

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

## КВАЛІФІКАЦІЙНА РОБОТА БАКАЛАВРА

на тему «АВТОМАТИЗАЦІЯ ПРОЦЕСУ УПРАВЛІННЯ  
ПРЕЙСКУРАНТАМИ ДЛЯ СИСТЕМ ОБЛІКУ ТА БІЛІНГУ»

Виконав студент 4 курсу, групи ЕК-61а

(номер курсу)

(шифр групи)

Спеціальності 051 «Економіка»

(Економічна кібернетика)

Нечепоренко І.Д.

(прізвище, ініціали студента)

Керівник професор, д.е.н., Олійник В.М.

(посада, науковий ступінь, прізвище, ініціали)

Суми – 2020 рік

## РЕФЕРАТ

кваліфікаційної роботи бакалавра на тему  
«АВТОМАТИЗАЦІЯ ПРОЦЕСУ УПРАВЛІННЯ ПРЕЙСКУРАНТАМИ ДЛЯ  
СИСТЕМ ОБЛІКУ ТА БІЛІНГУ»  
студента Нечепоренка Іллі Дмитровича

Актуальність теми, обраної для дослідження, визначається тим, що кожне підприємство прагне отримати максимальний прибуток і мінімізувати витрати. Тому автоматизація роботи підприємства допоможе зменшити фінансові витрати та витрати у часі.

Мета кваліфікаційної роботи полягає у розробці прототипу web-додатку для автоматизації процесу управління прейскурантами для систем обліку та білінгу.

Об'єктом дослідження є діяльність підприємства «Nicetec GmbH» та його фірмовий продукт “Netinsight” для IT-фінансів та управління спільним центром обслуговування.

Предметом дослідження є процес роботи та обліку прейскурантів та цін.

Задачі, які виконувались для досягнення мети роботи:

- дослідження суті поставленої задачі та предметної області
- дослідження діяльності IT-підприємства, загальна характеристика організації;
- дослідження роботи основного продукту компанії під назвою “Netinsight”;
- дослідження існуючої інформаційної системи для автоматизації процесу управління прейскурантами;
- формулювання основних вимог та обмежень до web-додатку ;
- розробка архітектури та вибір технологій створення прототипу;
- безпосередня розробка прототипу додатку з урахуванням усіх

обмежень та вимог накладених на нього;

- тестування прототипу.

Для досягнення поставленої мети та задач дослідження були використані такі методи дослідження: аналітичний; експериментальний.

Інформаційною базою кваліфікаційної роботи є діяльність підприємства «Nicetec GmbH» та функціонал основного продукту підприємства.

Основний науковий результат кваліфікаційної роботи полягає у створенні прототипу автоматизованої системи, що містить у собі всі необхідні функціональні модулі та ергономічний інтерфейс.

Одержані результати можуть бути використані на середніх та великих підприємствах.

Результати апробації основних положень кваліфікаційної роботи розглядалися на німецькому підприємстві «Nicetec GmbH», яке займається ІТ-менеджментом, розробкою ПЗ та консалтингом.

Ключові слова: автоматизація, прайс-лист, ціна, web-додаток, база даних, ASP.Net Core 3, C#, DevExpress, DevExtreme, Java Script.

Зміст кваліфікаційної роботи викладено на 38 сторінках. Список використаних джерел із 41 найменувань, розміщений на 5 сторінках. Робота містить 15 рисунків, а також 5 додатків, розміщених на 17 сторінках.

Рік виконання кваліфікаційної роботи – 2020 рік.

Рік захисту роботи – 2020 рік.

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

ЗАТВЕРДЖУЮ  
Завідувач кафедри  
д.е.н., професор  
\_\_\_\_\_ О.В. Кузьменко  
“\_\_” \_\_\_\_\_ 2020 р.

ЗАВДАННЯ НА КВАЛІФІКАЦІЙНУ РОБОТУ БАКАЛАВРА  
спеціальність 051 «Економіка (Економічна кібернетика)  
студенту 4 курсу, групи ЕК-61а

Нечепоренко Іллі Дмитровичу

(прізвище, ім'я, по батькові студента)

1. Тема роботи Автоматизація процесу управління преїскурантами для систем обліку та білінгу

затверджена наказом по університету від «\_\_» \_\_\_\_\_ 2020 року № \_\_\_\_\_

2. Термін подання студентом закінченої роботи «17» червня 2020 року

3. Мета кваліфікаційної роботи розробка прототипу web-додатку для автоматизації процесу управління преїскурантами для систем обліку та білінгу

4. Об'єкт дослідження діяльність підприємства «Nicaret GmbH» та його фірмовий продукт “Netinsight” для ІТ-фінансів та управління спільним центром обслуговування

5. Предмет дослідження процес роботи та обліку преїскурантів та цін

6. Кваліфікаційна робота виконується на матеріалах «Nicaret GmbH»

7. Орієнтовний план кваліфікаційної роботи, терміни подання розділів керівникові та зміст завдань для виконання поставленої мети

Розділ 1 Дослідження стану автоматизації бізнес-процесів і формування вимог до інформаційної системи

11 травня 2020 року

(назва – термін подання)

У розділі 1 дати загальну характеристику ІТ-компанії, проаналізувати стан автоматизації бізнес-процесів, сформувані вимоги до інформаційної системи компанії

(зміст конкретних завдань до розділу, які має виконати студент)

## Розділ 2 Реалізація прототипу автоматизованої інформаційної системи

9 червня 2020 року

(назва – термін подання)

У розділі 2 зробити опис основних бізнес-процесів компанії, представити архітектуру веб-орієнтованої інформаційної, розглянути основні технології створення та склад функціональної частини web-додатку, створити контрольний приклад та інструкцію по використанню

(зміст конкретних завдань до розділу, які повинен виконати студент)

### 8. Консультації з роботи:

Розділ	Прізвище, ініціали та посада консультанта	Підпис, дата	
		завдання видав	завдання прийняв
1			
2			
3			

### 9. Дата видачі завдання: «02» березня 2020 року

Керівник кваліфікаційної роботи

\_\_\_\_\_

( підпис)

В.М. Олійник

(ініціали, прізвище)

Завдання до виконання одержав

\_\_\_\_\_

(підпис)

І.Д. Нечепоренко

(ініціали, прізвище)

## ЗМІСТ

ВСТУП.....	3
РОЗДІЛ 1 ДОСЛІДЖЕННЯ СТАНУ АВТОМАТИЗАЦІЇ .....	
БІЗНЕС-ПРОЦЕСІВ І ФОРМУВАННЯ ВИМОГ ДО .....	
ІНФОРМАЦІЙНОЇ СИСТЕМИ.....	5
1.1 Загальна характеристика ІТ-компанії Nicetec GmbH.....	5
1.2 Аналіз стану автоматизації бізнес-процесів.....	7
1.3 Формування вимог до інформаційної системи .....	
підприємства.....	10
1.4 Архітектура інформаційної системи та технології .....	
вирішення поставлених задач.....	17
РОЗДІЛ 2 РЕАЛІЗАЦІЯ ПРОТОТИПУ .....	
АВТОМАТИЗОВАНОЇ ІНФОРМАЦІЙНОЇ СИСТЕМИ .....	25
2.1 Структура та особливості реалізації інформаційного .....	
забезпечення .....	25
2.2 Процес управління преїскурантами та цінами.....	32
2.3 Контрольний приклад та інструкція по використанню .....	35
ВИСНОВКИ .....	39
СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ.....	41
ДОДАТКИ .....	46

## ВСТУП

Управління сучасним підприємством – це доволі складне завдання, яке потребує організації взаємодії ресурсів різного роду. До таких ресурсів можна віднести, інформаційні системи, що забезпечують автоматизацію бізнес-процесів підприємства. Організація результативної системи функціонування IT-інфраструктури є одним із головних аспектів забезпечення заданого рівня якості IT-підприємства.

Будь-яка модель бізнесу заснована на впровадженні нововведень на практиці. Оперативна діяльність, яка визначає можливості її організаційного розвитку. Потреба в сучасних інформаційних технологіях (IT) актуалізується за ступенем розвитку компанії, коли стає важко ефективно контролювати економічну діяльність виключно людськими ресурсами, тому необхідне рішення – це пошук та реалізація нових технологічних рішень, які можуть покращити якість управління.

Завдяки швидкому розвитку технологій ми отримали можливість користуватися мережею Інтернет не тільки для пошуку інформації, але й використовувати її у роботі підприємства. Саме цією можливістю скористалось підприємство Nicetec GmbH при перенесенні функціоналу комп'ютерного додатку у web версію використовуючи сучасні інформаційні технології (IT).

Актуальність цього питання полягає у тому, що інвестиції у розвиток IT-структури – це показник, що компанія зосереджена для зростання та розвитку і конкурентоспроможності на ринку.

Метою дослідження даної роботи є розробка прототипу web-додатку для автоматизації процесу управління преїскурантами для систем обліку та білінгу.

Об'єктом дослідження є діяльність підприємства «Nicetec GmbH» та його фірмовий продукт “Netinsight” для IT-фінансів та управління спільним

центром обслуговування.

Предметом дослідження є процес роботи та обліку преїскурантів та цін.

Завданням дослідження є:

- дослідження суті поставленої задачі та предметної області;
- дослідження діяльності ІТ – підприємства, загальна характеристика організації;
- дослідження роботи основного продукту компанії під назвою “Netinsight”;
- дослідження існуючої інформаційної системи для автоматизації процесу управління преїскурантами;
- формулювання основних вимог та обмежень до web-додатку ;
- розробка архітектури та вибір технологій створення прототипу;
- безпосередня розробка прототипу додатку з урахуванням усіх обмежень та вимог накладених на нього;
- тестування прототипу.



## РОЗДІЛ 1 ДОСЛІДЖЕННЯ СТАНУ АВТОМАТИЗАЦІЇ БІЗНЕС-ПРОЦЕСІВ І ФОРМУВАННЯ ВИМОГ ДО ІНФОРМАЦІЙНОЇ СИСТЕМИ

### 1.1 Загальна характеристика ІТ-компанії Nicetec GmbH

Підприємство займається розробкою високоспеціалізованого програмного забезпечення та є консалтинговим партнером з управління ІТ-фінансами та спільними рішеннями щодо управління послугами, які підходять під вимоги клієнта. Надає змогу співпрацювати з ІТ-контролером через повне управління фінансами до глобальних послуг, які надаються та обліковуються в центрі спільного обслуговування. З часу заснування компанії в 1998 році компанія розробляє програмне забезпечення для планування, розрахунку цін, розподілу витрат та контролю витрат на ІТ - з постійною метою досягнення покращеної прозорості, гнучкості та надійності клієнтів. З 1999 року компанія спеціалізується на ІТ-контролінгу, управлінні фінансами ІТ та спільним управлінням сервісом та надає такі послуги:

- прозорість для ІТ витрат і послуг (як повернути контроль і впевнено вплинути на свої витрати);
- розрахунок послуги (як забезпечити прозорість для кожного кроку вашого розрахунку ціни);
- управління контрактом / угодами за угодою SLA (як керувати своїми контрактами з усіма їхніми послугами, термінами та умовами в централізованому режимі);
- управління каталогом послуг (як скласти каталог послуг таким чином, що орієнтований на бізнес-процеси та клієнтів);
- розподіл витрат (як виставляти рахунки за послуги вартісно, легко, зрозуміло та пов'язано з використанням);
- звітність та аналіз (як доставити звіти вищої якості більш

економічно, швидше та простіше).

Спільне управління сервісом за допомогою мережевої інформації. Спільні послуги являють собою сучасну форму підходу до обслуговування. Метою є об'єднання підтримуючих функцій перерізу компанії в окремих підрозділах та досягнення високої якості обслуговування за економічних витрат на обслуговування шляхом спеціалізації та оптимізації.

