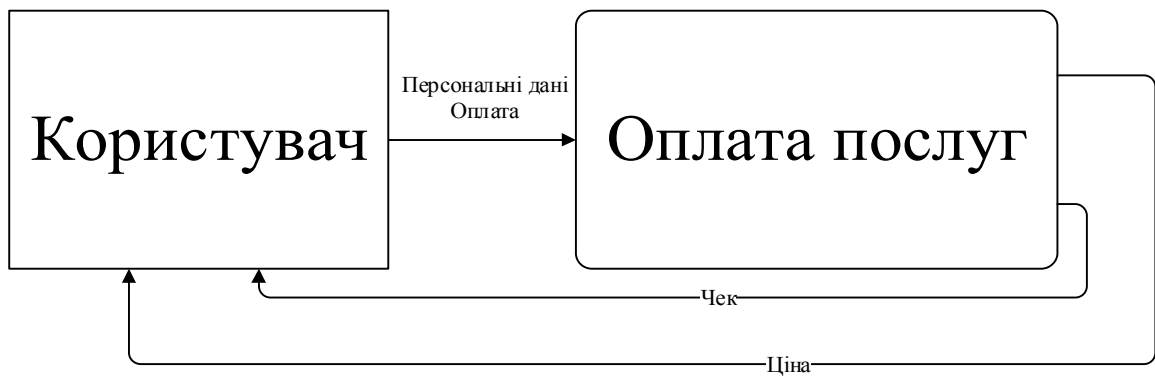


Рис. 2.7. Діаграма потоків даних верхнього рівня

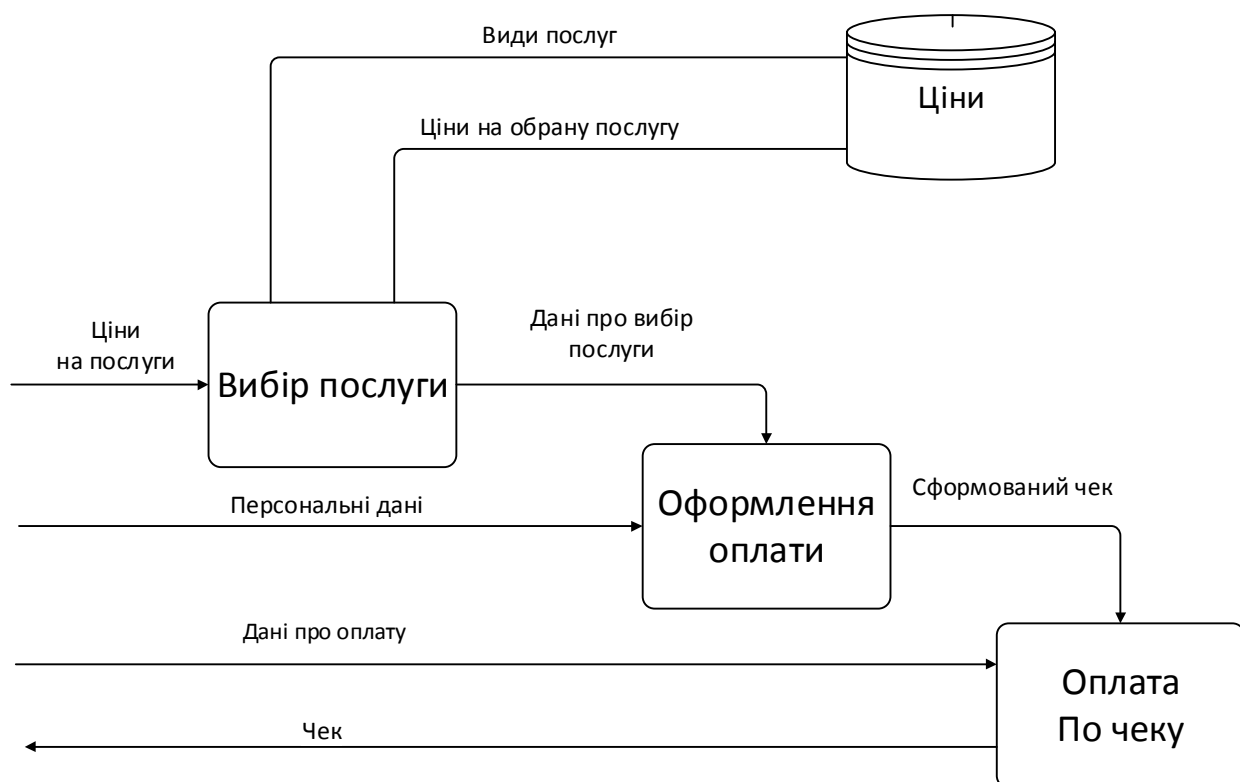


Рис. 2.8. Декомпозиція потоків даних

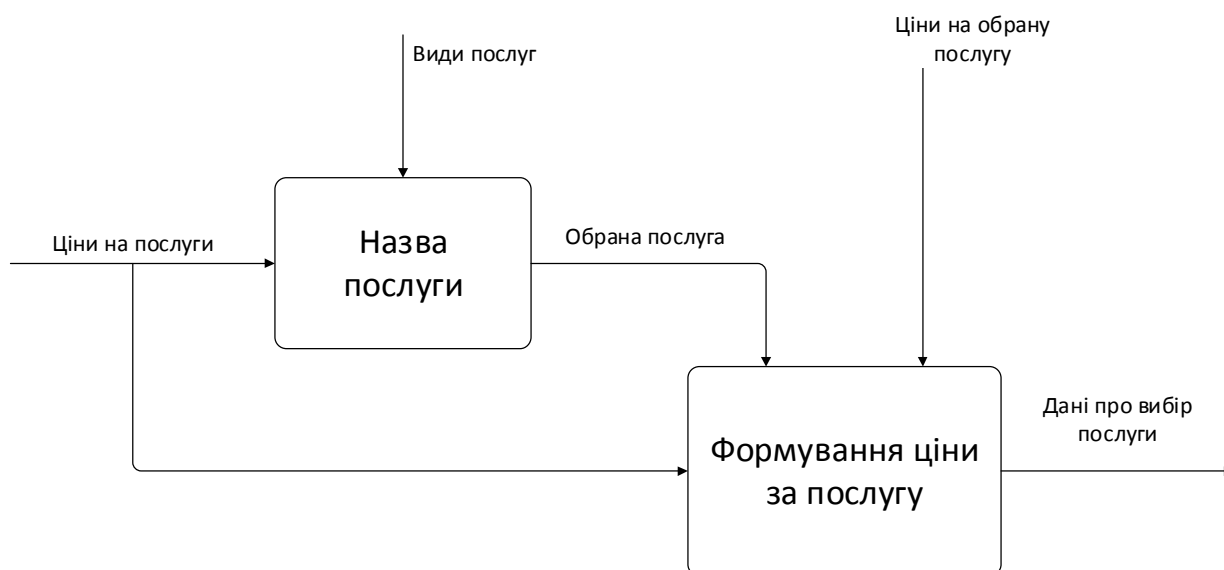


Рис. 2.9. Декомпозиція процесу «Вибір послуги»

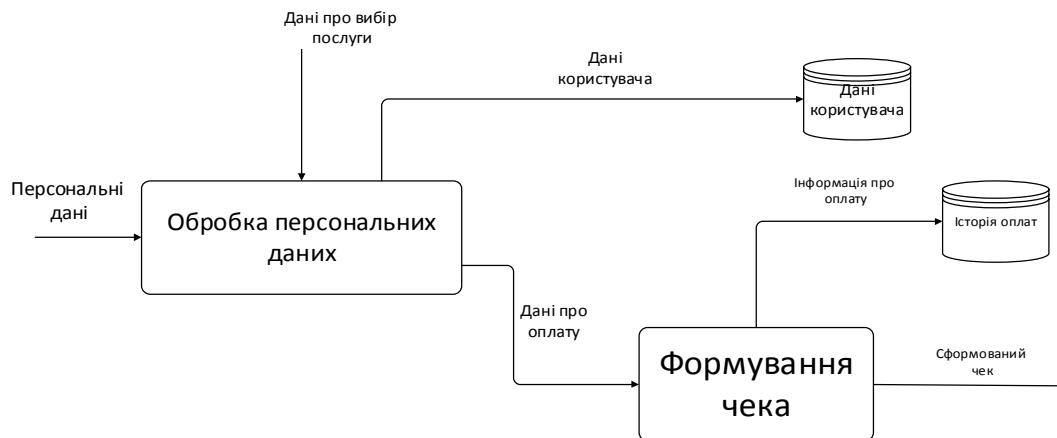


Рис. 2.10. Діаграма декомпозиції процесу «Оформлення оплати»

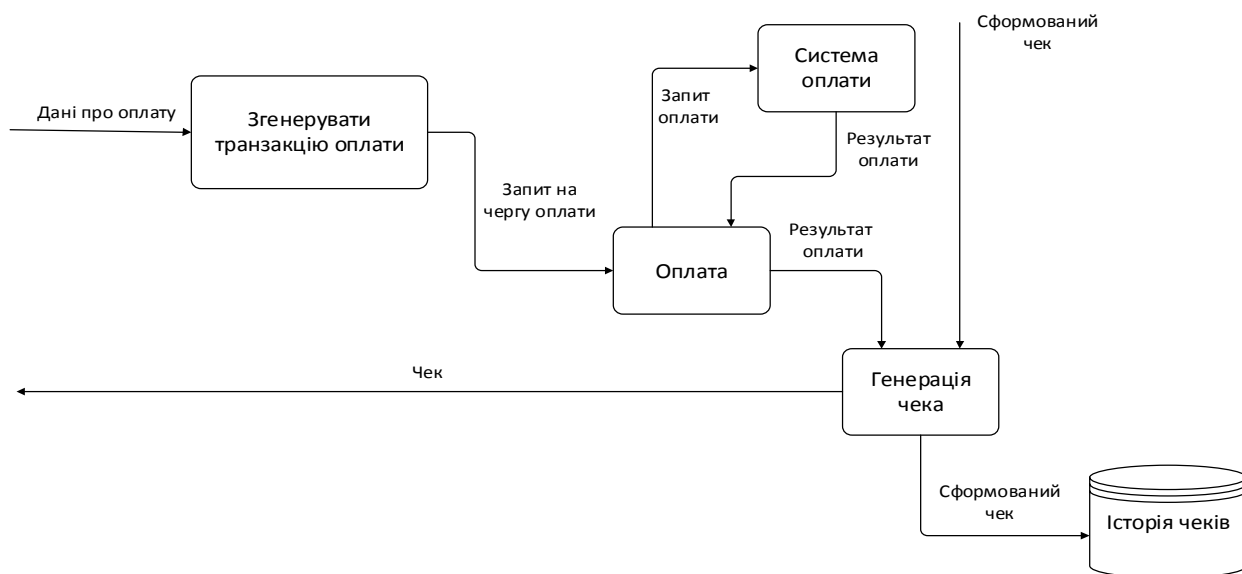


Рис. 2.11. Діаграма декомпозиції процесу «Оплата по чеку»

На вхід подаються два потоки даних, а саме авторизаційні дані і пошуковий запит від користувача. Після того, як користувач авторизувався в системі він має право відправити запит для оплати. На виході отримуємо також два потоки, це чек і інформація про тарифи.

Висновки до другого розділу :

- 1) У цьому розділі розроблено архітектуру програмного додатку, що дозволить краще зрозуміти функції основних його частин.
- 2) Розроблено та описано структурну схему, основними компонентами якої є: рівень клієнта, рівень бізнес-логіки та рівень даних. Описано функціональну структуру системи та її основних елементів – модулів обробки даних.

РОЗДІЛ 3 ПРОГРАМНА РЕАЛІЗАЦІЯ ІНТЕРНЕТ СЕРВІСУ ДЛЯ ОПЛАТИ КОМУНАЛЬНИХ ПОСЛУГ

3.1. Програмна реалізація системи

Для оплати комунальних послуг і розробки веб додатку, було обрано мову C#, технологію ASP.NET (Active Server Pages) - нова технологія, вона збільшує швидкість розробки веб додатків, роблячи її легкою і невимушеною. Додатки на ASP.NET виконуються швидше звичайних інтерпретованих мов. Проекти на ASP.NET виглядають красиво і структуровано, що полегшує їх подальше супроводження. Додатки написані на .NET це набір скриптів, які виконуються на віртуальній машині .Net Framework, тобто для роботи додатків на ASP.NET, крім самого веб сервера необхідний ще й .NET Framework. Ця технологія повністю задовольняє вимоги для зручного та швидкого розроблення продукту, щоб надалі можливо було його супроводжувати.

Виходячи з технічного завдання, програмне забезпечення можна розділити на дві частини:

- робота з БД (вибірка за певними критеріями, вставка, зміна, видалення записів);..
- графічний інтерфейс GUI - підсистема, що відповідає за обмін інформацією з користувачем (вивід на екран, отриманих записів з БД, введення даних з клавіатури, миші і т.д.).

Однією із найбільш функціональних технологій для написання Web ресурсів є asp.net, яка має розвинені засоби для програмування та обробки баз даних, і в той же час підтримує майже всі відомі формати даних.

Перевагами даної технології є:

- компіляція частин Web-додатка, які виконуються, завдяки якій вони працюють швидше, ніж інтерпретовані сценарії;
- відновлення розгорнутих Web-додатків без перезапуску сервера;

- доступ до інфраструктури .NET Framework, що доповнює можливості API Windows;
- використання відомої мови програмування Visual Basic, що тепер повністю підтримує об'єктно-орієнтоване програмування (ООП), і нової мови Visual C# - більше модифікованої версії C, що підтримує ООП і контроль типів;
- автоматичне керування станом елементів керування Web-сторінок. Завдяки цій особливості їхнє поведження мало відрізняється від звичайних елементів керування Windows;
- можливість створення нових, нестандартних серверних елементів керування на основі існуючих;
- забезпечення безпеки Web-додатків за допомогою убудованих захисних механізмів сервера Windows і інших методів аутентифікації й авторизації;
- інтеграція з ADO.NET, що дозволяє одержувати доступ до БД і застосовувати інструменти для конструювання БД із Visual Studio .NET;
- повна підтримка мови Extensible Markup Language (XML), каскадних таблиць стилів (CSS) і інших нових стандартів Web;
- убудовані механізми кешування часто запитуваних Web-сторінок на сервері, локалізації вмісту (приспосовування його для використання різних мов і регіональних стандартів) і визначення можливостей браузера.

Проект реалізований у вигляді веб-системи, в основі якого лежить клієнт-серверна парадигма роботи. Існує сервер, який виконує усі функціональні можливості, та роботою з базою даних. Є також клієнт який в свою чергу відправляє запити на сервер, і отримує веб-сторінки з певним представленим сервером функціоналом. Для користування даною парадигмою архітектури обов'язково потрібний доступ до мережі інтернет.

Для виконання проекту було обрано наступні засоби:

- ASP.NET – технологія розмітки та додавання функціональних елементів у систему. (Власне GUI)
- C# - мова яка забезпечує весь основний функціонал, обробку даних, переривань, комунікацію системи з користувачем.

- CSS – мова розмітки сторінок для форматування тексту, та елементів розмітки.