В цьому відіграють роль внутрішні фактори, такі як проектування послуг, планування та рахунок, а також розробка відповідних моделей обліку. Окрім високої якості обслуговування, клієнти також очікують професійних структур та високого рівня прозорості стосовно витрат на центр обслуговування. Спільні центри обслуговування часто конкурують з іншими постачальниками послуг на ринку. Якщо відповідальна сторона центру спільного обслуговування стикається з цими викликами та проблемами то Nicetec розробив нове рішення у вигляді SSC netinsight, яке приймає на себе особливі завдання в управлінні спільними послугами - просто, гнучко та економічно. Netinsight SSC - це центральне та інтегроване рішення управління спільним сервісом, яке поєднує наявну інформацію та дані з різних функцій, спеціалізованих областей та підсистем у загальне рішення та виконує щоденні функції, такі як:

- моделювання спільних серій / управління життєвим циклом обслуговування;
- планування та розрахунок цін на послуги;
- контроль витрат;
- процеси планування та управління попитом для ваших клієнтів;
- управління контрактами та угодами за угодою SLA для клієнтів та постачальником;
- автоматичне оцінювання та облік послуг;
- закупівля самообслуговування та обслуговування ваших клієнтів через Інтернет;

- аналізи та звітність для управління та клієнтів SSC.

Клієнтами є великі корпорації, страхові компанії, банки та великі міжнародні компанії середнього розміру, які надають послуги між окремими компаніями групи і змушені взаємо замінювати їх відповідно до положень законодавства та податкового законодавства. Компанія підтримує клієнтів широким спектром послуг і сильних партнерів [1].

Офіс компанії “Nicetec GmbH” знаходиться поруч із містом Оснабрюк на заході Німеччини. Штат компанії налічує близько 15 співробітників з різних країн, таких як: Іран, Росія, Албанія, Австрія.

## 1.2 Аналіз стану автоматизації бізнес-процесів

На першому етапі аналізу стану автоматизації бізнес-процесів ІТ-компанії було досліджено характер та структуру бізнес-процесів у типових існуючих організаціях даного напрямку діяльності.

Основною типовою фігурою в бізнес-процесах і у процесах прийняття рішень виступає генеральний директор, власник організації. Він керує усіма видами діяльності, координує роботу всіх підрозділів організації, контролює всі бізнес-процеси і приймає рішення щодо їх вдосконалення.

Ralf Meyer - головний виконавчий директор (CEO), заснував компанію у 1998 році. До обов'язків головного виконавчого директора входить:

- визначати загальну стратегію підприємства;
- приймати загальні рішення на вищому рівні, які забезпечуватимуть стійке зростання компанії;
- розподіляти грошові потоки згідно пріоритетів компанії;
- виконувати представницькі обов'язки.
- створювати і розширяти клієнтську базу;
- проводити переговори, організовувати презентації, укладати договори.

Для складання фінансової звітності агентство не наймає штатних співробітників, а використовує послуги аутсорсингу, оскільки обсяг існуючої роботи в цій галузі є недостатнім для збереження штатних співробітників. Компанія на довгостроковій основі користується послугами позаштатного бухгалтера, який займається фінансовою діяльністю організації, веде бухгалтерський облік і готує квартальні звіти про діяльність організації. Процес прийняття управлінських рішень керівництвом організації є цілеспрямованим і централізованим.

Існуючі бізнес-процеси організації включають низку функціональних підсистем:

- стратегічне та поточне управління (організація);
- планування;
- HR;
- управління оперативною діяльністю;
- фінансовий менеджмент;
- управління інвестиціями;
- управління маркетингом.

Функціональна підсистема поточного та стратегічного управління спрямована на постійне економічне зростання та підвищення конкурентоспроможності компанії. У рамках цієї підсистеми керівник організації постійно здійснює стратегічний аналіз змін у зовнішньому середовищі (кон'юнктура ринку, суспільно-політичне середовище, законодавча система тощо). Керівник організації розробляє стратегію розвитку та здійснює контроль за виконанням. Основна частина роботи над цією функціональною підсистемою проводиться керівником організації; на виході підсистеми формується стратегічний план розвитку, який має декларативний характер і є обов'язковим для ознайомлення з усіма працівниками організації. Поточне управління забезпечує безперервність та синхронізацію діяльності всіх функціональних підрозділів, спрямованих на

виконання поточних завдань агентства [2-3].

Основний відділ – це програмісти, розробники, які займаються створенням нового програмного забезпечення та виконують найбільший обсяг роботи у створенні нових продуктів. Також є бізнес-аналітики, тестувальники, системний адміністратор. Особливої уваги потребує відділ по відносинам з клієнтами, який працює з кожним клієнтом, не тільки від початку до продажу продукту, але й супроводжує та надає допомогу протягом усього часу подальшого використання клієнтом продукту компанії.

Для ведення операційної діяльності компанія користується сучасними засобами зв'язку та передачі інформації, такими як електронна пошта Outlook, IP-телефонія, Microsoft Teams – центр командної роботи в Microsoft 365, Azure DevOps.

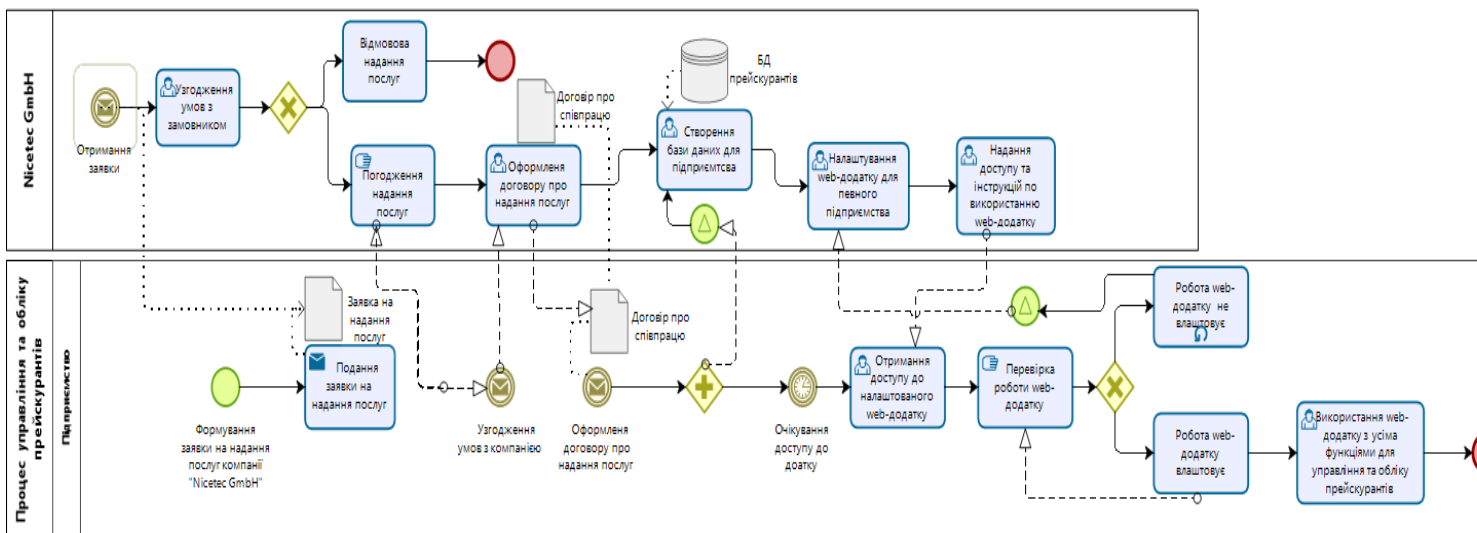
Для чіткого розуміння бізнес-процесів підприємства та для уникнення непорозумінь при розробці web-додатку для автоматизації процесу управління преїскурантами для систем обліку та білінгу було прийнято рішення побудувати BPMN діаграму бізнес-процесів (рис 1.1).

Business Process Modeling Notation (BPMN) - це метод ілюстрації бізнес-процесів у вигляді діаграми, аналогічної схемі потоків. BPMN спочатку була задумана та розроблена Ініціативою управління бізнес-процесами (BPMI). В даний час його підтримує Група управління об'єктами (OMG).

BPMN забезпечує стандартний, легкий для читання спосіб визначення та аналізу державних та приватних бізнес-процесів. BPMN пропонує стандартні позначення, які легко зрозумілі керівництву, аналітикам та розробникам. Первісний намір BPMN полягав у тому, щоб допомогти усунути прогалини в комунікаціях, які часто існують між різними відділами в організації чи підприємстві. BPMN може також допомогти забезпечити візуалізацію документів XML (розширюваної мови розмітки), призначених для виконання різноманітних бізнес-процесів, за допомогою загальної нотації.

Діаграма в BPMN зібрана з невеликого набору основних елементів, що полегшує технічним та нетехнічним спостерігачам зрозуміти пов'язані

процеси. Елементи класифікуються на три основні групи, що називаються об'єктами течії, що з'єднують об'єкти та плавки. Об'єкти потоку, позначені геометричними фігурами, такими як кола, прямокутники та ромби, вказують на конкретні події та дії. Об'єкти потоку пов'язані між собою сполучними об'єктами, які виглядають як суцільні, пунктирні або пунктирні лінії, які можуть містити стрілки для позначення напрямку процесу. Плавці, названі так через свою геометричну схожість з лініями смуги на дні олімпійського басейну, позначаються як суцільні прямі лінії, що проходять уздовж прямокутника, що називається басейном. Плавці організують різноманітні об'єкти потоку в категорії, що мають подібну функціональність [4-5].



Риснок 1.1 – BPMN діаграма бізнес-процесів

### 1.3 Формування вимог до інформаційної системи підприємства

Інформаційна система – організований набір елементів, використовуваних для зберігання, обробки та видачі інформації в інтересах досягнення поставленої мети. Інформаційна система складається із людей, обладнання, процесів, процедур, даних та операцій. Кожна інформаційна система складається з наступних компонентів:

- структура системи;
- функції кожного елемента системи;

- вхід і вихід кожного елемента і системи в цілому;
- мета і обмеження системи та її окремих елементів.

Інформаційна система не тільки показує функціонування об'єкта управління, а й впливає на нього через органи керування. Вона є сукупністю інформаційних процесів для задоволення потреби в інформації різних рівнів прийняття рішень[6].

До інформаційної системи, яка використовувалась в процесі роботи можна віднести web-додаток «NetInsight SSC».

Netinsight SSC - продукт, який показує потоки вартості ІТ у компанії, тим самим дозволяє цільове управління, а також виставлення рахунків на основі послуг та використання послуг, що надаються. Netinsight SSC дозволяє керувати своїми витратами так само легко, як і в будь-якій іншій області компанії.

Для створення web-додатку для електронного документообігу потрібно, щоб виконувалися певні принципи та вимоги. Згідно з основних принципів електронного документообігу, web-додаток повинен діяти на таких умовах:

- одноразова реєстрація документа;
- паралельне виконання різних операцій для скорочення часу руху документів та підвищення ефективності їх виконання. Дозволяє вам точно ідентифікувати документ у будь-якій установці цієї системи;
- безперервність руху документів - єдина база документальної інформації для централізованого зберігання документів і виключає копіювання документів. Дозволяє визначити особу, відповідальну за виконання документа (завдання), у будь-який час життя документа (процесу);
- єдина (або узгоджена розподілена) база документальної інформації, яка дає можливість не дублювати документи [7-8].

Ефективно організована система пошуку документів, надсилання та передача електронних документів (ЕД), дозволяє знайти документ з мінімальною інформацією про нього. Впровадження принципів електронного

документообігу на основі останніх інформаційних технологій за допомогою сучасних апаратних та програмних засобів дозволить створити єдиний інформаційний простір електронного документообігу, об'єднуючи в інформаційний вузол всі системи передачі та прийому ЕД. Інтеграція відбувається без зниження якості роботи з документами при збереженні форм традиційного діловодства. Основою такої інтеграції є надійне зберігання ЕД та взаємодіючої системи передачі, прийому та реєстрації електронної документації.

Всі ЕД повинні зберігатися в єдиному сховищі, що забезпечить оптимальний пошук та відбір інформації під час підготовки матеріалів. Такий архів ЕД не пов'язаний безпосередньо з електронним документообігом як таким, але робота з архівом є важливою компонентою пошуку. Інтеграція такого архіву та пов'язаного з ним системи електронних документів в єдиний інформаційний простір підприємства дозволить забезпечити доступ до матеріалів та їх обробку більш ефективним та простим.

Електронні технології створення, зберігання та передачі документів, мають декілька переваг перед паперовими документами. Найважливіші з них:

- швидкий пошук документів;
- простота відстеження руху документів на всіх етапах їх "життєвого циклу";
- зменшення строків підготовки та узгодження документів;
- полегшення організації документообігу, якщо декілька користувачів одночасно працюють з однаковими документами;
- можливість швидкого отримання консолідованих звітів та журналів;
- ефективність передачі документів необхідному адресату.

Перевага електронного документообігу перед традиційним є незаперечним, адже головна проблема традиційної технології управління документами - це практична неможливість централізованого відстеження руху



документів організації.

Аналіз вимог полягає у тому, щоб визначити умови та потреби, котрі висуваються для нового, або зміненого товару, враховуючи потенційно суперечливі вимоги різних клієнтів, таких як користувачі чи бенефіціари. Аналіз вимог має вирішальне значення для успішної розробки проекту. Вимоги повинні бути задокументовані, вимірювані, пов'язані з тестами, пов'язані з бізнесом та описуватися з рівнем деталізації, достатнього для проектування системи. Розробляючи програмне забезпечення, він підпадає під вимоги моделі FURPS+. Вперше класифікація вимог до системи FURPS + була запропонована в 1992 році Робертом Грейді, співробітником компанії Hewlett-Packard. Аббревіатура FURPS означає:

- функціональність, функціональність;
- працездатність, простота використання;
- надійність, надійність;
- продуктивність, продуктивність;
- підтримка, підтримується;
- також необхідно пам'ятати про такі імовірні обмеження, як:
  - а) обмеження проектування, проектування;
  - б) обмеження на розробку, реалізацію;
  - в) обмеження на інтерфейси, інтерфейс;
  - г) фізичні обмеження, фізичні.

Якщо застосувати до цієї класифікації популярний поділ вимог на функціональні та нефункціональні, вони повинні включати всі перераховані вище групи, крім першої, тобто URPS +.

Функціональні вимоги:

1. журнал роботи (auditing)- засоби для відстеження дій користувачів і системи шляхом запису в журнал безпеки певних типів подій;
2. ліцензування (licensing) - засіб для відстеження, установки, придбання, і контролю використання ліцензій;

3. локалізація (localization) - це переклад і культурна адаптація продукту до особливостей певної країни, регіону або групи населення;
4. пошта (mail) - сервіси для відправки та отримання повідомлень.
5. довідка (online help) - можливість надання підтримки користувачів в режимі реального часу;
6. друк (printing) - засіб для друку документів;
7. звітність (reporting) - інструменти для виготовлення і отримання звітів;
8. безпека (security) - засіб захисту доступу до певних інформаційних ресурсів;
9. управління системою (system management) - інструменти, що дають можливість управляти програмами в розподіленому середовищі;
10. технологічний процес (workflow) - підтримка документообігу, підтримуючи процеси перевірки, узгодження та затвердження.

Простота використання.

Простота використання містить наступні типи вимог:

- естетика і логіка призначеного для користувача інтерфейсу;
- захист від людського фактора;
- використовувана документація, її склад (керівництва, адміністратори і т. д.), галузь і стан. стандарти дизайну;
- кваліфікація користувачів і їх навчання;
- додаткова інформація в системі.

Надійність включає в себе такі характеристики системи, як:

1. відмови;
2. допустима частота / частота відмов;
3. середній час відмов і їх серйозність;
4. можливість відновлення системи після збоїв, в тому числі можливість попереднього резервного копіювання даних;
5. передбачуваність поведінки;

6. час роботи системи, режим роботи або час доступності системи (наприклад, «система повинна бути доступна 24 години на добу, 7 днів на тиждень»);

7. точність розрахунків.

Продуктивність. Продуктивність системи складається з наступних характеристик:

- швидкість, час відгуку системи;
- ефективність / ефективність;
- пропускна здатність, включаючи загальну і допустиму кількість одночасних користувачів, кількість призначених для користувача запитів, кількість звернень системи до бази даних і кількість запитаних / переданих даних за одиницю часу;

- час, необхідний для відновлення - швидкість відновлення;
- час, необхідний для запуску та вимикання - швидкість запуску і виключення;

- витрата ресурсів.

Обслуговування системи. Підтримка включає в себе можливості:

- тестування;
- розширення - збільшення додаткової функціональності системи;
- масштабування - тиражування, наприклад, в філіях / підрозділах організації;

- адаптація / адаптація до використання в даному середовищі, в тому числі шляхом попередньої настройки;

- сумісність;
- технічне обслуговування, супровід: оновлення даних та виправлення помилок, періодичність архівування та резервного копіювання;

- сервіс і ремонт;

- встановлення;

- локалізація;

- портативність;
- відповідність міжнародним стандартам.

Уже згадувана класифікація та обмеження відрізняє наступні 4 групи обмежень:

- обмеження конструкції;
- обмеження на реалізацію, розробку, конструювання, написання програмного коду;
- вимоги до інтерфейсів - обмеження (наприклад, на формати, протоколи) накладаються завдяки необхідності взаємодії з іншими системами;
- фізичні обмеження, що накладаються на технічні (апаратні) засоби.

При розробці системи можуть бути накладені інші обмеження, зокрема правові:

- міжнародні угоди: одиниці виміру, мови;
- авторське право;
- угоди про отримання ліцензій;
- законодавство;
- галузеві стандарти.

Основним завданням розробки web-додатку є:

- створення системи для автоматизації процесу управління преїскурантами для систем обліку та білінгу;
- створення системи, яка дозволить зменшити витрати на адміністрування і підтримку клієнтів;
- збільшити кількість користувачів додатку;
- позбавитися складнощів з оновленням і підтримкою викликаних обмеженістю прав користувачів;
- надання швидкого доступу до інформації без попередньої установки;
- використання додатку в хмарних інфраструктурах;

- надання користувачу можливості користуватися додатком незалежно від операційної системи та її версії [9-10].

Вхідною інформацією для інформаційної системи буде база даних PostgreSQL, яка містить усю інформацію стосовно управління і обліку преїскурантами.

Основними користувачами web-додатку будуть компанії, які придбають додаток «NetInsight SSC» для керування фінансами на підприємстві. Це можуть бути компанії як середніх, так і великих розмірів.

Основною функціональною частиною інформаційної системи буде можливість додавання, редагування, дуплікація, перегляд та видалення преїскурантів. Для цього потрібно створити таблиці, які будуть відображати актуальну інформацію про преїскуранти, беручи всю необхідну інформацію із бази даних. Також потрібно буде зробити перехід до сторінки на якій буде зберігатися інформація про ціни ті їх розрахунок, властиві преїскурантам.

#### 1.4 Архітектура інформаційної системи та технології вирішення поставлених задач

Поняття інформаційної архітектури стало відомим задовго до терміна UX і, взагалі, до появи перших сайтів та мобільних додатків, але саме завдяки розвитку Internet стало асоціюватися із інформаційними технологіями. На даному етапі розвитку IT під інформаційною архітектурою розуміють, як науку про організацію і структурування контенту в різних інформаційних системах, наприклад, веб-сайтах, мобільних додатках, яка дозволяє розкрити шлях вирішення певного завдання або пошуку інформації із будь-якої частини сервісу [11].

Щоб побудувати інформаційну архітектуру потрібно:

- ознайомитися з комп'ютерним додатком, на основі якого буде розроблюватися web-версія;

- ознайомитися з базою даних;
- розробити макет для web-сторінок;
- оцінити, чи підходить створений дизайн до загального проекту додатку.

Для розробки web-додатку було обрано клієнт-серверну архітектуру.

Клієнт-серверну архітектуру можна означити, як концепцію інформаційної мережі в якій основна частина її ресурсів зосереджена в серверах, обслуговуючих своїх клієнтів.

Архітектура клієнт-сервер – це концепція інформаційної мережі, в якій основна частина її ресурсів зосереджена в серверах, обслуговуючих своїх клієнтів. Архітектура клієнт-сервер є домінуючою концепцією у створенні розподілених мережних застосувань і передбачає взаємодію та обмін даними між ними. Вона передбачає такі головні компоненти:

- клієнтська частина веб-додатків, яка зображує графічний інтерфейс. Графічний інтерфейс веб-додатків виводиться на екранні засоби браузерів. Користувач взаємодіє з веб-орієнтованою інформаційною системою через браузер, оперує елементами веб-сторінки (меню, кнопки, посилання, поля вводу та ін.);

- серверна частина представляє собою програму, яка знаходиться на серверах. Серверна частина веб-орієнтованих інформаційних систем може програмувати на різних мовних програмах, найчастіше розповсюджуваних серед таких програм, як мови PHP, ASP.NET, Java, Ruby, Scala, Python, JavaScript, Perl. На серверній частині веб-орієнтованої інформаційної системи відбувається обробка запиту користувача [12-13];

- система управління базами даних (СУБД) представляє собою спеціалізоване програмне забезпечення, передбачене для збору, зберігання та надання інформаційних системних даних. Серверна частина веб-додатків із підтримкою спеціальних скриптів відома з СУБД, створює конкретні запити та отримує дані з баз даних, які в подальшому дозволяють створити програмні

скрипти і при необхідності виводити в користувацький інтерфейс в браузері користувачів. Найбільш популярними СУБД для веб-орієнтованих інформаційних систем є Oracle, MySQL, MS SQL, PostgreSQL, MongoDB, Firestore [14-15] (рис. 1.2). Сервери є незалежними один від одного. Користувачі також функціонують паралельно і незалежно один від одного. Крім того, користувач не прив'язаний до одного конкретного серверу. Така ситуація в наш час, коли один сервер одночасно обробляє запити від різних клієнтів; з іншого боку, клієнт може звертатися до різних серверів одночасно. Клієнти мають знати про доступні сервери, але не уявляти собі наявності інших клієнтів [16].