Ці засоби було обрано тому що: обґрунтуємо вибір програмних засобів

ASP.NET – це технологія яка забезпечує практично будь-які мови програмування, що входять у комплект .NET Framework (C#, Visual Basic.NET, і JScript. NET). ASP.NET має перевагу у швидкості в порівнянні із технологіями, які використовують скрипти, тому що при першому зверненні код компілюється і поміщається в спеціальний кеш, і згодом тільки виконується, не вимагаючи витрат часу на парсинг, оптимізацію, і т.д.

C# - мова програмування, яка дозволяє швидко створювати компоненти різних рівнів бізнес логіки. Ця мова є також і веб-орієнтованою: створивши будь-який компонент з простих вбудованих конструкцій, програміст може легко перетворити у Web-сервіс та звертатися за допомогою мережі Інтернет через будь-який браузер.

CSS – мова яка дозволяє розробнику формувати веб-сторінки різними стилями текстової інформації, та дозволяє управляти формами таблиць, рамок, та інше, вона є заміною верстки через таблиці для веб-сайтів. Ця мова задовольняє усім вимогам для зручного використання, можливо використовувати мову стилі для зміни одного елементу чи усього сайту, сайт після використання цієї мови стає менш об'ємним та більш структурованим, тому доцільно її використовувати, коли є можливість.

Щоб створити систему було застосовано Microsoft Visual Studio 2013. За допомогою цієї програми легко створювати зручний інтерфейс, що гарантує приємне користування системою.

Може бути представлено дві форми роботи із сайтом: у якості адміністратора та користувача. В обох випадках необхідно пройти реєстрацію або авторизацію в системі. Для адміністратора доступ до оплат, які були здійснені користувачами, є відкритим, але тільки в якості перегляду, редагування чи видалення інформації є неможливим. Для користувача, який не пройшов авторизацію, доступна інформація по тарифах, інструкція по

користуванню системою та новини. Після авторизації, клієнт може здійснювати оплату та переглядати інформацію про попередні оплати послуг.

Створення сайтів за допомогою технології ASP.NET та патерну “web - form” передбачає: розділення програмування та інтерфейсу системи та самого алгоритму продукту. У файлах з розширенням aspx.cs знаходиться код C#, а у файлах з розширенням aspx знаходиться створення елементів, їх розмітка та стилі у двох форматах ASP.NET та html .

Зображення головної сторінки веб-сайту для адміністратора і клієнта є однаковою, але при авторизації адміністратор і клієнт будуть переправлені на різні веб-форми, у яких різний функціонал.

Реалізація авторизації на сайті представлена на Рисунку 3.1.

The image shows a simple login form. It consists of two text input fields stacked vertically. The first field is labeled 'Логін' (Login) and the second is labeled 'Пароль:' (Password). Below the password field is a button labeled 'Увійти' (Login).

Рис. 3.1 Авторизація

Для авторизації було розроблено функціональний код якої наведено нижче.

```
protected void Button1_Click(object sender, EventArgs e)
{
    DataView dView =
(DataView)SqlDataSource1.Select(DataSourceSelectArguments.Empty);
    if (TextBox1.Text == "admin" && TextBox2.Text == "123")
    {
        Response.Redirect(HttpContext.Current.Request.Url.GetLeftPart(UriPartial.Authority) + "/View.aspx");
    }
    else
        if (dView.Count >= 1)
```

```

        {
            Label 3. Text = dView[0]["Id"] as string;

Response. Redirect(HttpContext.Current.Request.Url.GetLeftPart(UriPartial.Authority) + "/U
ser/UserMain.aspx?field1=" + dView[0]["Id"] as string);
        }
        else
        {
            Label 3. Text = "ERROR";
        }
    }
}
}

```

Ця частина коду реалізує функцію авторизації для адміністратора та клієнта. Якщо в першому та другому текстовому полі наявні логін та пароль адміністратора, то система направляє його в адміністраторську панель керування. В іншому випадку код створює запит до бази даних на наявність співпадання даних у текстових полях. Якщо обидва поля відповідають наявним у базі, то система зчитує порядковий номер користувача та відправляє клієнта з ідентифікатором на сторінку, яка містить інформацію про здійснені транзакції.

У наступному фрагменті коду відображається сприймання системою даних, які були передані з іншої сторінки:

```
userID = int.Parse(Request.QueryString["field1"]);
```

Перевірка наявності транзакцій для певного зареєстрованого клієнта відбувається наступним чином:

```

if (GridView1.Rows.Count >= 1)
{
    thediv.Visible = false;
}

```

Для того, щоб перевірити внесені зміни щодо здійснених операцій оплати, в першу чергу відбувається підключення до бази даних із наступним пошуком інформації про даного користувача. Відбувається зчитування його даних оплати.

Якщо у таблиці із здійсненими транзакціями відсутні дані, які беруться із запиту до бази, при умові ідентифікатора користувача, то на сайті відображається повідомлення про те, що не було здійснено жодної оплати. Якщо дані присутні, тоді приховується фрагмент сайту із повідомленням. Запит до БД показано на рисунку продемонстровано нижче:

```
SELECT Payment.Data AS [Дата оплати], Payment.Sum AS [Сума оплати],
[Transaction].PayMenthod AS [Метод оплати], Posluga.Name AS [Назва послуги],
Posluga.Coificient AS [Коефіцієнт оплати] FROM Payment INNER JOIN
[Transaction] ON Payment.TransactionId = [Transaction].Id INNER JOIN Posluga
ON Payment.PoslugaId = Posluga.Id WHERE ([Transaction].KlientId = @klient)
```

При натисненні кнопки «Оплата», відбувається перехід на форму разом із ідентифікатором, у якій користувач вводить необхідні дані. Сума оплати обраховується автоматично і залежить від типу обраної послуги, попереднього показника лічильника (якщо такий присутній) та тарифу. Тариф для певного виду комунальних послуг враховується згідно обраної користувачем. Якщо оплата за послугу була здійснена раніше, то в текстовому полі буде виводитись інформація про останній показник лічильника. На сторінці можливо здійснити одну транзакцію, яка передбачає до трьох сплат одночасно, тобто сплачувати одночасно за користування електроенергією, газом та водою.

Відображення видів послуг на сторінці відбувається за допомогою визначеного для елемента DropDownList атрибуту DataSource який посилається на об'єкт SqlDataSource, а також даного запиту в ньому «SELECT [Name], [Coificient] FROM [Posluga]».

Обрахування здійснюється автоматично після введення показників, та обраного виду послуги. Наступний фрагмент коду демонструє врахування тарифів під час розрахунку, та обчислення суми оплати згідно із введеними показниками:

```
protected void TextBox1_TextChanged(object sender, EventArgs e)
{
```

```

        SqlConnection con = new SqlConnection("Data Source=USER-
PC;Initial Catalog=JKG;Integrated Security=True");
        DataTable DT = new DataTable();
        con.Open();
        SqlCommand cmd = new SqlCommand("Select Coefficient From
Posluga Where Name = '"+n1.Text+'"', con);
        cmd.ExecuteNonQuery();
        SqlDataAdapter ad = new SqlDataAdapter(cmd);
        ad.Fill(DT);
        ad.AcceptChangesDuringFill = false;
        con.Close();
        DT.TableName = "Posluga";
        foreach (DataRow row in DT.Rows)
        {
            row.AcceptChanges();
        }
        GridView dataview = new GridView();
        dataview.DataSource = DT;
        string coi f = DT.Rows[0].ItemArray[0].ToString();
        float coi ff = float.Parse(coi f);
        float sum = (float.Parse(TextBox1.Text) -
float.Parse(p11.Text));
        sum = sum*coi ff;
        s1.Text = sum.ToString() ; }

```

При користуванні програмі формується чек, він формується залежно від того яку оплату користувач обрав, якщо оплата через банк, відкриється окрема форма з інформацією про виконаний платіж.

Важливим пунктом для користувача є додавання нових показників, вони додаються у формі оплати, та автоматично обраховуються після цього. Користувачу не потрібно переживати за не правильні показники, так як система перевіряє чи вірно вони введені.

При користуванні головним меню, наявні такі можливості:

- перейти на головну сторінку;
- інструкція по користуванню сайтом;

- про сервіс.

При натисненні кнопки «Про сервіс» ми переходимо на форму де знаходиться повна інформація про замовника сервісу, і цілі.

При переході на форму «Допомога» ми можемо бачити повну інструкцію користування сайтом де описано як відбувається оплата та розрахунок.

Хостинг в будь-якому місці для технології ASP.NET 5 і пакету програмних додатків Microsoft Visual Studio дозволяє розміщувати свій додаток на IIS або в режимі self hosting. Коли ви використовуєте Core CLR, ви можете розгорнути додаток з усіма залежностями зібраними в пакет розгортання. Таким чином, додаток і його залежності повністю автономні і більше не залежать від установки .NET в системі. Додаток може працювати на будь-якому типі пристрою або хостингової платформи. Ця нова можливість дає багато свободи. Для хостингу проекту було застосовано хмарну службу Microsoft Azure.

Реалізація хостингу на Microsoft Azure зображена на Рисунках 3.2. – 3.4.

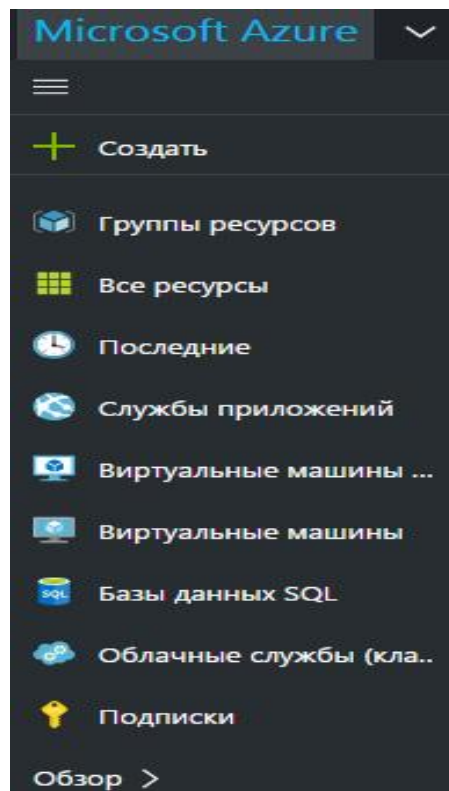


Рис. 3.2. Microsoft Azure додавання нового сервісу

На рисунках 3.2. – 3.3. зображено кроки для того щоб отримати хостинг сервер на хмарному сервісі Microsoft Azure, для того щоб сервіс працював у глобальній мережі інтернет, та звичайний користувач міг отримати доступ та користуватись ресурсом для зручності. Для того щоб отримати доступ до функціональних можливостей Microsoft Azure потрібно купити підписку користування, та оплачувати її кожен місяць.

Зручність полягає в тому що оплата здійснюється тільки за ті ресурси, якими користувач користується протягом місяця. Після проходження даного терміну Microsoft Azure виставляє рахунок користувачу, для того щоб користувач його сплатив, щоб і далі користуватись сервісом.

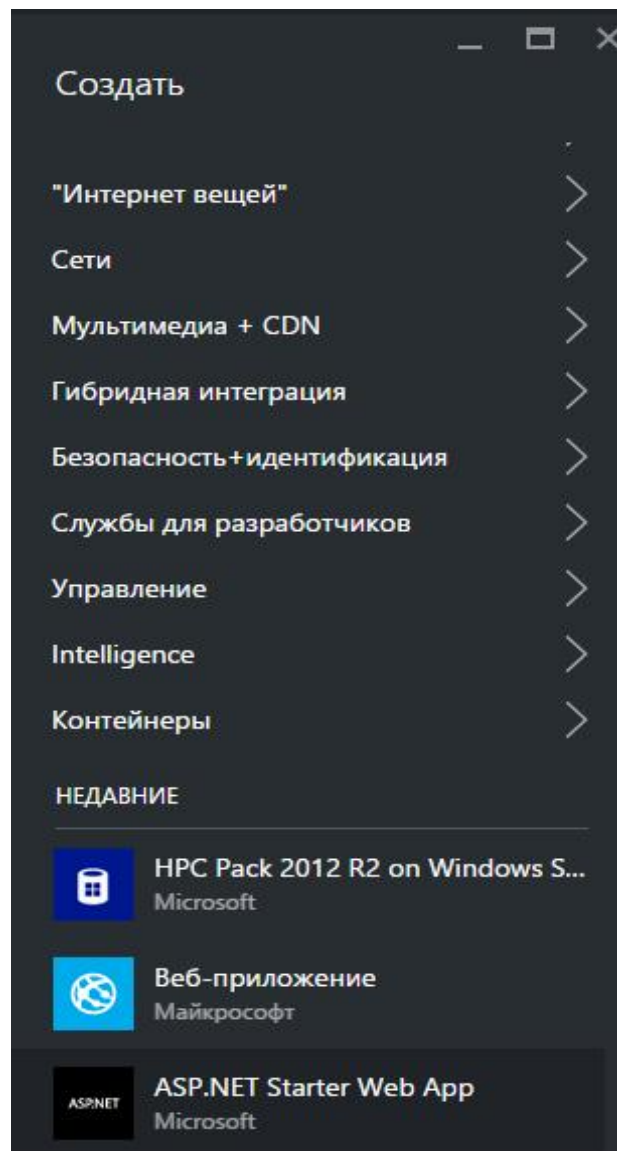
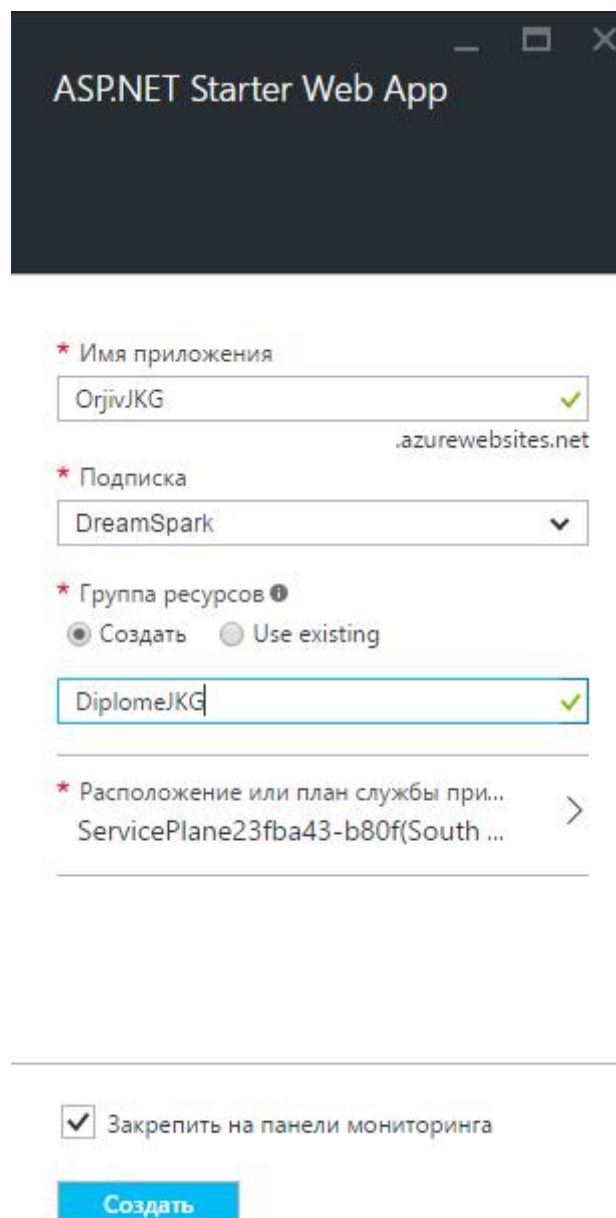


Рис. 3.3. Microsoft Azure вибір типу сервісу

У додатку Б дипломної роботи наведено повний програмний лістинг системи. Після додавання проекту до Azure, з ним можливо працювати віддалено, через Visual Studio або будь-який FTP клієнт. Також в Microsoft Azure можливий варіант компіляції з інших хмар, чи GIT, потрібно авторизувати Microsoft Azure і видати йому токен(авторизацію) для акаунту розробника, після цього Microsoft Azure автоматично скомпілює в ILL код весь проект та запустить його на вибраному сервері, якій відповідає підписці.