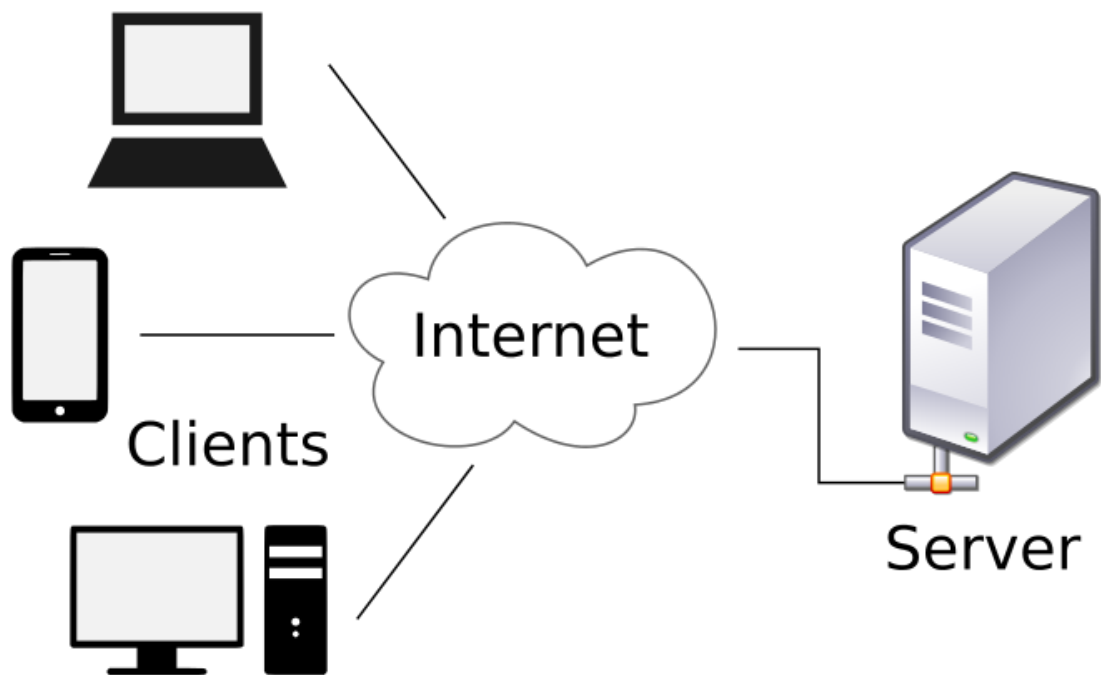


Рисунок 1.2 – Схема клієнт-серверної архітектури

Після ознайомлення з можливостями створення web-додатків і ознайомлення з аналогічними сайтами було сформовано уявлення того, якою має бути інформаційна система додатку для ІТ-компанії.

З метою упорядкування всіх веб-елементів та для вдалого досягнення завдань інформаційної системи було розроблено макет web-додатку. Макет був створений за допомогою програми для розробки інтерфейсів Adobe XD

(рис. 1.3). Аналогічним чином була розроблена сторінка, для управління і обліку цін та впливаючі вікна для додавання, редагування, видалення, дуплікації прейскурантів та цін.

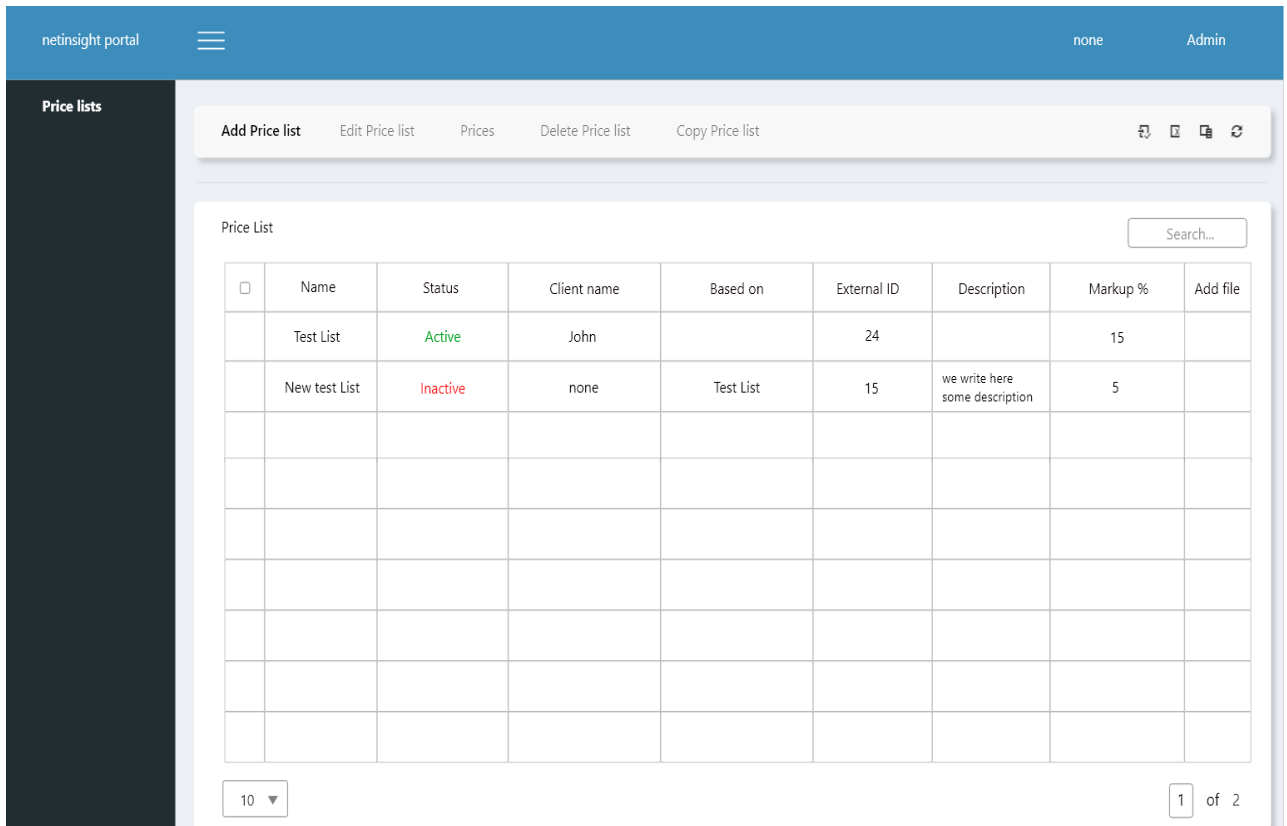


Рисунок 1.3 – Макет інтерфейсу сторінки прейскурантів

Важливість інформаційного забезпечення процесу настільки важлива, що багато експертів виділяють особливу інформаційну логістику, яка має своє значення в бізнесі та управлінні інформаційними потоками та ресурсами. Інформаційний потік - це сукупність повідомлень, що циркулює в межах логістичної системи або між логістичною системою та зовнішнім середовищем і призначена для управління діяльністю на підприємстві[17].

Компанія “Nicetec GmbH” використовує Azure DevOps для регулювання інформаційних потоків всередині підприємства між співробітниками, як одного, так і різних відділів.

DevOps дозволяє представникам раніше розрізнених підрозділів - розробки, IT-операцій, забезпечення якості та безпеки - координувати свої дії і спільно створювати більш якісні та надійні продукти. Впровадивши культуру



DevOps разом з відповідними методиками та інструментами, команди отримують можливість краще реагувати на потреби клієнтів, підвищувати довіру до створюваних додатків і швидше досягати бізнес-цілей[18-19].

За допомогою DevOps співробітники компанії мають доступ до коду та середі розробки, можуть одночасно працювати над проектом, вносити зміни, та бачити інші зміни, оновлюючи проект у реальному часі.

Також DevOps дає можливість заносити свою поточну діяльність та майбутні плани у так звані «Azure Boards», що дозволяє ,не тільки керівникам контролювати робочий процес співробітників, а також іншим колегам розуміти на якій стадії знаходиться проект.

Основними програмними засобами, що використовувалися при розробці web-додатку були Microsoft Visual Studio 2019, PostgreSQL, AzureDevOps, DBeaver та були використані такі технології та мови програмування як: HTML5, C#, Java Script, Asp.NET Core 3, CSS, Ajax, DevExtreme.

Microsoft Visual Studio – це лінійка програмних компонентів, призначений для полегшення написання коду для різних мов. Продукт дозволяє розробляти консольні і графічні додатки в тому числі з підтримкою технології Windows Forms, а також веб-сайти, веб-застосунки, веб-служби як в рідному, так і в керованому кодах для всіх платформ, що підтримуються. Актуальною на поточний момент є версія Visual Studio 2019, випускається вона в наступних редакціях:

1. community - безкоштовна версія середовища розробки Visual Studio;
2. professional - редакція містить професійні інструменти для розробки додатків;
3. enterprise - повнофункціональна версія Visual Studio. Комплексне рішення для розробки додатків. [20-21].

Завдяки величезній кількості налаштувань, підтримуваних технологій, швидкодії і зручності Visual Studio вважається однією з кращих середовищ

розробки.

PostgreSQL - це об'єктно-реляційна система управління базами даних (ORDBMS), заснована на POSTGRES, розроблена в Каліфорнійському університеті при кафедрі комп'ютерних наук Берклі. POSTGRES запровадив багато концепцій, які стали доступними лише в деяких системах комерційних баз даних значно пізніше.

PostgreSQL є нащадком цього оригінального коду Берклі з відкритим кодом. Він підтримує значну частину стандарту SQL і пропонує безліч сучасних функцій:

- складні запити;
- іноземні ключі;
- тригери;
- оновлення переглядів;
- транзакційна цілісність;
- мультиверсійний контроль одночасності.

Також PostgreSQL може бути розширений користувачем багатьма способами, наприклад, додаючи нові:

- типи даних;
- функції;
- оператори;
- сукупність функцій;
- індексні методи;
- процедурні мови[22].

DBeaver - це безкоштовний, відкритий код, графічний інструмент управління базами даних для розробників та адміністраторів баз даних.

Ви можете використовувати DBeaver для створення та управління базами даних у широкому діапазоні систем управління базами даних (СУБД). DBeaver працює з більшістю популярних СУБД, такими як MySQL, PostgreSQL, MariaDB, SQLite, Oracle, DB2, SQL Server, Sybase, Microsoft

Access, Teradata, Firebird, Derby тощо. Фактично, DBeaver може бути використаний у будь-якій базі даних, яка має драйвер JDBC або ODBC. Він також працює на деяких інших базах даних, у яких немає стандартного драйвера xDBC, таких як Mongo DB, Redis та WMI.

DBeaver має багато типових функцій, яких ви очікуєте від інструменту управління базами даних на основі GUI. Наприклад, є Навігатор баз даних, який дозволяє переходити по списку баз даних і розширювати їх для навігації по об'єктах бази даних у межах кожної бази даних. Є також редактор SQL з підсвічуванням синтаксису та автозаповненням. Список основних функцій DBeaver:

- графічний інтерфейс користувача (GUI). Дозволяє переглядати список баз даних та їх об'єкти та вибирати параметри з меню, "вказуючи та клацнувши". Це на відміну від інструменту командного рядка, де вам потрібно запам'ятати точну команду для відображення списку баз даних або виконання інших завдань;
- редагування даних. Надає можливість редагувати дані безпосередньо в таблиці (як і ви можете редагувати дані в електронній таблиці);
- аналіз баз даних. Створіть візуальні діаграми об'єктів бази даних та цілі схеми та експортуйте за потребою. Також включає управління метаданими;
- редактор SQL Дозволяє створювати adhoc SQL запити та запускати їх проти бази даних. Ви також можете завантажувати та зберігати сценарії за потребою. Деякі популярні функцій редактора SQL включають підсвічування синтаксису та автозаповнення;
- особливості бази даних. DBeaver підтримує багато функцій, характерних для певних СУБД. Наприклад, DBeaver підтримує перегляди DDL та зберігаються процедури на SQL Server, типи даних enum / set в MySQL, XML, типи даних типу Cursor в Oracle і DB2 тощо;

– теми / зовнішній вигляд. DBeaver поставляється з деякими темами за замовчуванням. Ви можете перемикатися між ними протягом декількох секунд. Ви також можете змінювати кольори на дуже деталізованому рівні (наприклад, колір будь-яких параметрів, коли вони написані в редакторі SQL)[23].

Платформа Asp.NET – це платформа для розробників, що складається з інструментів, мов програмування та бібліотек для створення багатьох різних типів додатків.

Базова платформа надає компоненти, які застосовуються до всіх різних типів додатків. Додаткові рамки, такі як ASP.NET, розширюють .NET за допомогою компонентів для створення конкретних типів додатків.

ASP.NET розширює платформу .NET з інструментами та бібліотеками спеціально для створення веб-додатків. Ось деякі речі, які ASP.NET додає до платформи .NET:

- база для обробки веб-запитів в C # або F #;
- синтаксис шаблонів веб-сторінок, відомий як Razor, для створення динамічних веб-сторінок за допомогою C #;
- бібліотеки загальних веб-шаблонів, таких як контролер перегляду моделі (MVC);
- система автентифікації, що включає бібліотеки, базу даних та сторінки шаблонів для обробки входів, включаючи багатфакторну автентифікацію та зовнішню автентифікацію за допомогою Google, Twitter тощо;
- розширення редактора, щоб забезпечити виділення синтаксису, доповнення коду та інші функціональні можливості спеціально для розробки веб-сторінок [24-25].

## РОЗДІЛ 2 РЕАЛІЗАЦІЯ ПРОТОТИПУ АВТОМАТИЗОВАНОЇ ІНФОРМАЦІЙНОЇ СИСТЕМИ

### 2.1 Структура та особливості реалізації інформаційного забезпечення

Організація інформаційного забезпечення автоматизованих інформаційних систем має дуже важливе значення, так як процес управління не може бути забезпеченим без перетворення інформаційних потоків. Систему інформаційного забезпечення автоматизованих інформаційних систем необхідно проектувати, враховуючи різноманітні фактори і передбачаючи створення єдиного інформаційного фонду, класифікацію та уніфікацію показників і документів, розробку засобів формалізованого опису даних тощо. Ці особливості описують складність і поліфакторність розробки та структурної побудови інформаційного забезпечення дівлячись від розв'язуваних завдань, загальної структури економічної системи, форм подання даних, складу функцій управління та засобів трансформації інформації.

Для функціональних підсистем АІС інформаційне забезпечення є сукупністю інформаційних ресурсів і найважливішим елементом автоматизованої інформаційної системи, оскільки воно являється забезпечувальною підсистемою при розв'язанні завдань управління і наповнює їх конкретним змістом.

Інформаційне забезпечення автоматизованої інформаційної системи призначене для поліпшення якості управління господарською діяльністю на основі отримання імовірних, підтверджених та своєчасних даних, які необхідні для прийняття управлінських рішень [26].

Найважливішою підсистемою інформаційного забезпечення web-додатку є створення сітки даних (Data Grid) для відображення актуальної інформації про існуючі преїскуранти та ціни. DataGrid - віджет, який

представляє дані з локального або віддаленого джерела у вигляді сітки. Цей віджет пропонує такі основні функції, як сортування, групування, фільтрування, а також більш розширені можливості, такі як державне зберігання, експорт на стороні клієнта, інтерфейс основної деталізації та багато інших [27-28].

Для побудови сітки даних було використано технологію DevExtreme. DevExtreme включає вичерпну колекцію високоефективних та чуйних віджетів інтерфейсу користувача для використання в традиційних мобільних додатках Інтернету та нового покоління. Набір постачається з повною функцією сіткою даних, інтерактивними віджетами діаграм, редакторами даних та багато іншого. DevExtreme ASP.NET MVC Data Grid - це керований серверним управлінням jQuery з великою різноманітністю функцій, включаючи прив'язку до даних з контролерів Web API та MVC, групування записів, фільтрацію та сортування, вибір рядків та багато іншого. Data Grid підтримує сильно набрані HTML-помічники, які надають йому перевірку на стороні клієнта на основі Анотацій даних за замовчуванням. Ви можете вибрати між синтаксисом Razor C # та Razor VB та використовувати лямбда-вирази під час його налаштування [29].

Дані для сітки було отримано із бази даних, яка мала 8 зв'язаних таблиць для преїскурантів (рис. 2.1) та 11 для цін.

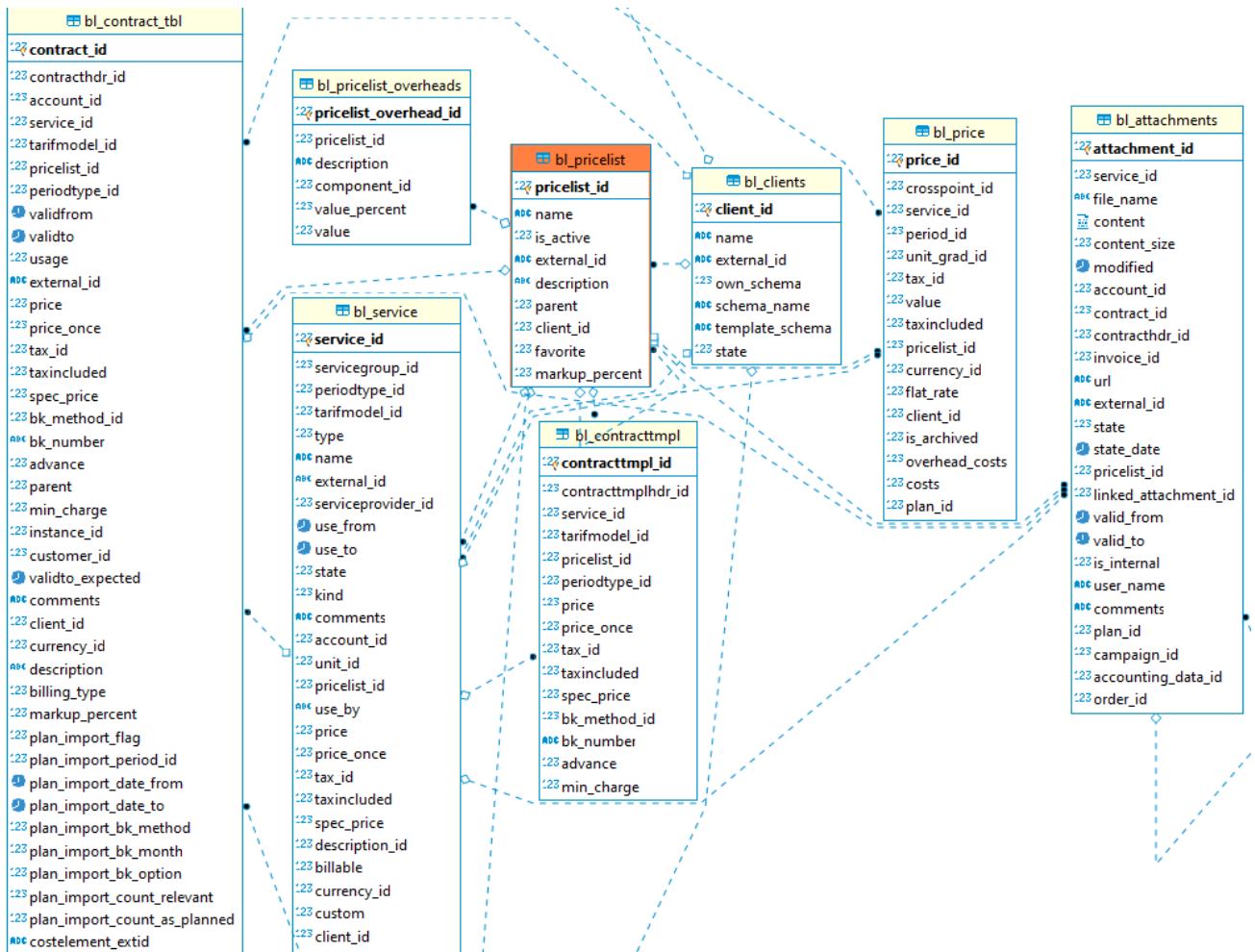


Рисунок 2.1 – ER Diagram для прейскурантів

До цих таблиць будуть додаватися нові дані, або будуть видалятися непотрібні дані під час роботи користувача із прейскурантами та цінами.

Переглянемо таблиці для управління та обліку прейскурантів.

В таблиці «bl\_pricelists» представлені стовбці, які відображають актуальну інформацію про прейскурант (рис. 2.2) . Дана таблиця має такі поля:

- ID прейскуранта ( pricelist\_id);
- Назва прейскуранта (Name);
- Активний/ Неактивний (is\_active);
- Зовнішній ID (external\_id);
- Опис (description);
- Наявність батьківського прейскуранту (parent);

- ID клієнта, користувача, залежно від цього надається рівень доступу до даних (client\_id);
- Обраний (favorite);
- Націнка у відсотках (markup\_percent).

Column Name	#	Data type	Length	Precision	Scale	Identity	Collation	Not Null	Default	Comment
123 pricelist_id	1	numeric		11				<input checked="" type="checkbox"/>		Pricelist unique identifier
ABC name	2	varchar	100	100			default	<input checked="" type="checkbox"/>		Price list name
123 is_active	3	numeric		1				<input checked="" type="checkbox"/>	1	Price list is active flag
ABC external_id	4	varchar	20	20			default	<input type="checkbox"/>		External Identifier for linking to an external CRM system (e.g. SAP/R3)
ABC description	5	varchar	255	255			default	<input type="checkbox"/>		Description to the price list
123 parent	6	numeric		11				<input type="checkbox"/>		Reference to the parent price list in case the price is based on another one
123 client_id	7	numeric		11				<input type="checkbox"/>		Client identifier
123 favorite	8	numeric		1				<input type="checkbox"/>	0	
123 markup_percent	9	numeric		10	5			<input type="checkbox"/>		Markup percent

Рисунок 2.2 – Структура таблиці «bl\_pricelists»

В таблиці «bl\_price» представлені стовбці, які відображають актуальну інформацію про ціни, які включають у себе преїскуранти (рис. 2.3) . Дана таблиця має такі поля:

- ID цін(price\_id);
- Кінцева точка тарифної моделі(crosspoint\_id);
- Сервіси (service\_id);
- Час, протягом якого діє ціна(period\_id);
- Градація цін(unit\_grad\_id);
- Податок(tax\_id);
- Вартість (value);
- Податок доданий у вартість (taxincluded);
- ID преїскуранту (pricelist\_id);
- Посилання до валюти (currency\_id);
- Фіксована ставка (flat\_rate);
- ID клієнту(client\_id);



- Чи є ціна за архівованою (is\_archived);
- Націнка (overhead\_costs);
- Розрахована вартість (costs);
- Зв'язок із планом (plan\_id).

Column Name	#	Data type	Length	Precision	Scale	Identity	Collation	Not Null	Default	Comment
123 price_id	1	numeric		11				<input checked="" type="checkbox"/>		Unique Identifier
123 crosspoint_id	2	numeric		11				<input checked="" type="checkbox"/>		The end point of tariff model
123 service_id	3	numeric		11				<input checked="" type="checkbox"/>	0	Service
123 period_id	4	numeric		11				<input checked="" type="checkbox"/>		The price is valid from date beginning of this period. The price is valid only for contracts with sam...
123 unit_grad_id	5	numeric		11				<input checked="" type="checkbox"/>		Unit gradation for the price (e.g. \$1/Kb or \$1/Mb)
123 tax_id	6	numeric		11				<input checked="" type="checkbox"/>		Tax for the price
123 value	7	numeric		22	6			<input type="checkbox"/>		Price value in the defined currency
123 taxincluded	8	numeric		5				<input checked="" type="checkbox"/>	0	Indicates that tax is included into the price
123 pricelist_id	9	numeric		11				<input checked="" type="checkbox"/>	0	Reference to the price list to which the price belongs to
123 currency_id	10	numeric		11				<input type="checkbox"/>		Reference to the currency in which price is defined
123 flat_rate	11	numeric		1				<input checked="" type="checkbox"/>	0	Indicates so-called flat rate. The price doesn't depend on the amount
123 client_id	12	numeric		11				<input type="checkbox"/>		Client identifier
123 is_archived	13	numeric		1				<input checked="" type="checkbox"/>	0	The record is archived flag
123 overhead_costs	14	numeric		22	6			<input type="checkbox"/>		Sum of base surcharges (overhead to the cost calculated in IT FM)
123 costs	15	numeric		22	6			<input type="checkbox"/>		Calculated cost of the service (normally calculated and filled by IT FM)
123 plan_id	16	numeric		11				<input type="checkbox"/>		Relation to the source plan

Рисунок 2.3 – Структура таблиці «bl\_price»

Таблиця «bl\_pricelist\_overheads» зберігає дані про накладні витрати, а саме : знижки або доплату та про цінові компоненти (рис. 2.4). Містить такі поля:

- ID накладних витрат (pricelist\_overhead\_id);
- ID преїскуранта (pricelist\_id);
- Опис (description);
- ID цінових компонентів (component\_id);
- Націнка або знижка у процентах (value\_percent);
- Націнка або знижка у грошовому еквіваленті (value).