ASP.NET Starter Web App

* Имя приложения
OrjivJKG ✓
.azurewebsites.net

* Подписка
DreamSpark ▼

* Группа ресурсов ⓘ
 Создать Use existing

DiplomeJKG ✓

* Расположение или план службы при...
ServicePlane23fba43-b80f(South ... >

Закрепить на панели мониторинга

Создать

Рис. 3.4. Microsoft Azure налаштування параметрів публікації

3.2. Програмна реалізація бази даних

Для проектної реалізації була обрана одна із найбільш гнучких баз даних яка має необхідну СУБД, дані які можуть зберігатись можуть доходити до 1 Тб та мати при великих об'ємах даних значну продуктивність. Із рішень з відкритим вихідним кодом, найбільш розповсюдженими є MySQL.

Одним із основних застосувань MySQL є власні розробки компанії: таблиці, для яких використовувалися ISAM — підпрограми низького рівня. Після цього створили новий SQL-інтерфейс.

Станом на сьогодні, MySQL являється базою відкритим вихідним кодом, з такими збірками MySQL Community Server і комерційний Server Enterprise. Вони об'єднують наступні можливості:

- Крос-платформна підтримка.
- Процедури та функції, які зберігаються.
- Тригери.
- Курсори.
- Оновлювані подання (представлення).
- Інформаційна схема (так званий системний словник, що містить метадані).
- Підтримка SSL.
- Кешування запитів.
- Вкладені запити SELECT.
- Підтримка реплікації.
- Повноцінна підтримка Юнікоду (UTF-8 і UCS2).
- Сегментування таблиць.

MySQL є простим компактним багатопоточним сервером баз даних та характеризується великою швидкістю, стійкістю.

Ця система вважається прекрасним рішенням для малих і середніх додатків. Вихідний код сервера компілюється на безліч платформ. Для некомерційного використання MySQL є безкоштовним.

Можливості сервера MySQL:

- простота у встановленні та використанні;
- підтримується необмежена кількість користувачів, що одночасно працюють із БД;
- кількість рядків у таблицях може досягати 50 млн.;
- висока швидкість виконання команд;
- наявність простої і ефективної системи безпеки.

SQL Server автоматично налаштований на .NET Framework. Завдяки цьому, процедури і запити до БД, що зберігаються, можуть бути написані на усіх мовах які підтримує платформа .NET з використанням NUGET бібліотек, доступних для .NET Framework.

Усі таблиці і сама БД створювались за допомогою SqlServerManagement Studio. Щоб почати програмну реалізацію потрібно для початку створити базу, задати її кодування, ім'я та спосіб підключення (по замовчуванню localhost та перевірка Windows). Після цього потрібно обирати створену базу, вказати авторизаційні дані Windows для подальшого користування. Для подальшої роботи, необхідно створити таблиці - для створення таблиці потрібно відкрити наступну послідовність: SQL Server, <Назва>, Databases. Обрати БД, яка була створена раніше, натиснути праву клавішу миші та вибрати в контекстному меню пункт "New", а потім пункт "Table...". Спочатку потрібно ввести ім'я таблиці, яку створюєте та натиснути кнопку Enter. На поточному екрані з'явиться таблиця в якій потрібно ввести: ім'я колонок, тип даних, довжину, розширення NULL. Ввівши всі дані можна зберегти їх натиснувши дискету, або з випадаючого меню "File"..

DDL-код таблиці «Klient»:

```

/***** Object: Table [dbo].[Klient]  Script Date: 20.05.2016 2:07:32 *****/
SET ANSI_NULLS ON
GO
SET QUOTED_IDENTIFIER ON
GO
SET ANSI_PADDING ON

```

```

GO
CREATE TABLE [dbo].[Klient](
    [Id] [int] IDENTITY(1,1) NOT NULL,
    [Name] [varchar](50) NULL,
    [Login] [varchar](50) NULL,
    [Password] [varchar](50) NULL,
    [Adress] [varchar](50) NULL,
    [KvNum] [int] NULL,
    [Mail] [varchar](50) NULL,
    CONSTRAINT [PK_Klient] PRIMARY KEY CLUSTERED
(
    [Id] ASC
)WITH (PAD_INDEX = OFF, STATISTICS_NORECOMPUTE = OFF, IGNORE_DUP_KEY =
OFF, ALLOW_ROW_LOCKS = ON, ALLOW_PAGE_LOCKS = ON) ON [PRIMARY]
) ON [PRIMARY]
GO
SET ANSI_PADDING OFF
GO

```

Таблиця створена для збереження даних про усіх клієнтів, які зареєстровані в системі. Всі поля повинні бути заповненні та дані введено коректно до їх типу і довжини.

Приклад виконання коду представлено на рисунку 3.5.

Id	Name	Login	Password	Adress	KvNum	Mail
1	Петро	Chumak	827ccb0eea8a7...	Більцька 8	1	Kozak@mail.ru
2	Оксана	Oksana	81dc9bdb52d04...	Чорновола 10	2	Ksana@gmail.c...
3	Людмила	Babiy	202cb962ac590...	Симоненка 6	3	Ludochka@gm...
1002	Ірина	Ira	202cb962ac590...	Карпенка 4	4	irina@gmail.com
1005	Настя	Nastya	202cb962ac590...	Живова 7	32	Nastyuysha959...

Рис. 3.5. Таблиця «Klient» в базі даних

Рисунок 3.5. відображає тільки ті дані, які коректно введені клієнтом ЖКГ – для подальшої оплати послуг, що в база даних є усі колонки, які наведені в кодї. Дані введено коректно, тому були збережені в базі даних.

Проектуючи веб-систему по оплаті комунальних платежів необхідно створити таблицю «Оплата», в зберігаються такі дані як: дата оплати, сума

оплати, посилання на тип послуги, і посилання на транзакцію (для здійснення мультиваріантивних платежів з декількома платежами). DDL-код таблиці «Payment»:

```

/***** Object: Table [dbo].[Payment]  Script Date: 20.05.2016 2:07:32 *****/
SET ANSI_NULLS ON
GO
SET QUOTED_IDENTIFIER ON
GO
CREATE TABLE [dbo].[Payment](
    [Id] [int] IDENTITY(1,1) NOT NULL,
    [Data] [nchar](10) NULL,
    [Sum] [float] NULL,
    [PoslugaId] [int] NULL,
    [TransactionId] [int] NULL,
    CONSTRAINT [PK_Payment] PRIMARY KEY CLUSTERED
(
    [Id] ASC
) WITH (PAD_INDEX = OFF, STATISTICS_NORECOMPUTE = OFF, IGNORE_DUP_KEY =
OFF, ALLOW_ROW_LOCKS = ON, ALLOW_PAGE_LOCKS = ON) ON [PRIMARY]
) ON [PRIMARY]
GO

```

Приклад виконання даного коду зображено на рисунку 3.6.

	Id	Data	Sum	PoslugaId	TransactionId
▶	1	20.04.2016	200	1	!

Рис. 3.6. Таблиця «Payment» в базі даних

У системі також передбачено реєстрацію показників. Кожен показник пов'язаний з певним клієнтом та послугою, для того щоб була можливість використовувати автоматичне обчислення при авторизації користувача у системі. Реалізація DDL-коду таблиці «Pokaznik»:

```

/***** Object: Table [dbo].[Pokaznik]  Script Date: 27.05.2016 16:07:32 *****/
SET ANSI_NULLS ON

```

```

GO
SET QUOTED_IDENTIFIER ON
GO
CREATE TABLE [dbo].[Pokaznik](
    [Id] [int] IDENTITY(1,1) NOT NULL,
    [PoslugaName] [nvarchar](max) NULL,
    [Num] [float] NULL,
    [KlientID] [int] NULL,
    CONSTRAINT [PK_Pokaznik] PRIMARY KEY CLUSTERED
(
    [Id] ASC
)WITH (PAD_INDEX = OFF, STATISTICS_NORECOMPUTE = OFF, IGNORE_DUP_KEY =
OFF, ALLOW_ROW_LOCKS = ON, ALLOW_PAGE_LOCKS = ON) ON [PRIMARY]
) ON [PRIMARY] TEXTIMAGE_ON [PRIMARY]
GO

```

Виконання даного коду в базі даних представлено на рисунку 3.7.

	Id	PoslugaName	Num	KlientID
	1	Вода	478,333	2
	2	Електроенергія	98,146	2
	3	Газ	257,793	3
	4	Вода	387,174	3
	5	Електроенергія	67,741	3
	6	Газ	366,821	1
	7	Вода	412,107	1

Рис. 3.7. Таблиця «Pokaznik» в базі даних

Наступна реалізація DDL-коду відображає таблицю доступних для оплати послуг, в майбутньому вона може розширюватись для збільшення функціональних можливостей сервісу, та покращення зручності клієнтів.

```

/***** Object: Table [dbo].[Posluga]  Script Date: 27.05.2016 16:07:32 *****/
SET ANSI_NULLS ON
GO
SET QUOTED_IDENTIFIER ON
GO
SET ANSI_PADDING ON
GO
CREATE TABLE [dbo].[Posluga](
    [Id] [int] IDENTITY(1,1) NOT NULL,

```

```

[Name] [varchar](50) NULL,
[Coificient] [float] NULL,
[NumForPayment] [int] NULL,
CONSTRAINT [PK_Posluga] PRIMARY KEY CLUSTERED
([Id] ASC
)WITH (PAD_INDEX = OFF, STATISTICS_NORECOMPUTE = OFF, IGNORE_DUP_KEY = OFF,
ALLOW_ROW_LOCKS = ON, ALLOW_PAGE_LOCKS = ON) ON [PRIMARY]
) ON [PRIMARY]
GO
SET ANSI_PADDING OFF
GO

```

Транзакції у системі пов'язують користувача який оплачує послуги ЖКГ, та оплати які він здійснює у цій системі, для того щоб було можливо зробити мультиоплату(Водночас платити за декілька послуг). Реалізація Transaction представлена в наступному коді.

```

/***** Object: Table [dbo].[Transaction]  Script Date: 27.05.2016 16:07:32 *****/
SET ANSI_NULLS ON
GO
SET QUOTED_IDENTIFIER ON
GO
SET ANSI_PADDING ON
GO
CREATE TABLE [dbo].[Transaction](
    [Id] [int] IDENTITY(1,1) NOT NULL,
    [PayMenthod] [varchar](50) NULL,
    [KlientId] [int] NULL,
    CONSTRAINT [PK_Transaction] PRIMARY KEY CLUSTERED
(
    [Id] ASC
)WITH (PAD_INDEX = OFF, STATISTICS_NORECOMPUTE = OFF, IGNORE_DUP_KEY = OFF,
ALLOW_ROW_LOCKS = ON, ALLOW_PAGE_LOCKS = ON) ON [PRIMARY]
) ON [PRIMARY]

```

Виконання даного коду в базі даних представлено на рисунку 3.8

	Id	PayMenthod	KlientId
	1	MasterCard	1002

Рис. 3.8. Таблица «Transaction» в базі даних

Можливий режим текстової роботи у базі даних, якщо увести команду, повертається вибір з таблиці, поля заповнюються даними, вони відповідають команді, яка була введена для вибірки.

Метою створення фізичної моделі є важливою інформацією для оновлення чи відновлення повної схеми усієї роботи, для тих хто працює та впроваджує базу даних у СКБД.

Microsoft SQL Management studio 2014 підтримує автоматичну генерацію будь-яких діаграм, при підключенні до певної бази, у яких наявні фізична інформація та мета-дані. Отримана фізична модель даних представлена на рисунку 3.9.

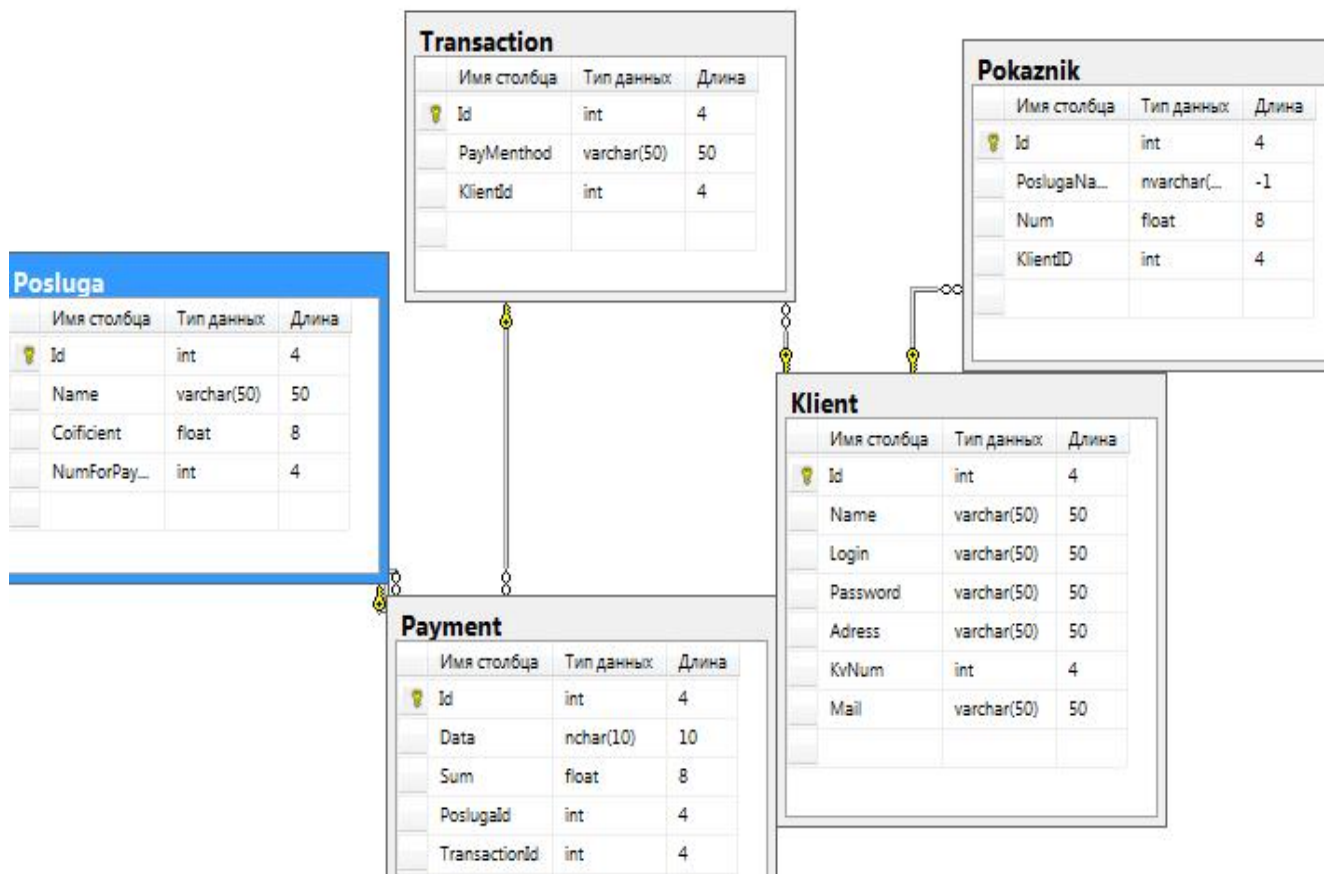


Рис. 3.9. Фізична модель даних

У додатку В дипломної роботи представлено DDL statement бази даних.