Column Name	#	Data type	Length	Precision	Scale	Identity	Collation	Not Null	Default	Comment
123 pricelist_overhead_id	1	numeric		11				<input checked="" type="checkbox"/>		Unique identifier
123 pricelist_id	2	numeric		11				<input checked="" type="checkbox"/>		Price list identifier
ABC description	3	varchar	100	100			default	<input type="checkbox"/>		The description or a name of the surcharge/discount
123 component_id	4	numeric		11				<input type="checkbox"/>		Price component identifier. If is a null, this is a common surcharge/discount, otherwise a surchar...
123 value_percent	5	numeric		20	5			<input type="checkbox"/>		Surcharge/discount in percents (1.0 matches 100%)
123 value	6	numeric		20	5			<input type="checkbox"/>		Surcharge/discount in absolute amount

Рисунок 2.4 – Структура таблиці «bl\_pricelist\_overheads»

Реалізація процесів функціонування web-додатку здійснюється на сервері окрім JS, але під час виконання запитів AJAX, який здійснюється на мові JS, процес здійснюється саме на сервері.

Авторизація користувачів та надання їм різного рівня доступу є важливим завданням, яке реалізується на сервері. Виконуючи спробу перейти на головну сторінку порталу користувач повинен заповнити форму авторизації з логіном і паролем. Після цього він повинен вибрати у правому верхньому куту сторінки, яким клієнтом користувач є і отримати певні права доступу до преїскурантів та цін, які відносяться тільки для даного клієнта (рис. 2.5).

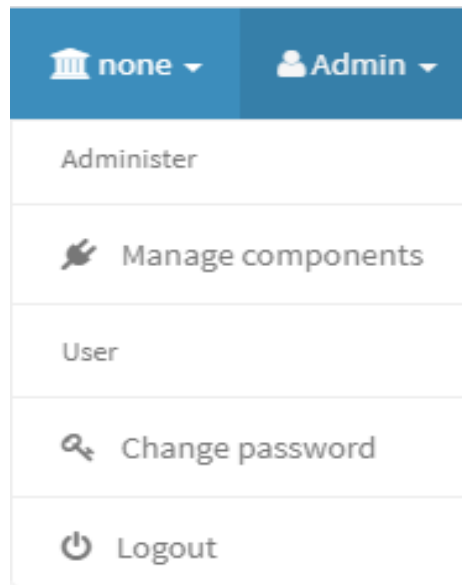


Рисунок 2.5 – Авторизація користувачів

На головній сторінці потрібно відобразити сітку даних з інформацією про преїскуранти, а також кнопки для додавання, редагування, дуплікації, видалення, переходу до цін та функціонал для сортування та фільтрації даних (рис. 2.6, 2.7).

netinsight portal

Price Lists

+ Add Price List Edit Price List Delete Price List Duplicate Price List Show prices

Price List	Active	Client Name	Based On	External ID	Description	Markup %
Q	(All)	Q	(All)	Q	Q	Q
Default	Yes					
New price list	No	none	Default	id	descr	6
Musterpreisliste	Yes	Test		2	sfdgdfg	5
tesst	No	none	Musterpreisliste			
ChildSave	No	Test		sd sdf	price list	
TestChild	Yes	Test	ChildSave	external	abcde	5
New child list	No	Test	TestChild	Ext id	CPI	3
New my list	Yes	Test	TestChild	Ext id	CPI	3
Test LIL Ya	Yes	Test	ChildSave	Extrnl	Testing evrthng	30
tesst	No	Test				
Default Kopie	Yes	none				
Bernds List	Yes	none	Default Kopie			
tesst-Kopie	Yes	none				
test	Yes	Test	tesst-Kopie	84	ook	
Mobile phone service	No	Test				
hugos preis	Yes	none				

20

Activate Windows < 1 of 1 >  
Go to Settings to activate Windows.

Рисунок 2.6 – Головна сторінка

+ Add Price List Edit Price List Delete Price List Duplicate Price List Show prices

Price List	Active	Client Name	Based On
Q	(All)	Q	(All)
Default	Yes		
New price list	No	none	Default
Musterpreisliste	Yes	Test	
tesst	No	none	Musterpreisliste
ChildSave	No	Test	
TestChild	Yes	Test	ChildSave

Рисунок 2.7 – Сітка даних та основні кнопки

Скориставшись функціоналом DevExtreme було створено фільтрацію та сортування для даних, а також пошук серед даних таблиці і розділення даних на певну кількість записів ( 20, 50, 100). У верхньому правому куті можна побачити налаштування: які дані будуть відображатися; відмінити всю

фільтрацію і сортування; відмінити вибір рядка; зробити пошук (рис. 2.8).

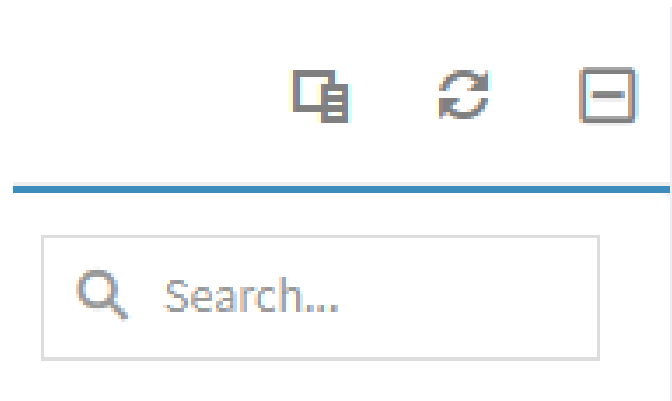


Рисунок 2.8 – Налаштування для фільтрації і сортування

## 2.2 Процес управління прейскурантами та цінами

Управління прейскурантами є одним з найбільш важливих аспектів будь-якого бізнесу в сфері електронної комерції. Користувач може створювати агреговані прайс-листи з будь-якою кількістю цінових рівнів або валют.

В цілому, управління цінами дозволяє:

- встановити гнучкі ціни на продукти для різних веб-сайтів, груп клієнтів та клієнтів;
- призначати ціни на нові товари автоматично;
- створювати графік тимчасових або постійних змін цін.

В якості ілюстрації залежності ціни товару від конкретного прайс-листа наведемо приклад продукту А з початковою ціною 100 доларів. Ця сума відображена у прайс-листі за замовчуванням, який є прямим і загальним для оптовиків, роздрібних продавців, гостей та інших користувачів.

Після того, як був підписаний контракт з клієнтом Х на умовах надання їм 20% знижки на товари з певних категорій на строк дії договору, ви можете потім створити прайс-лист, який перераховує ціни для обраних категорій продуктів на основі узгодженої знижки і відобразити відповідні ціни у таблицях с даними для всіх користувачів, зареєстрованих під клієнтом Х, до

закінчення терміну дії договору.

Також є можливість мати окремий прейскурант для групи клієнтів оптовиків, який відображає ціни зі знижкою 10%. Всі оптові покупці (наприклад, компанії B, C, D), що належать до цієї групи, можуть придбати продукт A за 90 доларів. Впроваджуючи програми лояльності і надаючи додаткові знижки своїм оптовим клієнтам вони можуть придбати один і той же товар за різними цінами, відповідно до знижки.

Можливо мати один або кілька прейскурантів з цінами, введеними в доларах США і євро, що визначають ціни для всіх ваших клієнтів, інший прайс-лист з цінами в доларах США, доступний тільки для великих американських дистриб'юторів в США, і інший прейскурантів з цінами в євро доступні тільки деяким обраним європейським партнерам вашої компанії [30].

У даному проекті управління здійснюється під час створення або редагування цін (рис. 2.9). На вкладці «General» користувач може обирати:

- назву прейскуранта;
- сервіси пов'язані з даним прейскурантом (рис. 2.10);
- тарифну модель (назву тарифу) та самі умови тарифу (Crosspoint);
- період існування ціни( тижневий, місячний, годовий) (рис. 2.11);
- задати конкретний період для ціни (дату).

The screenshot shows a web form titled "Add new Price" with a close button (x) in the top right corner. The form has three tabs: "General" (selected), "Cost", and "Properties". The "General" tab contains the following fields:

- Price Lists:** A dropdown menu with a grid icon.
- Service Name:** A dropdown menu with "none" selected.
- Tariff name:** A dropdown menu with "Select..." selected.
- Crosspoint:** A dropdown menu with "Crosspoint" selected.
- Period Type:** A dropdown menu with "none (once)" selected.
- Valid from:** A date input field with "1/1/1970" entered.
- Price type:** Two radio buttons: "Price per unit" (selected) and "Flat rate".
- Base Price:** An input field and a dropdown menu with "Select..." selected.
- End Price:** A dashed input field.
- Unit:** A dropdown menu with "-" selected.
- Tax:** A dropdown menu with "none" selected.
- Tax included:** A checkbox that is currently unchecked.

Рисунок 2.9 – Створення цін для прейскурантів

none

Service Name	External ID
🔍	🔍
Bildschirm	
Büro	
Grafikkarte	
GSC Service 1	
none	
PC	
Prozessor	
Roberts Service	
Stuhl	
Tagessatz	

Рисунок 2.10 – Сервіси певного прейскуранту

Weekly

- none (once)
- Monthly
- Yearly
- Weekly

Рисунок 2.11 – Періоди існування ціни

Для ціноутворення користувачу надається два варіанти на вибір, ціна за певну одиницю товару, або фіксована ціна. Одиниці виміру товарів підвантажуються в залежності від обраного сервісу (Service Name).

Далі користувач у випадяючому списку може обрати валюту

розрахунку. Також можливо додати податок з наданих стандартних варіантів, та через «Check Box» відмітити, чи включати даний податок у загальну вартість, чи ні.

Кінцева вартість розраховується за формулою:

$$\text{End Price} = \text{Base Price} + \text{Sum}(\text{Price Components} * \text{Price list Components}).$$

На вкладці «Costs» можна відображаються витрати та додана вартість у відсотках, або у грошовому еквіваленті, які враховуються у розрахунку кінцевої вартості (рис 2.12).

The screenshot shows a software interface with three tabs: "General", "Cost", and "Properties". The "Cost" tab is active. Below the tabs are two input fields: "Base Costs" and "Base Overhead". Below these is a section titled "Components" with three icons: a plus sign, a copy icon, and a trash icon. A table lists the components with columns for "Price component", "%", "Price", and "Cost".

<input type="checkbox"/>	Price component	%	Price	Cost
	Preiskomponente 2	36	0.36	0
	PK1	20	0.2	0

Рисунок 2.12 – Цінові витрати та додана вартість

### 2.3 Контрольний приклад та інструкція по використанню

Після розробки web-додатку потрібно зробити контрольне тестування всього функціоналу. Контрольне тестування системи - це рівень тестування, який підтверджує повний і повністю інтегрований програмний продукт. Мета тесту на систему - оцінити технічні характеристики системи. Зазвичай програмне забезпечення є лише одним елементом більшої комп'ютерної системи. Зрештою, програмне забезпечення взаємодіє з іншими програмними / апаратними системами. Тестування системи - це фактично серія різних тестів, єдиною метою яких є здійснення повноцінної комп'ютерної системи [31-33].

Після розробки web-додатку для обліку і управління преїскурантами його було розміщено на веб-портал підприємства до інших модулів додатку.