Висновки до третього розділу:

1. У даному розділі обґрунтовано технологію, мову програмування та розроблено програмну систему.
2. Обґрунтовано засоби розробки бази даних та створено програмну реалізацію бази даних програмної системи за допомогою механізму збережених процедур.

РОЗДІЛ 4 ТЕСТУВАННЯ ТА ДОСЛІДНА ЕКСПЛУАТАЦІЯ

4.1. Тестування

Тестування є важливим етапом ЖЦ системи в цілому. Існують такі основні методи тестування для системи:

Функціональне тестування: Модульне тестування (Test case) було виконано вручну; Автоматизоване тестування (Стрес тест) виконано програмним забезпеченням Selenium;

Usability testing – серед користувачів, та тестерів системи було проведено анкетування, для визначення зручності та оптимальності використання системи.

Наступні методи були призначення для дослідження роботи окремих модулів і використовуються під час програмування. Модульне тестування досліджує роботу як окремих модулів системи так і визначення помилок в ній. Test case – вказується які дані потрібно ввести та як саме натиснути на кнопку щоб тестування було ефективним, и було можливим визначення помилок які можуть виникати у ході роботи системи, для їх подальшого усунення. Отримана Test case системи зображена на рисунку 4.1.

Steps	Test steps	Test Data	Expected result	Actual restul	Status(Pass/Fail)
1	Login page		Admin redirect to adminPanel	Admin successful login	Pass
	Provide username	Admin			
	Provide password	412446			
	Click"Увійти"				
2	Login page		User redirect to adminPanel with ID	User successful login witch get ID	Pass
	Provide username	Nastya			
	Provide password	1234ds			
	Click"Увійти"				
3	Click"Допомога"		Redirected to instruction.aspx	User successful view page	Pass
4	Click"Про нас"		Redirected to About.aspx	User successful view page	Pass
5	Click"Реєстрація"		System register new user	User successful registered	Pass
	Provide Valid text field Ім'я	Tester			
	Provide Valid text field Логін	Test			
	Provide Valid text field Пароль	245456			
	Provide Valid text field Підтвердіть пароль	245456			
	Provide Valid text field Адреса	Tested 5			
	Provide Valid text field Номер квартири	58			
Provide Valid text field @mail	test@gmail.com				
	Click"Зареєструватись"				

Рис. 4.1. Test case системи

Функціональне тестування системи «Електронний-платник»

Призначене для виявлення основних функцій та пріоритету їх роботи в системі.

Таблиця 4.1

Таблиця функціонального тестування

ID	1	2	3
PRIORITY	Обов'язкова	Обов'язкова	Опційна
TESTING FUNCTIONALITY	Здійснити оплату	Перегляд транзакцій	Використати оплату банку
PRE-CONDITIONS	Підключення до Інтернет мережі активне і успішне підключення до сервера	Підключення до Інтернет мережі активне і успішне підключення до сервера	Підключення до Інтернет мережі активне і успішне підключення до сервера
POST-CONDITIONS	Оплата здійснена і додана у БД	Клієнт прееглянув усі транзакції	Оплата здійснена і додана у БД
TEST'S STEPS	1. Відкрити сайт 2. З'єднатися з сервером . 3. Пройти авторизацію 4. Вибрати вкладку «Здійснити оплату» в меню 4.Ввести дані про оплату 5.Натиснути кнопку «Оплатити»	1. Відкрити сайт 2. З'єднатися з сервером . 3. Пройти авторизацію 4. Переглянути усі оплати	1. Відкрити сайт 2. З'єднатися з сервером . 3. Пройти авторизацію 4. Вибрати вкладку «Оплата» меню 4.Вибрати зі списку потрібний спосіб оплати 5.Натиснути «Оплатити»
EXPECTED RESULT	Повинно бути збережено нову оплату	Користувач переглянув свої оплати(транзакції)	Повинно бути збережено нову оплату
ACTUAL RESULT	Збережено нову оплату і усі введені дані	Користувач зміг переглянути усі своїх транзакції	Збережено нову оплату і усі введені дані
STATUS	Протестовано	Протестовано	Протестовано
DISCOVERED BUGS	Не знайдено	Не знайдено	Не знайдено

Продовження таблиці 4.1

ID	4	5	6
PRIORITY	Опційна	Обов'язкова	Опційна
TESTING FUNCTIONALITY	Здійснити пошук потрібної оплати	Додати нові покази	Переглянути тарифи
PRE-CONDITIONS	Підключення до Інтернет мережі активне і успішне підключення до сервера	Підключення до Інтернет мережі активне і успішне підключення до сервера	Підключення до Інтернет мережі активне і успішне підключення до сервера
POST-CONDITIONS	Відображено дані відносно пошуку	Будуть додані нові покази лічильника	Буде відображено актуальні тарифи
TEST'S STEPS	1. Відкрити сайт 2. З'єднатися з сервером. 3. Пройти авторизацію 4. Ввести дані про оплату у рейс. 5. Натиснути кнопку «Знайти»	1. Відкрити сайт 2. З'єднатися з сервером . 3. Пройти авторизацію 4. Вибрати вкладку «Покази» в меню. 4. Обрати певну послугу. 5. Додати потрібні дані. 5. Натиснути кнопку «Додати».	1. Відкрити сайт 2. З'єднатися з сервером . 3. Пройти авторизацію 4. Вибрати вкладку «Тарифи» в меню 4. Вибрати зі списку потрібний 5. Натиснути «Переглянути»
EXPECTED RESULT	Відображено дані відносно пошуку	Покази лічильника збережені	Звіти відносно тарифів повинні бути відображені
ACTUAL RESULT	Показано всі дані відносно пошуку	Покази лічильника успішно збереглись в БД	Показано усі тарифи на актуальний термін
STATUS	Протестовано	Протестовано	Протестовано
DISCOVERED BUGS	Не знайдено	Не знайдено	Не знайдено

Для функціонального тестування було створено 20 випадків, з них у звітності наведено 6, усі мали успішний результат, тому можна зробити висновок, функціонал системи повністю відповідає вимогам, які описані в тех-завданні.

Автоматизоване тестування

Тестування було проведено за допомогою програмного продукту Selenium. Автоматизоване тестування призначене для автоматизації виконаних дій на сайті. Було протестоване функціональне забезпечення сайту яке складалось із 6 випадків з яких усі пройдено успішно. Тестування можна вважати вдалим.

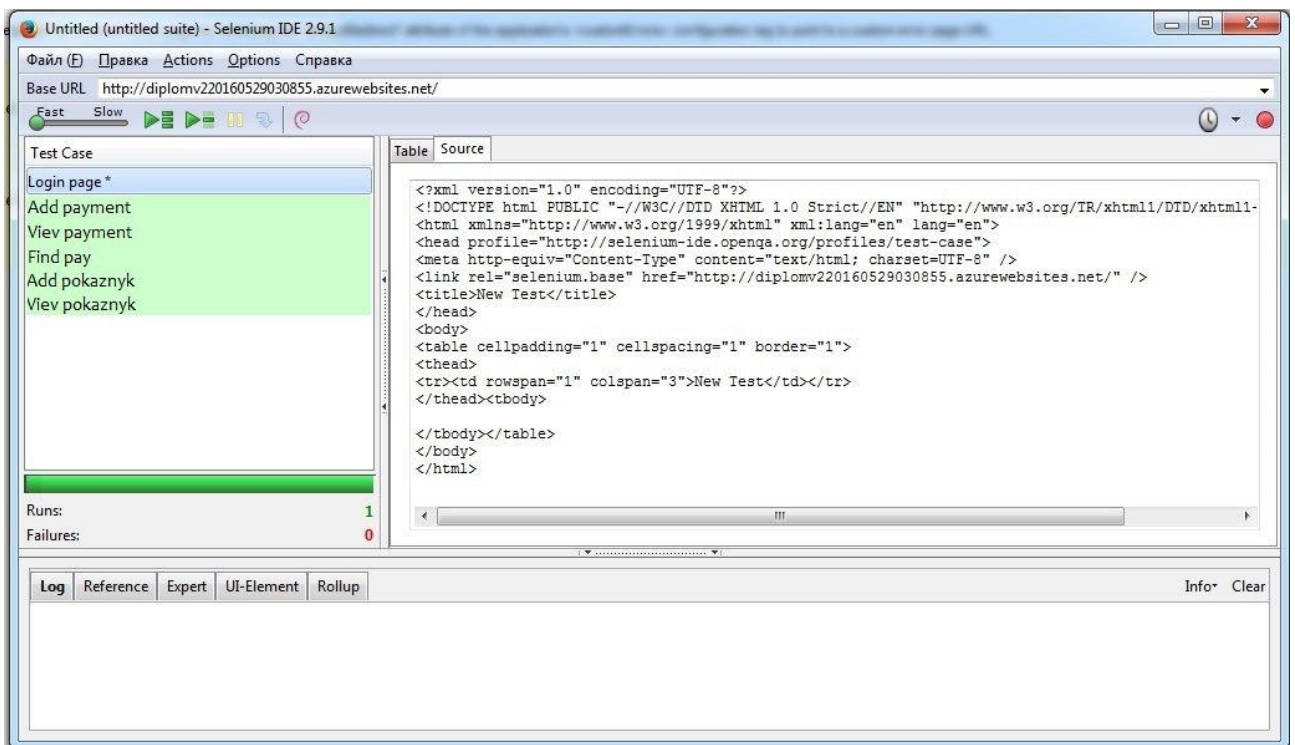


Рис. 4.2. Результат виконання одного з тестів

Після проведеного тестування веб-сервісу різними браузерами контент і інтерфейс відображався правильно та без помилок. Всі основні модулі сайту були відображені вірно та в робочому стані.

4.2. Розгортання програмного продукту

Система «Електронний платник» повинна бути розміщена на сервері. Для доступу до сайту клієнту чи адміністратору потрібен комп'ютер чи ноутбук з встановленим будь-яким браузером та обов'язковим доступом до мережі Інтернет. Для роботи на сайті не потрібно встановлювати додаткових програм чи драйверів.

Вимоги до сервера:

- Програмне забезпечення:
 - операційна система Windows 7 x64 (Windows Server 2012 x64);
 - MS SQL Server 2012;
- Апаратне забезпечення:
 - центральний процесор Intel® Core™ i7-4600 Processor 3,4ГГц;
 - оперативно-запам'ятовуючий пристрій із об'ємом пам'яті 8Гб;
 - жорсткий диск з вільними 100 Гб пам'яті.

Вимоги до клієнта:

- Програмне забезпечення:
 - Windows 7 x32-x64 або вище;
 - Веб-браузер ([Google Chrome](#), [Internet Explorer](#), [Firefox](#), [Safari](#) і [Opera](#).)
- Апаратне забезпечення:
 - центральний процесор PentiumDualCoreCPU T4300 2,1 ГГц;
 - оперативно-запам'ятовуючий пристрій із об'ємом пам'яті 2Гб;
 - жорсткий диск з вільними 500Мб пам'яті.

4.3. Інструкція користувача

Для роботи на сайті користувачеві потрібно:

1. Підключитися до мережі Інтернет.
2. Відкрити браузер.
3. Ввести URL-адресу сайту.
4. Дочекатися відкриття сайту.

Коректна робота системи буде забезпечена лише при стабільному підключенні до Інтернет мережі.

У цій системі розроблено наступні функції:

- Відображення тарифів;
- Можливість пошуку оплати за введеними даними;
- Можливість здійснити оплату;
- Вхід клієнта;
- Вхід під адміністратором;
- Додавання показів;
- Перегляд інструкції;
- Перегляд новин;
- Можливість контролю даних адміністратором системи.

Для відображення усіх оплат користувача потрібно перейти на вкладку «Головна користувача». Користувач може здійснити пошук оплати за введеним словом, який зображено на рисунку 4.3. Пошук оплат розміщено на головній сторінці сайту для зручності пошуку.

Дата оплати	Сума оплати	Метод оплати	Назва послуги	Коефіцієнт оплати
20.04.2016	200	MasterCard	Електроенергія	0,99

Здійснити Оплату
Фільтр

Рис. 4.3. Головна користувача

Для оплати клієнту потрібно перейти на вкладку «Здійснити оплату», яке зображено на рисунку 4.4. Для здійснення оплати йому потрібно заповнити усі поля та натиснути кнопку «Оплатити.»

Назва послуги	Показник и	Сумма	Спосіб оплати
Електроенергія	67.741	2,236413	MasterCard/Visa
Газ	70	303,0463	
Вода	257.793	80,80373	
	300		
	387.174		
	400		
			Загальна сума
			386,0864

Buttons: Оплатити, Обрахувати

Payment form fields: XXXX XXXX XXXX XXXX, CVC2/CVV2: XXX

Рис. 4.4. Сторінка оплати для клієнта

У вкладці «Показники» також є форма додавання нового показника щоб була можливість додати стартові данні для користувача. Усі поля повинні бути заповненні та надіслані дані для додавання, якщо клієнт не додав усі дані, йому не буде доступна форма оплати, тому що для оплати потрібні початкові показники для обрахунків. Форма додавання нових показників зображена на рисунку 4.5.

Назва послуги	Показник и
Електроенергія	67.741
	97,123

Рис. 4.5. Сторінка додавання нових показників для клієнта

Для надання прав адміністратора потрібно пройти авторизацію, тобто ввести певний логін і пароль. Також і клієнту потрібно пройти авторизацію для того щоб його здійснені оплати коректно додавались до БД, для подальшого перегляду. Сторінка авторизації зображена на рисунку 4.6.

Логін

Пароль:

Рис. 4.6. Вікно авторизації

Якщо було виведено повідомлення про помилку при введенні логіну чи паролю, то спробуйте їх ввести знову.

Усім користувачам системи завжди доступні такі пункти меню для їх перегляду

- Інструкція користувача
- Про сервіс
- Головна

При переході на інструкцію користувача користувач може читати як саме відбувається оплата, у яких термінах, інструкція зображена на рисунку 4.7. – 4.9.

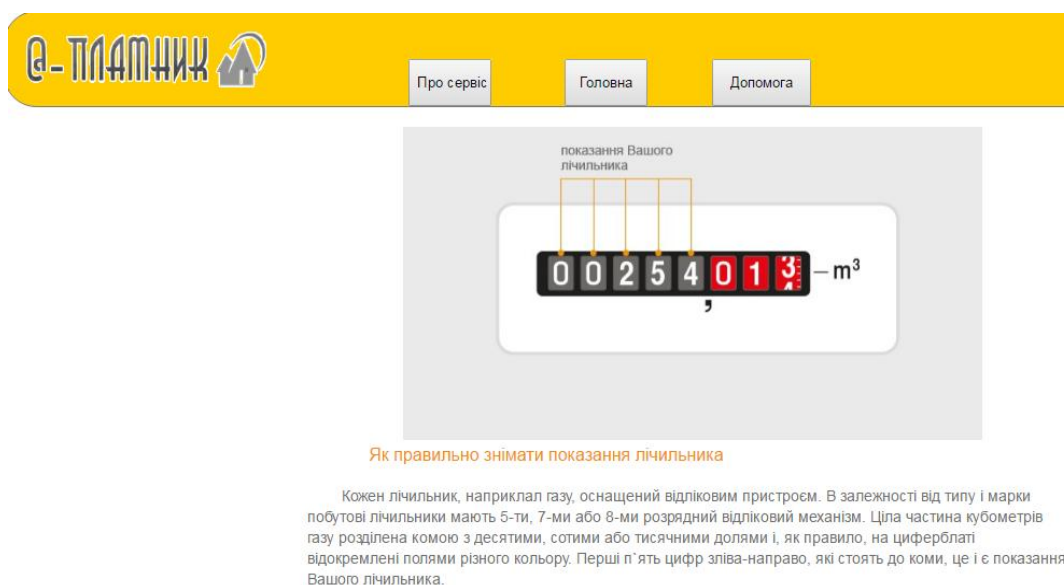


Рис. 4.7. Інструкція користувача «Покази лічильника»



Як записати показання і на що звернути увагу

На циферблаті показання 00254,013. Запишіть 5 номерів, показаних зліва-направо. Не звертайте уваги на цифри після коми, які можуть бути на полі іншого кольору. В розрахункову квитанцію впишіть цифри 254, два нулі попереду не записуються.

Якщо показання останньої цифри цілої частини знаходяться між двома цифрами, Ви вправі вирішувати, яке показання записати – на півкуба більше чи менше. Коли показання зупинилися між 9 і 0, запишіть попереднє число, тобто 9.



Рис. 4.8. Інструкція користувача «Покази лічильника»

Після переходу на вкладнику «про сервіс» користувач сервісу може переглянути основні функції та інформацію про сервіс який їм надається. Вкладника «Про сервіс» зображена на рисунку 4.9

e-ПЛАТНИК

[Про сервіс](#)
[Головна](#)
[Допомога](#)

Тепер вам не потрібно гурбуватися через пропущені квитанції чи недоайливих листинош. Електронний платник отримує рахунки за вас прямо від постачальника послуг.

Ми доставляємо сплачені квитанції вам за допомогою пошти України раз на квартал. Вся звітність завжди при вас. Також ми висилаємо квитанції за запитом клієнта в будь-який час за невелику плату.

Не завжди встигаєте вчасно дійти до каси і отримуєте штрафні санкції? Забудьте про це раз і на завжди! Електронний платник дозволяє налаштувати дати оплати рахунків.

Ласкаво просимо на сайт Електронний платник

Електронний платник - це сервіс прийому платежів в інтернеті за допомогою якого мешканці смт. Оржів мають можливість сплачувати комунальні послуги, використовувати банківські карти.
Місія:
Сприяти розвитку ринку платежів і його безпеці;
Популяризувати оплати через інтернет серед населення Оржівіа;
Впроваджувати новітні технології та методи захисту карткових даних, забезпечувати максимальну безпеку платників під час інтрнет-платежів;
Розвивати власні методи від інтернет-шахрайства та пропонувати їх ринку з метою захисту клієнтів.

Рис. 4.9. Вкладника «Про сервіс»

Також користувачу сервісу «Електронний-платник» має доступ до головної сторінки з такими функціями як, авторизація, реєстрація, та перегляд новин.

Адміністратор так як і клієнт повинен пройти процедуру авторизації для того щоб отримати доступ до головних функцій системи. Щоб авторизуватись клієнт повинен для початку зареєструватись увести усі потрібні дані, які потрібні системі для того щоб коректно опрацьовувати транзакції користувача. Головна сторінка сайту відображена на рисунку 4.10.

e-Платник

Про сервіс Головна Допомога

Сервіс прийому платежів онлайн
Все що необхідно -
пара хвилин вільного часу

Реєстрація **Вхід**

Новини


Оплатити комунальні : банк , пошта або Інтернет ?

Оплата комунальних послуг - справа обов'язкова, але не завжди приємна. Особливо тому що займає чимало часу. Своєчасна оплата може бути справжньою проблемою для тих, хто живе активно і у кого розписаний весь день, хто затиснутий вимогами робочого графіка, хто буває у відрядженнях або, навпаки, прив'язаний до місця проживання. Для погашення рахунку «по-старому» - в банку або поштовому відділенні, необхідно виділити кілька годин на одну тільки оплату комунальних послуг. А більш сучасний підхід - оплата через Інтернет - все ще залишається незвичним для багатьох клієнтів. Немає прив'язки до конкретних робочих годинах каси. Економічно вигідно. Онлайн - сервіси стягують доступні комісії за вчинення платежів.


В Європі кожен п'ятий власник карти Visa платить безконтактно

Рис. 4.10. Інструкція користувача «Покази лічильника»





На рис 4.11 – 4.12 зображено зміну функціональності на головній формі користувача, залежно від наявності в системі його платежів.


Про нас Головна Допомога

Здійснити Оплату


Інформація по платежах відсутня


До оплати приймаються:

Зберігайте свій час разом з нами!

© 2016р @-Platnik
Технічна підтримка: Vlad.Zaech@gmail.com





Рис. 4.11. Головна форма перегляду здійснених платежів при відсутності проведених транзакцій


Про нас Головна Допомога

Дата оплати	Сума оплати	Метод оплати	Назва послуги	Коефіцієнт оплати
20.04.2016	200	MasterCard	Електроенергія	0,99

Здійснити Оплату

До оплати приймаються:

Зберігайте свій час разом з нами!

© 2016р @-Platnik
Технічна підтримка: Vlad.Zaech@gmail.com

Рис. 4.12. Головна форма перегляду здійснених платежів при наявності проведених транзакцій.

Висновок до четвертого розділу:

- 1) Проведено три тестування даної системи та описано їх результати;
- 2) Визначено особливості даної системи для нормального функціонування;
- 3) Створено інструкції для користувача.

ВИСНОВКИ

Під час виконання дипломної роботи була досягнена мета - створення web-сайту для оплати комунальних послуг. Доступ до сайту будуть мати користувачі персональних комп'ютерів та ноутбуків з активним підключенням до мережі Інтернет. Система повністю відповідає створеному переліку вимог.

Аналізуючи предметну область сайту "Електронний платник" було розглянуто три системи-аналоги, в яких виділено недоліки та функціональні можливості. На основі аналізу існуючих систем створено специфікацію вимог до проекту.

Створення програмної системи складається з проектування архітектури та бази даних сайту "Електронний платник ».

Кодування розроблюваної системи складається з двох етапів: програмна реалізація системи мовою програмування C# за допомогою технології ASP.NET та проектування бази даних SQL Server 2012

Для тестування системи було використано сучасні види функціонального тестування такі як: ручне тестування (Test case), і автоматизоване тестування за допомогою програми (Selenium), також за допомогою цього програмного продукту було проведено стресове тестування, для визначення середнього часу безвідмовної роботи. Також було проведено Usability testing (анкетування), визначили, що система досить зручна та зрозуміла у використанні. Всього було виконано 35 тестів.

Проведено детальний опис вимог до програмного забезпечення та характеристик сервера на якому система буде нормально функціонувати.

Система «Електронний платник» передбачає два користувача: адміністратора та клієнта, тому було створено детальний опис інструкцій користувача, в якій наведено зображення екранів основних функцій системи.

СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ

1. Ландэ Д. В., Снарский А. А., Безсуднов И. В. Интернетика: Навигация в сложных сетях: модели и алгоритмы. — М.:Либроком (Editorial URSS), 2009. — 264 с. ISBN=978-5-397-00497-8.
2. Буров Є. В. Комп'ютерні мережі: підручник / Євген Вікторович Буров. — Львів: «Магнолія 2006», 2010. — 262 с. ISBN 966-8340-69-8
3. Ландэ Д. В., Снарский А. А., Безсуднов И. В. Интернетика: Навигация в сложных сетях: модели и алгоритмы. — М.:Либроком (Editorial URSS), 2009. — 264 с. ISBN=978-5-397-00497-8.
4. Комп'ютерні мережі: [навчальний посібник] / А. Г. Микитишин, М. М. Митник, П. Д. Стухляк, В. В. Пасічник. — Львів: «Магнолія 2006», 2013. — 256 с. ISBN 978-617-574-087-3
5. Инженерное программирование для проектирования программного обеспечения. -М.: Радио і связь, 1985, -512с.
6. Бойков.В., Савинков В.М. Проектирование баз данных информационных систем. М. Мир 1997
7. Бердтис А. Структуры данных. - М.: Статистика, 1974, - 408 с.
8. Горбань О.М., Бахрушин В.Є. Основі теорії систем та системного аналізу. - Запоріжжя, ГУ "ЗІДМУ", 2004, ISBN 966-8227-23-9
9. Майо Д. Самоучитель Microsoft Visual Studio 2010 = Microsoft Visual Studio 2010: A Beginner's Guide (A Beginners Guide). — С.: «БХВ-Петербург», 2010. — С. 464. — ISBN 978-5-9775-0609-0
10. Алекс Макки Введение в .NET 4.0 и Visual Studio 2010 для профессионалов = Introducing .NET 4.0: with Visual Studio 2010. — М.: «Вильямс», 2010. — С. 416. — ISBN 978-5-8459-1639-6
11. Кен Хендерсон Професійне керівництво з SQL Server: структура та реалізація. — М.: Издательский дом «Вильямс», 2006. — С. 1056. ISBN 5-8459-0912-0

12. Чураков Михаил. Муравьиные алгоритмы [Электронный ресурс] / Михаил Чураков, Андрей Якушев. // Режим доступа: <http://rain.ifmo.ru/cat/data/theory/unsorted/ant-algo-2006/article.pdf>.
13. Бормашов Д. А. “Кластерный анализ текстов”: Дипломная работа [Электронный ресурс] /Д. А. Бормашов. Режим доступа: <http://inf.tsu.ru/library/DiplomaWorks/CompScience/2006/bormashov/diplom.pdf>.
14. Чубукова И.А. Data Mining БИНОМ. Лаборатория знаний, Интернет-университет информационных технологий - ИНТУИТ.ру, 2006.
15. Дюк В.А., Самойленко А.П. Data Mining: учебный курс. – СПб.: Питер, 2001.
16. Барсегян А. А., Куприянов М. С., Степаненко В. В., Холод И. И. Методы и модели анализа данных: OLAP и Data Mining. – СПб.: БХВ-Петербург, 2004. – 336 с.
17. Дивак М. П., Шпінталь М.Я., Козак О.Л., Струбицька І.П., Спільчук В.М., Піговський Ю.Р. Методичні рекомендації до виконання дипломної роботи освітньо кваліфікаційного рівня «бакалавр» студентам усіх форм навчання для напряму підготовки 6.050103 – «Програмна інженерія» // Тернопіль : ФОП Шпак П.П. - 2013. - 54 с.

ДОДАТОК А

ГЛОСАРІЙ

Основні поняття предметної області	
Термін	Опис терміну
ЖКГ	підприємство, що пропонує послуги пов'язані з комунальними послугами.
Клієнт	користувач системи (комп'ютер, ноутбук)
Користувач	людина, яка є клієнтом ЖКГ


```

using System.Collections.Generic;
using System.Data;
using System.Data.SqlClient;
using System.Linq;
using System.Threading;
using System.Web;
using System.Web.UI;
using System.Web.UI.WebControls;

namespace DiplomV2
{
    public partial class Login : System.Web.UI.Page
    {
        protected void Page_Load(object sender, EventArgs e)
        {

        }

        protected void Button1_Click(object sender, EventArgs e)
        {
            DataView dView =
(DataView)SqlDataSource1.Select(DataSourceSelectArguments.Empty);
            if (TextBox1.Text == "admin" && TextBox2.Text == "123")
            {
                Response.Redirect(HttpContext.Current.Request.Url.GetLeftPart(UriPartial.Authority) +
"/View.aspx");
            }
            else
            {
                if (dView.Count >= 1)
                {
                    Label3.Text = dView[0]["Id"] as string;
                    Label3.Text = "you " + Label3.Text;
                }

                Response.Redirect(HttpContext.Current.Request.Url.GetLeftPart(UriPartial.Authority)+"/User/Us
erMain.aspx?fielid1=" + dView[0]["Id"] as string);
            }
            else
            {
                Label3.Text = "ERROR";
            }
        }
        Label3.Visible = true;
    }
}