Для користування додатком користувачу не потрібно завантажувати та встановлювати програмне забезпечення. Користувачу необхідно створити

власний обліковий запис для входу у додаток та мати пристрій для входу в мережу інтернет із встановленим браузером та інтернет з'єднання.

Інтерфейс web-додатку «NetInsight SSC» є дружнім до користувача, зручним, зрозумілим та виконаний у мінімалістичному стилі. Користувач має можливість переглянути існуючі прейскуранти, або перейти до перегляду цін натиснувши кнопку «Show Prices», попередньо вибравши прейскурант зі списку.

Для роботи із прейскурантами та цінами користувачу потрібно скористатися кнопками над основною сіткою даних.

Для додавання прейскуранту потрібно натиснути «Add Price List», відкриється спливаюче вікно (рис. 2.9).

Add new Price list

General Surcharge / Discount Extra Properties Attachments

Name

Based On Select...

Active No

External ID

Description

Cancel Save

Рисунок 2.9 – Додавання нового прейскуранту

Користувачу надається можливість переходити на різні вкладки та вносити дані до основної вкладки (General), додавати націнку або знижку (Discount/Surcharge), вносити додаткові властивості (Extra Properties) та



прикріпляти файли (Attachments). Для додавання націнок і знижки було створено дві сітки даних і кнопки додати та видалити. Редагування робиться методом «Batch» після подвійного кліку на компонент (рис. 2.10).

The screenshot shows a window titled "Add new Price list" with a close button (X) in the top right corner. Below the title bar are four tabs: "General", "Surcharge / Discount", "Extra Properties", and "Attachments". The "Surcharge / Discount" tab is selected. The main area contains two data grids. The left grid is titled "Component" and has a header with "Price component" and "Percent Value". The right grid is titled "General" and has a header with "Description", "Fixed amount", and "Percent Value". Both grids are currently empty and display "No data". At the bottom right, there are "Cancel" and "Save" buttons.

Рисунок 2.10 – Додавання націнки або знижки

Для редагування вже існуючого прейскуранту потрібно натисканням миші вибрати бажаний прейскурант та натиснути кнопку «Edit Price List», відкриється спливаюче вікно, у якому поля вже будуть заповнені даними про прейскурант із сесії (рис. 2.11).

The screenshot shows a web form titled "Edit Price list — TestChild". It has four tabs: "General", "Surcharge / Discount", "Extra Properties", and "Attachments". The "General" tab is active. The form contains the following fields:

- Name:** Text input field containing "TestChild".
- Based On:** A checked checkbox followed by a dropdown menu containing "ChildSave".
- Active:** A dropdown menu containing "Yes".
- External ID:** Text input field containing "externl".
- Description:** A large text area containing "abcde".

At the bottom right of the form, there are two buttons: "Cancel" and "Save".

Рисунок 2.11 – Редагування існуючого прейскуранта

Видалення та дуплікація здійснюються вибором потрібного прейскуранта а потім натисканням відповідної кнопки «Delete Price List» «Duplicate Price List» на панелі інструментів і підтвердженням дії (рис 2.12).

The screenshot shows a confirmation dialog box with the following content:

- Title:** "Delete Price list"
- Message:** "Do you want to delete these price lists: "Musterpreisliste" ?"
- Buttons:** Two buttons, "Yes" and "No". The "Yes" button is highlighted in grey, indicating it is the default or selected option.

Рисунок 2.12 – Видалення прейскурантів

## ВИСНОВКИ

Процес створення інформаційних систем на базі Інтернету не відрізняється від написання програм. Етапи визначення вимог, аналіз, реалізація, проектування, тестування цього процесу також проходять. Під час роботи веб-сайту, незалежно від того, для якої технології він використовується, користувач, як і у звичайній програмі, вводить дані та отримує інформацію, працює з вікнами та меню, зберігає дані на сервері та отримує відповіді.

В результаті виконання кваліфікаційної роботи бакалавра було досліджено діяльність підприємства «Nicetec GmbH». Було наведено загальну характеристику діяльності компанії, проаналізовано існуючі бізнес-процеси та наявний рівень автоматизації бізнес-процесів. Після дослідження стало зрозуміло, що компанія є одним із лідерів ринку, але для подальшого розвитку і конкурентоспроможності потребувала зміни у своєму продукті.

Наступним етапом був досліджений процес управління преїскурантами для систем обліку та білінгу. Було встановлено, що кращим варіантом для автоматизації цього процесу є створення web-додатку на основі комп'ютерної версії. Додаток дає змогу повністю контролювати процес управління преїскурантами та вести їх облік, надаючи можливість використовувати ряд функцій для створення таблиць з даними та сортувати їх, відображаючи інформацію у зручному для користувача вигляді.

Перед розробкою та перед проектуванням було сформовано перелік вимог до web-додатку «NetInsight SCC».

Було розроблено архітектуру web-додатку для процесу управління преїскурантами для систем обліку та білінгу. Структурно-функціональна схема web-додатку була спроектована відповідно до основних задач системи. Для всіх сторінок додатку були розроблені макети та вказані основні елементи інтерфейсу сторінок і форм.

З урахуванням переваг та можливостей окремих технологій було підбрано набір сучасних технологій розробки для створення web-додатку. Використовувалися такі технології і програмні засоби як :

- Asp.NET Core 3
- C#
- CSS3
- SQL
- JavaScript
- Ajax
- DevExtreme
- HTML5
- Microsoft Visual Studio 2019
- AdobeXD.

Побудувавши систему та всі її модулі було проведено контрольне тестування, яке дало чітке розуміння, про коректну роботу усіх функціональних елементів у відповідності до поставлених вимог і обмежень.

Для успішного впровадження і полегшення розуміння можливостей системи та способу її використання було розроблено інструкцію використання web-додатку для процесу управління прејскурантами для систем обліку та білінгу.

## СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ

1. Nicetec GmbH [Електронний ресурс] – Режим доступу до ресурсу: <https://www.nicetec.de/en/>.
2. Аналіз бізнес-процесів підприємства [Електронний ресурс] – Режим доступу до ресурсу: [https://pidru4niki.com/1471121353661/ekonomika/analiz\\_biznes-protseviv\\_pidpriyemstva](https://pidru4niki.com/1471121353661/ekonomika/analiz_biznes-protseviv_pidpriyemstva).
3. Бізнес-процеси підприємства [Електронний ресурс] – Режим доступу до ресурсу: [http://sophus.at.ua/publ/2016\\_12\\_14\\_kampodilsk/sekcija\\_section\\_3\\_2016\\_12\\_14/biznes\\_procesi\\_pidpriemstva\\_sutnist\\_ta\\_klasifikacija/136-1-0-2192](http://sophus.at.ua/publ/2016_12_14_kampodilsk/sekcija_section_3_2016_12_14/biznes_procesi_pidpriemstva_sutnist_ta_klasifikacija/136-1-0-2192).
4. Business Process Modeling Notation [Електронний ресурс] – Режим доступу до ресурсу: <https://searchcio.techtarget.com/definition/Business-Process-Modeling-Notation>.
5. How to create BPMN diagram [Електронний ресурс] – Режим доступу до ресурсу: <https://www.visual-paradigm.com/tutorials/how-to-create-bpmn-diagram/>.
6. Інформаційні системи та їх роль в управлінні економікою [Електронний ресурс] – Режим доступу до ресурсу: <https://www.uzhnu.edu.ua/uk/infocentre/get/6742>.
7. Проектування додатків [Електронний ресурс] – Режим доступу до ресурсу: [https://stud.com.ua/97678/informatika/proektuvannya\\_dodatktiv](https://stud.com.ua/97678/informatika/proektuvannya_dodatktiv).
8. Розробка веб-додатків [Електронний ресурс] – Режим доступу до ресурсу: <https://armedsoft.com/ua/services/rozrobka-veb-dodatktiv>.
9. Вимоги до системи: класифікація FURPS+ [Електронний ресурс] – Режим доступу до ресурсу: <https://sysana.wordpress.com/2010/09/16/furps/>.
10. Complete Beginner's Guide to Information Architecture UX Booth [Електронний ресурс] – Режим доступу до ресурсу: [http://www.uxbooth.com/articles/complete-beginners-guide-toinformation-](http://www.uxbooth.com/articles/complete-beginners-guide-to-information-)

architecture/.

11. WP eCommerce [Електронний ресурс] – Режим доступу до ресурсу: <https://wordpress.org/plugins/wp-e-commerce/>.

12. Вимоги до системи: класифікація FURPS+ [Електронний ресурс] – Режим доступу до ресурсу: eShop Languages. 2019. URL: <https://ru.wordpress.org/plugins/eshop-languages/> (дата звернення: 05.11.2019). 70. eShop Support. 2014. URL: <http://quirm.net/2014/05/19/eshop-support/> (дата звернення: 05.11.2019).

13. Клієнт серверна архітектура та ролі серверів [Електронний ресурс] – Режим доступу до ресурсу: <https://medium.com/@IvanZmerzlyi/%D0%BA%D0%BB%D1%96%D1%94%D0%BD%D1%82-%D1%81%D0%B5%D1%80%D0%B2%D0%B5%D1%80%D0%BD%D0%B0-%D0%B0%D1%80%D1%85%D1%96%D1%82%D0%B5%D0%BA%D1%82%D1%83%D1%80%D0%B0-%D1%82%D0%B0-%D1%80%D0%BE%D0%BB%D1%96-%D1%81%D0%B5%D1%80%D0%B2%D0%B5%D1%80%D1%96%D0%B2-9893d8048229>.

14. Rouse, Margaret (2017). Microsoft SQL Server. Available at: <http://searchsqlserver.techtarget.com/definition/SQL-Server>.

15. Parker, Zachary. Poe, Scott. Vrbsky, V Susan (2013). Comparing noSQL MongoDB to an SQL DB. Proceedings of the 51st ACM Southeast Conference. DOI: 10.1145/2498328.2500047.

16. Wu, Ming Chieh. Huang, Fu Yin. Lee John (2015). Comparison Between MongoDB and MS-SQL Databases on the TWC Website. American Journal of Software Engineering and Applications. DOI: 10.11648/j.ajsea.20150402.12.

17. Архітектура клієнт-сервер. Сховища даних. [Електронний ресурс] – Режим доступу до ресурсу: <http://educational.mariroz.com/InformTechVInfrastrRynku/lect/lect8.pdf>.

18. Azure DevOps [Електронний ресурс] – Режим доступу до ресурсу: <https://azure.microsoft.com/ru-ru/overview/what-is-devops/>.
19. What is Azure DevOps [Електронний ресурс] – Режим доступу до ресурсу: <https://www.devopsgroup.com/insights/resources/tutorials/all/what-is-azure-devops/#:~:text=Azure%20DevOps%20is%20a%20Software,for%20orchestrating%20a%20DevOps%20toolchain.>
20. Microsoft Visual Studio [Електронний ресурс] – Режим доступу до ресурсу: <https://visualstudio.microsoft.com/>.
21. What`s new in Microsoft Visual Studio 2019 [Електронний ресурс] – Режим доступу до ресурсу: [https://docs.microsoft.com/en-us/visualstudio/ide/whats-new-visual-studio-2019?view=vs-2019.](https://docs.microsoft.com/en-us/visualstudio/ide/whats-new-visual-studio-2019?view=vs-2019)
22. What is PostgreSQL [Електронний ресурс] – Режим доступу до ресурсу: [https://www.postgresql.org/docs/current/intro-what-is.html.](https://www.postgresql.org/docs/current/intro-what-is.html)
23. What is DBeaver [Електронний ресурс] – Режим доступу до ресурсу: <https://database.guide/what-is-dbeaver/>.
24. The .NET Platform [Електронний ресурс] – Режим доступу до ресурсу: [https://dotnet.microsoft.com/learn/aspnet/what-is-aspnet.](https://dotnet.microsoft.com/learn/aspnet/what-is-aspnet)
25. Руководство по Asp.NET Core 3 [Електронний ресурс] – Режим доступу до ресурсу: <https://metanit.com/sharp/aspnet5/>.
26. Особливості та структура інформаційного забезпечення автоматизованої інформаційної системи [Електронний ресурс] – Режим доступу до ресурсу: [http://studcon.org/osoblyvosti-ta-struktura-informaciynogo-zabezpechennya-avtomatyzovanoyi-informaciynoyi-systemy.](http://studcon.org/osoblyvosti-ta-struktura-informaciynogo-zabezpechennya-avtomatyzovanoyi-informaciynoyi-systemy)
27. dxDataGrid [Електронний ресурс] – Режим доступу до ресурсу: [https://js.devexpress.com/Documentation/ApiReference/UI\\_Widgets/dxDataGrid/](https://js.devexpress.com/Documentation/ApiReference/UI_Widgets/dxDataGrid/).
28. dxDataGrid Configuration [Електронний ресурс] – Режим доступу до ресурсу: [https://js.devexpress.com/Documentation/ApiReference/UI\\_Widgets/dxDataGrid/Configuration/](https://js.devexpress.com/Documentation/ApiReference/UI_Widgets/dxDataGrid/Configuration/).