```



```
}}
```

Головна сторінка Asp.net:

```
<%@ Page Title="" Language="C#" MasterPageFile="~/jkg.Master" AutoEventWireup="true"
CodeBehind="Main.aspx.cs" Inherits="DiplomV2.WebForm5" %>
<asp:Content ID="Content1" ContentPlaceHolderID="head" runat="server">
  <style type="text/css">
    .auto-style1 {
      width: 582px;
    }
    .auto-style2 {
      width: 254px;
    }
pre
    {margin-bottom: .0001pt;
font-size: 10.0pt;
font-family: "Courier New";
      margin-left: 0cm;
      margin-right: 0cm;
      margin-top: 0cm;
    }
    .auto-style3 {
      width: 254px;
      height: 175px;
    }
    .auto-style4 {
      width: 582px;
      height: 175px;
    }
    .auto-style5 {
      height: 175px;
    }
  </style>
</asp:Content>
<asp:Content ID="Content2" ContentPlaceHolderID="ContentPlaceHolder1" runat="server">
  <div style="background-position: center center; height: 219px; margin-top:
0px; margin: auto">
    <asp:Image ID="Image4" runat="server" Height="219px"
ImageUrl="~/Image/Slogan.gif" Width="843px" />
  </div>
  <div style="margin: 0px auto auto auto; height: 87px; width: 366px;">
    <br />
    <asp:ImageButton ID="ImageButton2" runat="server" ImageUrl="~/Image/R.gif"
OnClick="ImageButton2_Click" />
```

```

&nbsp; &nbsp; &nbsp; &nbsp; &nbsp; &nbsp; &nbsp; &nbsp; &nbsp; &nbsp; &nbsp; &nbsp; &nbsp; &nbsp; &nbsp; &nbsp; &nbsp; &nbsp; &nbsp; &nbsp; &nbsp; &nbsp; &nbsp; &nbsp; &nbsp;
<asp:ImageButton ID="ImageButton3" runat="server" ImageUrl ="~/Image/Vh i d. gi f"
OnCl i ck="ImageButton3_Cl i ck" />
</div>
<br />
&nbsp; &nbsp; &nbsp; &nbsp; &nbsp; &nbsp; &nbsp; &nbsp; &nbsp; &nbsp; &nbsp; &nbsp; &nbsp; &nbsp; &nbsp; &nbsp; &nbsp; &nbsp; &nbsp; &nbsp; &nbsp; &nbsp;
<table style="width: 100%; margin: auto">
    <tr>
        <td class="auto-style2">&nbsp;</td>
        <td class="auto-style1" style="text-align: center; font-size: x-
Large;">Новини</td>
    </tr>
    <tr>
        <td class="auto-style3">
            <asp:Image ID="Image5" runat="server" Height="154px"
ImageUrl ="~/Image/Internet.jpg" Width="234px" />
        </td>
        <td class="auto-style4">
            <pre style="background: white"><span style="font-family:
&quot;Ari al &quot;; sans-seri f; col or: #212121; background: white">Оплатити комунальні : банк
, пошта або Інтернет ?</o:p></o:p></span></pre>
            <pre style="background: white"><span style="font-family:
&quot;Ari al &quot;; sans-seri f; col or: #212121; background: white"><o:p>&nbsp;</o:p></span>
<span style="font-family: &quot;Ari al &quot;; sans-seri f; col or: #212121; background:
white">Оплата комунальних послуг - справа обов&#39;язкова, але не завжди приємна. Особливо
</span></pre>
            <pre style="background: white"><span style="font-family:
&quot;Ari al &quot;; sans-seri f; col or: #212121; background: white">тому що займає чимало часу.
Своєчасна оплата може бути справжньою проблемою для тих, </span></pre>
            <pre style="background: white"><span style="font-family:
&quot;Ari al &quot;; sans-seri f; col or: #212121; background: white">хто живе активно і у кого
розписаний весь день, хто затиснутий вимогами робочого графіка,</span></pre>
            <pre style="background: white"><span style="font-family:
&quot;Ari al &quot;; sans-seri f; col or: #212121; background: white"> хто буває у відрадженнях
або, навпаки, прив&#39;язаний до місця проживання. Для погашення </span></pre>
            <pre style="background: white"><span style="font-family:
&quot;Ari al &quot;; sans-seri f; col or: #212121; background: white">рахунку «по-старому» - в
банку або поштовому відділенні, необхідно виділити кілька годин на </span></pre>
            <pre style="background: white"><span style="font-family:
&quot;Ari al &quot;; sans-seri f; col or: #212121; background: white">одну тільки оплату
комунальних послуг. А більш сучасний підхід - оплата через Інтернет - </span></pre>
            <pre style="background: white"><span style="font-family:
&quot;Ari al &quot;; sans-seri f; col or: #212121; background: white">все ще залишається

```

```

незвичним для багатьох клієнтів. </span><span lang="UK" style="font-
family: &quot; inherit&quot;; serif; color: #212121; mso-ansi-l language:
    UK">Немає прив&#39;язки до конкретних робочих </span></pre>
        <pre style="background: white"><span lang="UK" style="font-
family: &quot; inherit&quot;; serif; color: #212121; mso-ansi-l language:
    UK">годинах каси . Економічно вигідно . Онлайн - сервіси стягують доступні комісії за
вчинення платежів</span><span lang="EN-US" style="font-
family: &quot; inherit&quot;; serif; color: #212121; mso-ansi-l language:
    EN-US">. <o: p></o: p></span></pre>
            </td>
            <td class="auto-style5"></td>
        </tr>
        <tr>
            <td class="auto-style2">
                <asp: Image ID="Image6" runat="server" Height="154px"
ImageUrl ="~/Image/bk.jpg" Width="234px" />
            </td>
            <td class="auto-style1">
                <pre><span style="font-family: &quot; Arial &quot;; sans-serif; color:
#212121; background: white">В Європі кожен п&#39;ятий власник карти Vi sa платить
безконтактно</span><span lang="EN-US" style="font-family: &quot; Arial &quot;; sans-serif;
color: #212121; background: white; mso-ansi-l language: EN-US"><o: p></o: p></span></pre>
                    <pre><span lang="EN-US" style="font-family: &quot; Arial &quot;; sans-
serif; color: #212121; background: white; mso-ansi-l language: EN-
US"><o: p>&nbsp;</o: p></span></pre>
                        <pre><span lang="EN-US" style="mso-ansi-l language: EN-US">
                            </span><span style="font-family: &quot; Arial &quot;; sans-serif; color: #212121;
background: white">Кількість</span><span style="font-family: &quot; Arial &quot;; sans-serif;
color: #212121; background: white; mso-ansi-l language: EN-US"> </span><span style="font-
family: &quot; Arial &quot;; sans-serif; color: #212121; background:
white">безконтактних</span><span style="font-family: &quot; Arial &quot;; sans-serif; color:
#212121; background: white; mso-ansi-l language: EN-US"> </span><span style="font-family:
&quot; Arial &quot;; sans-serif; color: #212121; background: white">платежів</span><span
style="font-family: &quot; Arial &quot;; sans-serif; color: #212121; background: white; mso-
ansi-l language: EN-US"> </span><span style="font-family: &quot; Arial &quot;; sans-serif; color:
#212121; background: white">весь</span><span style="font-family: &quot; Arial &quot;; sans-
serif; color: #212121; background: white; mso-ansi-l language: EN-US"> </span><span
style="font-family: &quot; Arial &quot;; sans-serif; color: #212121; background:
white">час</span><span style="font-family: &quot; Arial &quot;; sans-serif; color: #212121;
background: white; mso-ansi-l language: EN-US"> </span><span style="font-family:
&quot; Arial &quot;; sans-serif; color: #212121; background: white">зростає</span><span
lang="EN-US" style="font-family: &quot; Arial &quot;; sans-serif; color: #212121; background:

```

```

white; mso-ansi-l language: EN-US" > - </span><span style="font-family: &quot;Arial&quot;; sans-
serif; color: #212121; background: white">фіксує</span><span lang="EN-US" style="font-
family: &quot;Arial&quot;; sans-serif; color: #212121; background: white; mso-ansi-l language:
EN-US" > Visa Europe, </span><span style="font-family: &quot;Arial&quot;; sans-serif; color:
#212121; background: white">європейське</span><span style="font-family:
&quot;Arial&quot;; sans-serif; color: #212121; background: white; mso-ansi-l language: EN-US" >
</span></pre>

```

```

<pre><span style="font-family: &quot;Arial&quot;; sans-serif; color:
#212121; background: white">представництво</span><span style="font-family:
&quot;Arial&quot;; sans-serif; color: #212121; background: white; mso-ansi-l language: EN-US" >
</span><span style="font-family: &quot;Arial&quot;; sans-serif; color: #212121; background:
white">однієї</span><span style="font-family: &quot;Arial&quot;; sans-serif; color: #212121;
background: white; mso-ansi-l language: EN-US" > </span><span style="font-family:
&quot;Arial&quot;; sans-serif; color: #212121; background: white">з</span><span style="font-
family: &quot;Arial&quot;; sans-serif; color: #212121; background: white; mso-ansi-l language:
EN-US" > </span><span style="font-family: &quot;Arial&quot;; sans-serif; color: #212121;
background: white">найбільших</span><span style="font-family: &quot;Arial&quot;; sans-serif;
color: #212121; background: white; mso-ansi-l language: EN-US" > </span><span style="font-
family: &quot;Arial&quot;; sans-serif; color: #212121; background:
white">платіжних</span><span style="font-family: &quot;Arial&quot;; sans-serif; color:
#212121; background: white; mso-ansi-l language: EN-US" > </span><span style="font-family:
&quot;Arial&quot;; sans-serif; color: #212121; background: white">систем</span><span
style="font-family: &quot;Arial&quot;; sans-serif; color: #212121; background: white; mso-
ansi-l language: EN-US" > </span><span style="font-family: &quot;Arial&quot;; sans-serif; color:
#212121; background: white">в</span><span style="font-family: &quot;Arial&quot;; sans-serif;
color: #212121; background: white; mso-ansi-l language: EN-US" > </span><span style="font-
family: &quot;Arial&quot;; sans-serif; color: #212121; background: white">світі</span><span
lang="EN-US" style="font-family: &quot;Arial&quot;; sans-serif; color: #212121; background:
white; mso-ansi-l language: EN-US" >. </span><span style="font-family: &quot;Arial&quot;; sans-
serif; color: #212121; background: white">Тільки за рік проведено</span></pre>

```

```

<pre><span style="font-family: &quot;Arial&quot;; sans-serif; color:
#212121; background: white"> близько трьох мільярдів безконтактних транзакцій в
Європейському регіоні. Цей показник</span></pre>

```

```

<pre><span style="font-family: &quot;Arial&quot;; sans-serif; color:
#212121; background: white"> в три рази перевищує аналогічний в 2014 році. У грошовому
еквіваленті безконтактних</span></pre>

```

```

<pre><span style="font-family: &quot;Arial&quot;; sans-serif; color:
#212121; background: white"> платежів здійснено на суму понад 13 мільйонів євро (на 12%
більше, ніж рік тому). </span></pre>

```

```

<pre><span style="font-family: &quot;Arial&quot;; sans-serif; color:
#212121; background: white">Щомісяця кількість безконтактних транзакцій зростає. У квітні
поточного року європейці</span></pre>

```

```

        <pre><span style="font-family: &quot;Arial&quot;; sans-serif; color:
#212121; background: white"> близько 360 мільйонів разів оплачували покупки і послуг
безконтактними картами.</span><span style="font-
family: &quot;inherit&quot;; serif; color: #212121"><o:p></o:p></span></pre>
        <pre><span style="font-
family: &quot;inherit&quot;; serif; color: #212121"><o:p>&nbsp;</o:p></span></pre>
        <pre><o:p></o:p></pre>
        <pre><span style="font-
family: &quot;inherit&quot;; serif; color: #212121"><o:p>&nbsp;</o:p></span></pre>
    </td>
    <td>&nbsp;</td>
</tr>
<tr>
    <td class="auto-style2">
        <asp:Image ID="Image7" runat="server" Height="157px"
ImageUrl="~/Image/Tok.jpg" Width="218px" />
    </td>
    <td class="auto-style1">
        <pre><span lang="UK" style="font-
family: &quot;inherit&quot;; serif; color: #212121; mso-ansi-language: UK">Браслет з зарядом
струму допоможе витратити менше</span><span style="font-
family: &quot;inherit&quot;; serif; color: #212121"><o:p></o:p></span></pre>
        <pre><span lang="EN-US" style="font-family: &quot;inherit&quot;; serif;
color: #212121; mso-ansi-language: EN-US"><o:p>&nbsp;</o:p></span></pre>
        <pre><span lang="UK" style="font-family: &quot;inherit&quot;; serif;
color: #212121; mso-ansi-language: UK">Перш за все, інновації стосуються контролю
кількості витрачених грошей. По-перше, браслет </span></pre>
        <pre><span lang="UK" style="font-family: &quot;inherit&quot;; serif;
color: #212121; mso-ansi-language: UK">здатний впливати на власника - при перевитраті
коштів з банківської карти, чиї платіжні дані</span></pre>
        <pre><span lang="UK" style="font-family: &quot;inherit&quot;; serif;
color: #212121; mso-ansi-language: UK"> завантажені в програмне забезпечення пристрою,
браслет легенько б&#39;є власника струмом.</span></pre>
        <pre><span lang="UK" style="font-family: &quot;inherit&quot;; serif;
color: #212121; mso-ansi-language: UK"> Невеликий безболісний укол, проте, відчутний - і
реально попередить власника картки про те, </span></pre>
        <pre><span lang="UK" style="font-family: &quot;inherit&quot;; serif;
color: #212121; mso-ansi-language: UK">що рахунок вже порожній.</span><span lang="UK"
style="font-family: &quot;inherit&quot;; serif; color: #212121"> </span><span lang="UK"
style="font-family: &quot;inherit&quot;; serif; color: #212121; mso-ansi-language:
UK">По-друге, пристрій Pavlok має всі властивості для управління </span></pre>
        <pre><span lang="UK" style="font-
family: &quot;inherit&quot;; serif; color: #212121; mso-ansi-language:

```

UK">«розумним будинком». Знову ж таки, в зв'язці з керуванням банківським рахунком. Наприклад, </pre>

```
<pre><span lang="UK" style="font-family: "inherit"; serif; color: #212121; mso-ansi-language:
```

UK">можна налаштувати функцію регулювання роботи опалювальної системи, щоб контролювати </pre>

```
<pre><span lang="UK" style="font-family: "inherit"; serif; color: #212121; mso-ansi-language:
```

UK">витрати на опалення. <o:p></o:p></pre>

```
<pre><span lang="EN-US" style="font-family: "inherit"; serif; color: #212121; mso-ansi-language: EN-US"><o:p>&nbsp;</o:p></span></pre>
```

```
</td>
```

```
<td>&nbsp;</td>
```

```
</tr>
```

```
<tr>
```

```
<td class="auto-style2">
```

```
<asp:Image ID="Image8" runat="server" Height="143px" ImageUrl="~/Image/otp.jpg" Width="222px" />
```

```
</td>
```

```
<td class="auto-style1">
```

```
<pre><span lang="UK" style="font-family: "inherit"; serif; color: #212121; mso-ansi-language: UK">Туристи в Японії зможуть робити всі платежі, використовуючи відбитки пальців</span><span style="font-family: "inherit"; serif; color: #212121"><o:p></o:p></span></pre>
```

```
<pre><span style="font-family: "inherit"; serif; color: #212121"><o:p>&nbsp;</o:p></span></pre>
```

```
<pre><span lang="UK" style="font-family: "inherit"; serif; color: #212121; mso-ansi-language: UK">Японія стане першою країною, де туристи будуть реєструвати свої відбитки пальців і потім скрізь</span></pre>
```

```
<pre><span lang="UK" style="font-family: "inherit"; serif; color: #212121; mso-ansi-language: UK"> розплачуватися електронним гаманцем .</span><span lang="UK" style="font-family: "inherit"; serif; color: #212121"></span><span lang="UK" style="font-family: "inherit"; serif; color: #212121; mso-ansi-language:
```

```
UK">Програму оплати за допомогою біометричної системи</span></pre>
```

```
<pre><span lang="UK" style="font-family: "inherit"; serif; color: #212121; mso-ansi-language:
```

```
UK"> розпізнання клієнтів впроваджують на загальнодержавному рівні.</span><span style="font-family: "inherit"; serif; color: #212121"><span style="mso-spacerun: yes">
```

```

</span></span><span lang="UK" style="font-
family: &quot; inherit&quot;; serif; color: #212121; mso-ansi-languag
e:
    UK">При здійсненні оплати людині</span></pre>
    <pre><span lang="UK" style="font-
family: &quot; inherit&quot;; serif; color: #212121; mso-ansi-languag
e:
    UK"> буде досить доторкнутися до спеціального сенсорного екрану , щоб гроші були
зняті з його рахунку . </span></pre>
    <pre><span lang="UK" style="font-
family: &quot; inherit&quot;; serif; color: #212121; mso-ansi-languag
e:
    UK">Чи не буде потрібно навіть використовувати пластикову карту або смартфон
.</span><span lang="EN-US" style="font-family: &quot; inherit&quot;; serif; color: #212121; mso-
ansi-languag
e:
    EN-US"><o:p></o:p></span></pre>
    <pre><span lang="EN-US" style="font-
family: &quot; inherit&quot;; serif; color: #212121; mso-ansi-languag
e: EN-
US"><o:p>&nbsp;</o:p></span></pre>
    </td>
    <td>&nbsp;</td>
</tr>
</table>
<br />
<br />
</asp:Content>

```

Реалізація C#:

```

using System;
using System.Collections.Generic;
using System.Linq;
using System.Web;
using System.Web.UI;
using System.Web.UI.WebControls;

namespace Di pl omV2
{
    public partial class WebForm5 : System.Web.UI . Page
    {
        protected void Page_Load(object sender, EventArgs e)
        {

        }

        protected void Button6_Click(object sender, EventArgs e)
        {

```

```

Response.Redirect(HttpContext.Current.Request.Url.GetLeftPart(UriPartial.Authority) + "/User/Regi ster.aspx");
    }

    protected void Button7_Click(object sender, EventArgs e)
    {
Response.Redirect(HttpContext.Current.Request.Url.GetLeftPart(UriPartial.Authority) +
"/User/Logi n.aspx");
    }

    protected void ImageButton2_Click(object sender, ImageClickEventArgs e)
    {
Response.Redirect(HttpContext.Current.Request.Url.GetLeftPart(UriPartial.Authority) +
"/User/Regi ster.aspx");
    }

    protected void ImageButton3_Click(object sender, ImageClickEventArgs e)
    {
Response.Redirect(HttpContext.Current.Request.Url.GetLeftPart(UriPartial.Authority) +
"/User/Logi n.aspx");
    }
}}

```

Рестрація користувача Asp.net:

```

<%@ Page Title="" Language="C#" MasterPageFile="~/j kg.Master" AutoEventWi reup="true"
CodeBehi nd="Regi ster.aspx.cs" Inheri ts="Di pl omV2.WebForm2" %>
<asp:Content ID="Content1" ContentPl aceHol derID="head" runat="server">
</asp:Content>
<asp:Content ID="Content2" ContentPl aceHol derID="ContentPl aceHol der1" runat="server">

    <div style="width: 50%; height: 50%; position: relative; top: 5px; left:
238px;"><table>
        <tr><td>Ім'я</td><td><asp:TextBox ID="TextBox1"
runat="server"></asp:TextBox></td></tr>
        <tr><td>Логін</td><td><asp:TextBox ID="TextBox2"
runat="server"></asp:TextBox></td></tr>
        <tr><td>Пароль</td><td><asp:TextBox ID="TextBox3" runat="server"
TextMode="Password"></asp:TextBox></td></tr>
        <tr><td>Підтвердіть пароль</td><td><asp:TextBox ID="TextBox7"
runat="server" TextMode="Password"></asp:TextBox></td></tr>

```



```

        <tr><td>Адреса</td><td><asp: TextBox ID="TextBox4"
runat="server"></asp: TextBox></td></tr>
        <tr><td>Номер квартири</td><td><asp: TextBox ID="TextBox5"
runat="server"></asp: TextBox></td></tr>
        <tr><td>@mai l</td><td><asp: TextBox ID="TextBox6"
runat="server"></asp: TextBox></td></tr>
        <tr><td></td><td><asp: Button ID="Button1" runat="server"
Text="Зареєструватись" OnClick="Button1_Click"></asp: Button></td></tr>
        <tr><td colspan="2"> <asp: Label ID="Label 1" runat="server" Text="Реєстрацію
успішно завершено!" Visible="False" ForeColor="Black"></asp: Label ></td></tr>
    </table></div>

```

```
</asp: Content>
```

Реалізація C#:

```

using System;
using System.Collections.Generic;
using System.Data.SqlClient;
using System.Linq;
using System.Web;
using System.Web.UI;
using System.Web.UI.WebControls;

namespace Di pl omV2
{
    public partial class WebForm2 : System.Web.UI . Page
    {
        protected void Page_Load(object sender, EventArgs e)
        {

        }

        protected void Button1_Click(object sender, EventArgs e)
        {
            SqlConnection con = new SqlConnection("Data Source=USER-PC; Initial
Catalog=JKG; Integrated Security=True");
            con.Open();
            SqlCommand cmd = new SqlCommand("insert into Klient(Name, Login,
Password, Adress, KvNum, Mail) values(' " + TextBox1.Text + "', ' " + TextBox2.Text + "', ' " +
TextBox3.Text + "', ' " + TextBox4.Text + "', ' " + int.Parse(TextBox5.Text) + "', ' " +
TextBox6.Text + "' )", con);
            cmd.ExecuteNonQuery();
            con.Close();
            Label 1. Visible = true;
        }
    }
}

```

```

    }
}

```

Реалізація Оплати Asp.net:

```

<%@ Page Title="" Language="C#" MasterPageFile="~/jkg.Master" AutoEventWireup="true"
CodeBehind="Payment.aspx.cs" Inherits="DiplomV2.User.Payment" %>
<asp:Content ID="Content1" ContentPlaceHolderID="head" runat="server">
    <style type="text/css">
        .auto-style1 {
            height: 25px;
        }
        .auto-style2 {
            width: 162px;
        }
        .auto-style3 {
            height: 25px;
            width: 162px;
        }
        .auto-style5 {
            width: 318px;
        }
        .auto-style6 {
            height: 29px;
            width: 318px;
        }
        .auto-style7 {
            height: 29px;
        }
    </style>
</asp:Content>
<asp:Content ID="Content2" ContentPlaceHolderID="ContentPlaceHolder1" runat="server">
    <table style="width: 100%; ">
        <tr>
            <td> <asp:Label ID="Label1" runat="server" Text="Назва
послуги"></asp:Label >
            </td>
            <td>
<asp:Label ID="Label2" runat="server" Text="Показник"></asp:Label >
                и</td>
            <td class="auto-style2">Сумма</td>
            <td class="auto-style2" id="pay" runat="server">Додати платіж</td>
            <td>
<asp:Label ID="Label3" runat="server" Text="Спосіб оплати"></asp:Label >
            </td>

```

```

        </tr>
        <tr>
            <td class="auto-style1">
                <asp: DropDownList ID="n1" runat="server" DataSourceID="SqlDataSource1"
                DataTextFieldID="Name" DataValueFieldID="Name" AppendDataBoundItems="True" AutoPostBack="True"
                OnTextChanged="n1_TextChanged" >
                    <asp: ListItem Text="Оберіть послугу" Value="" />
                </asp: DropDownList>
            </td>
            <td class="auto-style1"><asp: DropDownList ID="pl1" runat="server"
                AutoPostBack="True" DataSourceID="SqlDataSource2" DataTextFieldID="Num" DataValueFieldID="Num"
                Height="16px">
                </asp: DropDownList>
            </td>
            <td class="auto-style3">
                <asp: TextBox ID="s1" runat="server" Enabled="False"
                AutoPostBack="True"></asp: TextBox>
            </td>
            <td class="auto-style1" id="pay2" runat="server">
                <asp: Button ID="Button2" runat="server" Height="23px" Text="+" Width="46px"
                OnClick="Button2_Click" />
            </td>
            <td class="auto-style2"><asp: DropDownList ID="DropDownList17"
                runat="server">
                <asp: ListItem>MasterCard/Visa</asp: ListItem>
                <asp: ListItem>Maestro</asp: ListItem>
            </asp: DropDownList>
            </td>
        </tr>
        <tr>
            <td class="auto-style1"></td>
            <td class="auto-style1">
                <asp: TextBox ID="TextBox1" runat="server" Width="138px"
                OnTextChanged="TextBox1_TextChanged" AutoPostBack="True"></asp: TextBox>
            </td>
            <td class="auto-style3"></td>
            <td class="auto-style3"></td>
            <td class="auto-style3"></td>
        </tr>
    </tr>
    <td>

```

```

        <asp:DropDownList ID="n2" runat="server" DataSourceID="SqlDataSource1"
DataTextFieldID="Name" DataValueFieldID="Name" AutoPostBack="True" AppendDataBoundItems="True"
OnTextChanged="n2_TextChanged" Visible="False">
        <asp:ListItem Text="Оберіть послугу" Value="" />
    </asp:DropDownList>
    </td>
    <td> <asp:DropDownList ID="pl2" runat="server" AutoPostBack="True"
DataSourceID="SqlDataSource3" DataTextFieldID="Num" DataValueFieldID="Num" Visible="False">
    </asp:DropDownList>
    </td>
    <td>
        <asp:TextBox ID="s2" runat="server" Enabled="False"
Visible="False"></asp:TextBox>

    </td>
    <td>
        <asp:Button ID="Button4" runat="server" Height="23px" Text="+" Width="46px"
OnClick="Button4_Click" Visible="False" />
    </td>
    <td> &nbsp; </td>
</tr>
<tr>
    <td> &nbsp; </td>
    <td>
        <asp:TextBox ID="TextBox3" runat="server" AutoPostBack="True"
OnTextChanged="TextBox3_TextChanged" Visible="False"></asp:TextBox>
    </td>
    <td> &nbsp; </td>
    <td> &nbsp; </td>
    <td> &nbsp; </td>
</tr>
<tr>
    <td>
        <asp:DropDownList ID="n3" runat="server" DataSourceID="SqlDataSource1"
DataTextFieldID="Name" DataValueFieldID="Name" AutoPostBack="True" AppendDataBoundItems="True"
OnTextChanged="n3_TextChanged" Visible="False">
        <asp:ListItem Text="Оберіть послугу" Value="" />
    </asp:DropDownList>
    </td>
    <td> <asp:DropDownList ID="pl3" runat="server" AutoPostBack="True"
DataSourceID="SqlDataSource4" DataTextFieldID="Num" DataValueFieldID="Num" Visible="False">
    </asp:DropDownList>

```



```

        <asp:ControlParameter ControlID="n1" Name="PoslugaName"
PropertyName="SelectedValue" Type="String" />
    </SelectParameters>
</asp:SqlDataSource>
    <asp:SqlDataSource ID="SqlDataSource3" runat="server" ConnectionString="<%"$
ConnectionString:JKGConnectionString %>" SelectCommand="SELECT [Num] FROM [Pokaznik] WHERE
(([KlientID] = @KlientID) AND ([PoslugaName] = @PoslugaName))">
    <SelectParameters>
        <asp:ControlParameter ControlID="TextBox12" Name="KlientID"
PropertyName="Text" Type="Int32" />
        <asp:ControlParameter ControlID="n2" Name="PoslugaName"
PropertyName="SelectedValue" Type="String" />
    </SelectParameters>
</asp:SqlDataSource>
    <asp:SqlDataSource ID="SqlDataSource4" runat="server" ConnectionString="<%"$
ConnectionString:JKGConnectionString %>" SelectCommand="SELECT [Num] FROM [Pokaznik] WHERE
(([KlientID] = @KlientID) AND ([PoslugaName] = @PoslugaName))">
    <SelectParameters>
        <asp:ControlParameter ControlID="TextBox12" Name="KlientID"
PropertyName="Text" Type="Int32" />
        <asp:ControlParameter ControlID="n3" Name="PoslugaName"
PropertyName="SelectedValue" Type="String" />
    </SelectParameters>
</asp:SqlDataSource>
<br />
<div runat="server" id="cardboard" Visible="False" style="background-image:
url('../Image/card2.png'); background-repeat: no-repeat; height: 211px; ">
    <table style="position: relative; top: 111px; left: 14px; ">
        <tr>
            <td class="auto-style6">
                <asp:TextBox ID="TextBox16" runat="server" style="margin-left:
0px" Height="16px" Width="285px"></asp:TextBox>
            </td>
            <td class="auto-style7">
                </td>
            <td class="auto-style7"></td>
        </tr>
        <tr>
            <td class="auto-style5">&nbsp;</td>
            <td&nbsp;</td>
            <td>

```



```

    }
}

protected void n1_TextChanged(object sender, EventArgs e)
{
}

protected void TextBox1_TextChanged(object sender, EventArgs e)
{
    SqlConnection con = new SqlConnection("Data Source=USER-PC;Initial
Catalog=JKG;Integrated Security=True");

    DataTable DT = new DataTable();
    con.Open();
    SqlCommand cmd = new SqlCommand("Select Coeficient From Posluga Where
Name = '"+n1.Text+'"', con);
    cmd.ExecuteNonQuery();
    SqlDataAdapter ad = new SqlDataAdapter(cmd);
    ad.Fill(DT);
    ad.AcceptChangesDuringFill = false;

    con.Close();
    DT.TableName = "Posluga";
    foreach (DataRow row in DT.Rows)
    {
        row.AcceptChanges();
    }
    GridView dataview = new GridView();
    dataview.DataSource = DT;

    string coi f = DT.Rows[0].ItemArray[0].ToString();

    float coi ff = float.Parse(coi f);
    float sum = (float.Parse(TextBox1.Text) - float.Parse(pl1.Text));
    sum = sum*coi ff;
    s1.Text = sum.ToString() ;
}

protected void Button2_Click(object sender, EventArgs e)
{
    PayCount=2;
}

```



```

        Button2.Visible = false;
        Button4.Visible = true;
        n2.Visible = true;
        pl2.Visible = true;
        s2.Visible = true;
        TextBox3.Visible = true;
    }

    protected void Button4_Click(object sender, EventArgs e)
    {
        PayCount=3;
        Button4.Visible = false;
        n3.Visible = true;
        pl3.Visible = true;
        s3.Visible = true;
        TextBox13.Visible = true;
        pay.Visible = false;
        pay2.Visible = false;
    }

    protected void n2_TextChanged(object sender, EventArgs e)
    {
    }

    protected void n3_TextChanged(object sender, EventArgs e)
    {
    }

    protected void TextBox3_TextChanged(object sender, EventArgs e)
    {
        SqlConnection con = new SqlConnection("Data Source=USER-PC;Initial
        Catalog=JKG;Integrated Security=True");

        DataTable DT = new DataTable();
        con.Open();
        SqlCommand cmd = new SqlCommand("Select Coefficient From Posluga Where
        Name = '" + n2.Text + "'", con);
        cmd.ExecuteNonQuery();
        SqlDataAdapter ad = new SqlDataAdapter(cmd);
    }

```

```

ad.Fill(DT);
ad.AcceptChangesDuringFill = false;

con.Close();
DT.TableName = "Posluga";
foreach (DataRow row in DT.Rows)
{
    row.AcceptChanges();
}
GridView dataview = new GridView();
dataview.DataSource = DT;

string coef = DT.Rows[0].ItemArray[0].ToString();

float coiff = float.Parse(coef);
float sum = (float.Parse(textBox3.Text) - float.Parse(pl2.Text));
sum = sum * coiff;
s2.Text = sum.ToString();
}

protected void TextBox13_TextChanged(object sender, EventArgs e)
{
    SqlConnection con = new SqlConnection("Data Source=USER-PC; Initial
Catalog=JKG; Integrated Security=True");

    DataTable DT = new DataTable();
    con.Open();
    SqlCommand cmd = new SqlCommand("Select Coefficient From Posluga Where
Name = '" + n3.Text + "'", con);
    cmd.ExecuteNonQuery();
    SqlDataAdapter ad = new SqlDataAdapter(cmd);
    ad.Fill(DT);
    ad.AcceptChangesDuringFill = false;

    con.Close();
    DT.TableName = "Posluga";
    foreach (DataRow row in DT.Rows)
    {
        row.AcceptChanges();
    }
    GridView dataview = new GridView();
    dataview.DataSource = DT;

```

```

string coef = DT.Rows[0].ItemArray[0].ToString();

float coiff = float.Parse(coef);
float sum = (float.Parse(TextBox13.Text) - float.Parse(pl3.Text));
sum = sum * coiff;
s3.Text = sum.ToString();
}

protected void Оплатити_Click(object sender, EventArgs e)
{
    float temp;
    cardboard.Visible = true;
    Button6.Visible = true;
    switch (PayCount)
    {

        case 1:
        {
            temp = float.Parse(s1.Text);
            TextBox14.Text = temp.ToString();
        }
        break;
        case 2:
        {
            temp = float.Parse(s1.Text)+float.Parse(s2.Text);
            TextBox14.Text = temp.ToString();
        }
        break;
        case 3:
        {
            temp =
float.Parse(s1.Text)+float.Parse(s2.Text)+float.Parse(s3.Text);
            TextBox14.Text = temp.ToString();
        }
        break;

    }
}

protected void Button6_Click(object sender, EventArgs e)
{
    if (TextBox15.Text.Length != 0 && TextBox16.Text.Length != 0 &&
    TextBox17.Text.Length != 0)

```

```
    {  
        Label 4. Vi si ble = true;  
    }  
}
```

ДОДАТОК В

DDL-код БД

```
USE [master]
GO
/***** Object: Database [JKG]  Script Date: 27.05.2016 16:07:31 *****/
CREATE DATABASE [JKG]
    CONTAINMENT = NONE
    ON PRIMARY
    ( NAME = N'JKG', FILENAME = N'C:\Program Files\Microsoft SQL
Server\MSSQL12.MSSQLSERVER\MSSQL\DATA\JKG.mdf' , SIZE = 5120KB ,
MAXSIZE = UNLIMITED, FILEGROWTH = 1024KB )
    LOG ON
    ( NAME = N'JKG_log', FILENAME = N'C:\Program Files\Microsoft SQL
Server\MSSQL12.MSSQLSERVER\MSSQL\DATA\JKG_log.ldf' , SIZE = 2048KB
, MAXSIZE = 2048GB , FILEGROWTH = 10%)
GO
ALTER DATABASE [JKG] SET COMPATIBILITY_LEVEL = 120
GO
IF (1 = FULLTEXTSERVICEPROPERTY('IsFullTextInstalled'))
begin
EXEC [JKG].[dbo].[sp_fulltext_database] @action = 'enable'
end
GO
ALTER DATABASE [JKG] SET ANSI_NULL_DEFAULT OFF
GO
ALTER DATABASE [JKG] SET ANSI_NULLS OFF
GO
ALTER DATABASE [JKG] SET ANSI_PADDING OFF
GO
```

```
ALTER DATABASE [JKG] SET ANSI_WARNINGS OFF
GO
ALTER DATABASE [JKG] SET ARITHABORT OFF
GO
ALTER DATABASE [JKG] SET AUTO_CLOSE OFF
GO
ALTER DATABASE [JKG] SET AUTO_SHRINK OFF
GO
ALTER DATABASE [JKG] SET AUTO_UPDATE_STATISTICS ON
GO
ALTER DATABASE [JKG] SET CURSOR_CLOSE_ON_COMMIT OFF
GO
ALTER DATABASE [JKG] SET CURSOR_DEFAULT GLOBAL
GO
ALTER DATABASE [JKG] SET CONCAT_NULL_YIELDS_NULL OFF
GO
ALTER DATABASE [JKG] SET NUMERIC_ROUNDABORT OFF
GO
ALTER DATABASE [JKG] SET QUOTED_IDENTIFIER OFF
GO
ALTER DATABASE [JKG] SET RECURSIVE_TRIGGERS OFF
GO
ALTER DATABASE [JKG] SET DISABLE_BROKER
GO
ALTER DATABASE [JKG] SET AUTO_UPDATE_STATISTICS_ASYNC
OFF
GO
ALTER DATABASE [JKG] SET DATE_CORRELATION_OPTIMIZATION
OFF
GO
```

```
ALTER DATABASE [JKG] SET TRUSTWORTHY OFF
GO
ALTER DATABASE [JKG] SET ALLOW_SNAPSHOT_ISOLATION OFF
GO
ALTER DATABASE [JKG] SET PARAMETERIZATION SIMPLE
GO
ALTER DATABASE [JKG] SET READ_COMMITTED_SNAPSHOT OFF
GO
ALTER DATABASE [JKG] SET HONOR_BROKER_PRIORITY OFF
GO
ALTER DATABASE [JKG] SET RECOVERY SIMPLE
GO
ALTER DATABASE [JKG] SET MULTI_USER
GO
ALTER DATABASE [JKG] SET PAGE_VERIFY CHECKSUM
GO
ALTER DATABASE [JKG] SET DB_CHAINING OFF
GO
ALTER DATABASE [JKG] SET FILESTREAM(
NON_TRANSACTED_ACCESS = OFF )
GO
ALTER DATABASE [JKG] SET TARGET_RECOVERY_TIME = 0
SECONDS
GO
ALTER DATABASE [JKG] SET DELAYED_DURABILITY = DISABLED
GO
/***** Object: Login [User-PC\User]    Script Date: 27.05.2016 16:07:31
*****/
CREATE LOGIN [User-PC\User] FROM WINDOWS WITH
DEFAULT_DATABASE=[master], DEFAULT_LANGUAGE=[русский]
```

GO

/* For security reasons the login is created disabled and with a random password. */

***** Object: Login [sarmkadan] Script Date: 27.05.2016 16:07:31
 *****/

```
CREATE          LOGIN          [sarmkadan]          WITH
PASSWORD=N'b
```

```
åln"S@Û  "9¥Mû-^ DV äkpr"  B',          DEFAULT_DATABASE=[master],
DEFAULT_LANGUAGE=[русский],          CHECK_EXPIRATION=OFF,
CHECK_POLICY=ON
```

GO

```
ALTER LOGIN [sarmkadan] DISABLE
```

GO

***** Object: Login [NT SERVICE\Winmgmt] Script Date: 27.05.2016
 16:07:31 *****/

```
CREATE LOGIN [NT SERVICE\Winmgmt] FROM WINDOWS WITH
DEFAULT_DATABASE=[master], DEFAULT_LANGUAGE=[русский]
```

GO

***** Object: Login [NT SERVICE\SQLWriter] Script Date: 27.05.2016
 16:07:31 *****/

```
CREATE LOGIN [NT SERVICE\SQLWriter] FROM WINDOWS WITH
DEFAULT_DATABASE=[master], DEFAULT_LANGUAGE=[русский]
```

GO

***** Object: Login [NT SERVICE\ReportServer] Script Date:
 27.05.2016 16:07:31 *****/

```
CREATE LOGIN [NT SERVICE\ReportServer] FROM WINDOWS WITH
DEFAULT_DATABASE=[master], DEFAULT_LANGUAGE=[русский]
```

GO


```

/***** Object: Login [NT Service\MSSQLSERVER] Script Date:
27.05.2016 16:07:31 *****/

```

```

CREATE LOGIN [NT Service\MSSQLSERVER] FROM WINDOWS WITH
DEFAULT_DATABASE=[master], DEFAULT_LANGUAGE=[русский]

```

```
GO
```

```

/***** Object: Login [NT AUTHORITY\система] Script Date:
27.05.2016 16:07:31 *****/

```

```

CREATE LOGIN [NT AUTHORITY\система] FROM WINDOWS WITH
DEFAULT_DATABASE=[master], DEFAULT_LANGUAGE=[русский]

```

```
GO
```

```

/* For security reasons the login is created disabled and with a random
password. */

```

```

/***** Object: Login [##MS_PolicyTsqlExecutionLogin##] Script Date:
27.05.2016 16:07:31 *****/

```

```

CREATE LOGIN [##MS_PolicyTsqlExecutionLogin##] WITH
PASSWORD=N'Qü; | ¥-

```

```

,m°Kç ³Ò sÓØ-T%11" $æìÿ', DEFAULT_DATABASE=[master],
DEFAULT_LANGUAGE=[us_english], CHECK_EXPIRATION=OFF,
CHECK_POLICY=ON

```

```
GO
```

```
ALTER LOGIN [##MS_PolicyTsqlExecutionLogin##] DISABLE
```

```
GO
```

```

/* For security reasons the login is created disabled and with a random
password. */

```

```

/***** Object: Login [##MS_PolicyEventProcessingLogin##] Script Date:
27.05.2016 16:07:31 *****/

```

```

CREATE LOGIN [##MS_PolicyEventProcessingLogin##] WITH
PASSWORD=N' ò×Qûè- !

```

```

      È.w÷Oð°îÂL/WâàO`f½êWiÆ',          DEFAULT_DATABASE=[master],
DEFAULT_LANGUAGE=[us_english],           CHECK_EXPIRATION=OFF,
CHECK_POLICY=ON

```

```
GO
```

```
ALTER LOGIN [##MS_PolicyEventProcessingLogin##] DISABLE
```

```
GO
```

```
ALTER SERVER ROLE [sysadmin] ADD MEMBER [User-PC\User]
```

```
GO
```

```
ALTER SERVER ROLE [sysadmin] ADD MEMBER [NT
SERVICE\Winmgmt]
```

```
GO
```

```
ALTER SERVER ROLE [sysadmin] ADD MEMBER [NT
SERVICE\SQLWriter]
```

```
GO
```

```
ALTER SERVER ROLE [sysadmin] ADD MEMBER [NT
Service\MSSQLSERVER]
```

```
GO
```

```
USE [JKG]
```

```
GO
```

```

/***** Object: Table [dbo].[Payment]    Script Date: 27.05.2016 16:07:32
*****/

```

```
*****/
```

```
SET ANSI_NULLS ON
```

```
GO
```

```
SET QUOTED_IDENTIFIER ON
```

```
GO
```

```
CREATE TABLE [dbo].[Payment](
```

```
    [Id] [int] IDENTITY(1,1) NOT NULL,
```

```
    [Data] [nchar](10) NULL,
```

```
    [Sum] [float] NULL,
```

```

        [PoslugaId] [int] NULL,
        [TransactionId] [int] NULL,
    CONSTRAINT [PK_Payment] PRIMARY KEY CLUSTERED
    (
        [Id] ASC
    )WITH (PAD_INDEX = OFF, STATISTICS_NORECOMPUTE = OFF,
    IGNORE_DUP_KEY = OFF, ALLOW_ROW_LOCKS = ON,
    ALLOW_PAGE_LOCKS = ON) ON [PRIMARY]
    ) ON [PRIMARY]

```

GO

```

/***** Object: Table [dbo].[Pokaznik]   Script Date: 27.05.2016 16:07:32
*****/

```

```

SET ANSI_NULLS ON

```

GO

```

SET QUOTED_IDENTIFIER ON

```

GO

```

CREATE TABLE [dbo].[Pokaznik](
    [Id] [int] IDENTITY(1,1) NOT NULL,
    [PoslugaName] [nvarchar](max) NULL,
    [Num] [float] NULL,
    [KlientID] [int] NULL,
    CONSTRAINT [PK_Pokaznik] PRIMARY KEY CLUSTERED
    (
        [Id] ASC
    )WITH (PAD_INDEX = OFF, STATISTICS_NORECOMPUTE = OFF,
    IGNORE_DUP_KEY = OFF, ALLOW_ROW_LOCKS = ON,
    ALLOW_PAGE_LOCKS = ON) ON [PRIMARY]
    ) ON [PRIMARY] TEXTIMAGE_ON [PRIMARY]

```

GO

/***** Object: Table [dbo].[Posluga] Script Date: 27.05.2016 16:07:32

*****/

SET ANSI_NULLS ON

GO

SET QUOTED_IDENTIFIER ON

GO

SET ANSI_PADDING ON

GO

CREATE TABLE [dbo].[Posluga](

[Id] [int] IDENTITY(1,1) NOT NULL,

[Name] [varchar](50) NULL,

[Coificient] [float] NULL,

[NumForPayment] [int] NULL,

CONSTRAINT [PK_Posluga] PRIMARY KEY CLUSTERED

([Id] ASC

)WITH (PAD_INDEX = OFF, STATISTICS_NORECOMPUTE = OFF,

IGNORE_DUP_KEY = OFF, ALLOW_ROW_LOCKS = ON,

ALLOW_PAGE_LOCKS = ON) ON [PRIMARY]

) ON [PRIMARY]

GO

SET ANSI_PADDING OFF

GO

/***** Object: Table [dbo].[Transaction] Script Date: 27.05.2016 16:07:32

*****/

SET ANSI_NULLS ON

GO

SET QUOTED_IDENTIFIER ON

GO

SET ANSI_PADDING ON

```

GO
CREATE TABLE [dbo].[Transaction](
    [Id] [int] IDENTITY(1,1) NOT NULL,
    [PayMenthod] [varchar](50) NULL,
    [KlientId] [int] NULL,
    CONSTRAINT [PK_Transaction] PRIMARY KEY CLUSTERED
    ([Id] ASC
    )WITH (PAD_INDEX = OFF, STATISTICS_NORECOMPUTE = OFF,
IGNORE_DUP_KEY = OFF, ALLOW_ROW_LOCKS = ON,
ALLOW_PAGE_LOCKS = ON) ON [PRIMARY]
    ) ON [PRIMARY]

GO
SET ANSI_PADDING OFF
GO
ALTER TABLE [dbo].[Payment] WITH CHECK ADD CONSTRAINT
[FK_Payment_Posluga] FOREIGN KEY([PoslugaId])
REFERENCES [dbo].[Posluga] ([Id])
GO
ALTER TABLE [dbo].[Payment] CHECK CONSTRAINT
[FK_Payment_Posluga]
GO
ALTER TABLE [dbo].[Payment] WITH CHECK ADD CONSTRAINT
[FK_Payment_Transaction] FOREIGN KEY([TransactionId])
REFERENCES [dbo].[Transaction] ([Id])
GO
ALTER TABLE [dbo].[Payment] CHECK CONSTRAINT
[FK_Payment_Transaction]
GO

```

```
ALTER TABLE [dbo].[Pokaznik] WITH CHECK ADD CONSTRAINT
[FK_Pokaznik_Klient] FOREIGN KEY([KlientID])
REFERENCES [dbo].[Klient] ([Id])
GO
ALTER TABLE [dbo].[Pokaznik] CHECK CONSTRAINT
[FK_Pokaznik_Klient]
GO
ALTER TABLE [dbo].[Transaction] WITH CHECK ADD CONSTRAINT
[FK_Transaction_Klient] FOREIGN KEY([KlientId])
REFERENCES [dbo].[Klient] ([Id])
GO
ALTER TABLE [dbo].[Transaction] CHECK CONSTRAINT
[FK_Transaction_Klient]
GO
USE [master]
GO
ALTER DATABASE [JKG] SET READ_WRITE
GO
```