29. DevExpress Documentation [Електронний ресурс] – Режим доступу до ресурсу: <https://js.devexpress.com/Documentation/>.
30. Price Management Concept Guide[Електронний ресурс] – Режим доступу до ресурсу: <https://doc.oroinc.com/user/concept-guides/pricing/#user-guide-pricing>.
31. System Testing [Електронний ресурс] – Режим доступу до ресурсу: <https://www.guru99.com/system-testing.html>.
32. Testing and quality control [Електронний ресурс] – Режим доступу до ресурсу: <http://www.isd.dp.ua/ru/services/testing-and-quality-control?q=/services/testing-and-quality-control.html>.
33. Особливості та структура інформаційного забезпечення автоматизованої інформаційної системи [Електронний ресурс] – Режим доступу до ресурсу: <http://studcon.org/osoblyvosti-ta-struktura-informaciyного-zabezpechennya-avtomatyzovanoyi-informaciynoyi-systemy>.
34. Adam Freeman, 2020, Pro ASP.NET Core 3: Develop Cloud-Ready Web Applications Using MVC 3, ISBN 978-1484203989.
35. Бібліографічні посилання: загальні положення та правила складання (ДСТУ 8302:2015) [Електронний ресурс] : презентація / Наук. б-ка НАУКМА ; уклад. Т. О. Патрушева. – Київ, 2016. – Режим доступу :<https://www.slideshare.net/naukmalibrary/83022015>.–(дата звернення: 12.06.2017).
36. Бібліографічне посилання: загальні положення та правила складання [Електронний ресурс] : презентація / Наук. б-ка Харків. нац. мед.ун-ту ; уклад. С. А. Кравченко. – Харків, 2016. – Режим доступу : <http://repo.knmu.edu.ua/handle/123456789/14078>. – (дата звернення: 12.06.2017) ДСТУ 8302:2015. Інформація та документація. Бібліографічні посилання. Загальні положення та правила складання. – Чинний від 2016–07– Київ : ДП «УкрНДНЦ», 2016. – 16 с.
37. Міжнародні стилі цитування та посилання в наукових роботах : метод. рек. / автори-уклад.: О. Боженко, Ю. Корян, М. Федорець ; редкол.: В.



С. Пашкова, О. В. Воскобойнікова-Гузєва, Я. Є. Сошинська, О. М. Бруй ; Наук-техн. б-ка ім. Г. І. Денисенка Нац. техн. ун-ту України «Київ. політехн. ін-т ім. Ігоря Сікорського», Укр. бібл. Асоціація. – Київ : УБА, 2016. – 117 с.

38. Приклади оформлення використаних джерел: відповідно до Нац. стандарту України ДСТУ 8302:2015 [Електронний ресурс] / ВД «Академперіодика» НАН України. – Київ, 2016. – Режим доступу : <http://histj.oa.edu.ua/assets/files/Posylannia.pdf>. – (дата звернення: 12.06.2017).

39. Приклади оформлення списку використаних джерел, відповідно до ДСТУ 8302:2015 [Електронний ресурс] / Наук. б-ка Нац. юрид. ун-ту ім. Ярослава Мудрого; уклад. О. І. Самофал. – Харків, 2017. – Режим доступу : <http://library.nlu.edu.ua/biblioteka/sait/nauka/gost/spisok-DSTU.pdf>. – (дата звернення: 27.03.2017).

40. Web-development definition [Електронний ресурс] – Режим доступу до ресурсу: <https://blog.openclassrooms.com/en/2018/03/28/web-development-definition/>.

41. What is a web developer [Електронний ресурс] – Режим доступу до ресурсу: <https://www.bitdegree.org/tutorials/what-is-a-web-developer/>.

# ДОДАТКИ

Додаток А  
(обов'язковий)

## SUMMARY

Necheporenko I.D. Automation for price list management process for accounting and billing systems – Bachelor-level Qualification Thesis. Sumy State University, Sumy, 2020.

In the bachelor`s thesis the essence of work of the IT company, currently popular systems and technologies of web distribution and automated work. The analysis of interaction of the enterprise with clients in web-oriented systems is carried out. The description of main business processes at the enterprise is carried out. The main goal of the research is automation for price list management process with the help of web-technologies.

Key words: automation, enterprise, web application, information system, price list management.

## АНОТАЦІЯ

Нечепоренко І.Д. Автоматизація процесу управління прейскурантами для систем обліку та білінгу. – Кваліфікаційна робота бакалавра. Сумський державний університет, Суми, 2020 р.

У роботі досліджено сутність роботи ІТ-компанії, актуальні системи і технології веб-розробки та автоматизації. Проаналізовано взаємодію підприємства з клієнтами у веб-орієнтованих системах. Здійснений опис основних бізнес-процесів на підприємстві. Головною метою дослідження є

автоматизація процесу управління преїскурантами за допомогою веб-технологій.

Ключові слова: автоматизація, підприємство, web-додаток, інформаційна система, управління преїскурантами.

Додаток Б  
(інформаційний)



May 31, 2020

**Recommendation Letter**

This letter is to confirm that Illia Necheporenko, a Sumy State University student, born September 7, 1998, served as an intern at our company from March 1 through May 31, 2020.

nicetec GmbH is one of the leading providers in the D/A/CH market with its solutions in the area of IT Finance Management and Intercompany Service Charging. Our business areas include the development and distribution of our solutions as well as the associated consulting and training services.

The tasks assigned to Mr. Necheporenko at nicetec included the following main activities:

- Development and implementation of a web-module for an automated price list management process within the accounting and billing system.
- Research and layout for this web-application.

During his term he learned some innovative technologies and programming languages, which he has used practically. This tasks, in particular, required a high degree of professional skills, tenacity and persistence.

Mr. Necheporenko worked with ASP.NET Core 3.0, which is a server side technology in web development for writing sever side logic. He is familiar now with designing web pages with HTML5. He worked with the client side language JavaScript and libraries like JQuery and DevExtreme. He also learned how to work with AJAX technology for partial rendering of web pages. Also, he is now familiar with debugging on the browser side (debugging JavaScript code and network diagnostic) in a variety of major browsers like Microsoft Edge, Mozilla Firefox or Google Chrome. On the server side he used the Visual Studio IDE for writing C# code and debugging server side code. Furthermore he gained experience in PostgreSQL database development and writing SQL queries, triggers and functions.

We have been very pleased with the work Mr. Necheporenko has accomplished in his internship with our company. He has demonstrated good working knowledge and he showed high commitment and initiative. He always demonstrated a good understanding of technical interrelationships. His performance always deserved our full satisfaction. His personal conduct was impeccable.

As of today, Mr. Necheporenko is leaving our company, as his internship ends. We thank him for the good and successful cooperation and regret his departure. We wish him all the best in his future career and life. If we can provide you with any additional information or if you have any questions regarding Mr. Necheporenko's work, please feel free to contact us.

Sincerely,



Ralf Meyer, CEO

nicetec GmbH Gewerbepark 5b D-49143 Bissendorf Germany  
Phone +49 5402-6446-0 Fax +49 5402-6446-10 E-Mail: info@nicetec.de Internet: www.nicetec.de  
CEO: Ralf Meyer Amtsgericht Osnabrück HRB 19413

Рисунок Б.1 – Рекомендаційний лист від підприємства «Nicetec GmbH»

## Додаток В

### Файл PriceLists.cshtml

```

@{
    ViewData["Title"] = "PriceLists";
}
@model NetinsightWebPortalExt.Models.BIPricelist
<script src="~/js/Prices/Pricelists.js"></script>

<link rel="stylesheet" type="text/css"
href="~/css/Prices/PriceLists.css" media="screen" />
<link rel="stylesheet" type="text/css"
href="~/css/Common/dxOverrides.css"
media="screen" />

<header id="pricelistsHeader">
    <h3 class="content-
header">@Resources.strPriceLists</h3>
    <br />
    <div>
        @(Html.DevExtreme().Toolbar()
            .ID("priceListsViewToolbar")
            .Items(items =>
                {
                    items.Add()
                        .Widget(w => w
                            .Button()
                                .FocusStateEnabled(false)
                                    Location(ToolBarItemLocation.Before);

                            items.Add()
                                .Widget(w => w
                                    .Button()
                                        .FocusStateEnabled(false)
                                            .ID("deletePriceListsDDB")
                                                .StylingMode(ButtonStylingMode.Text)
                                                    .Icon("minus")
                                                        .Text(Resources.strDeletePriceList)
                                                            .Disabled(true)
                                                                .OnClick("deletePriceList")
                                                                    )
                                                                .LocateInMenu(ToolBarItemLocateInMenuMode.Auto)
                                                                    .Location(ToolBarItemLocation.Before);

                            items.Add()
                                .Widget(w => w
                                    .Button()
                                        .FocusStateEnabled(false)
                                            .ID("duplicatePriceList")
                                                .StylingMode(ButtonStylingMode.Text)
                                                    .Icon("far fa-copy")
                                                        .Text(Resources.strDuplicatePriceList)
                                                            .Disabled(true)
                                                                .OnClick("duplicatePriceList")
                                                                    )
                                                                .LocateInMenu(ToolBarItemLocateInMenuMode.Auto)
                                                                    .Location(ToolBarItemLocation.Before);

                            items.Add()
                                .Widget(w => w
                                    .Button()
                                        .FocusStateEnabled(false)
                                            .ID("addNewPriceListsDDB")
                                                .Icon("plus")
                                                        .Text(Resources.strAddPriceList)
                                                            .OnClick("addNewPriceList")
                                                                    )
                                                                .LocateInMenu(ToolBarItemLocateInMenuMode.Auto)
                                                                    .Location(ToolBarItemLocation.Before);

                            items.Add()
                                .Widget(w => w
                                    .Button()
                                        .FocusStateEnabled(false)
                                            .ID("editPriceListToolbarButton")
                                                .Icon("fas fa-wrench")
                                                        .Text(Resources.strEditPriceList)
                                                            .Disabled(true)
                                                                .OnClick("editPriceList")
                                                                    )
                                                                .LocateInMenu(ToolBarItemLocateInMenuMode.Auto)
                                                                    .Location(ToolBarItemLocation.Before);

                            items.Add()
                                .Widget(w => w
                                    .Button()
                                        .FocusStateEnabled(false)
                                            .ID("pricesToolbarButton")
                                                .Icon("fas fa-tag")
                                                        .Text(Resources.strPrices)
                                                            .OnClick("goToPricesPage")
                                                                .Disabled(true)
                                                                    )
                                                                .LocateInMenu(ToolBarItemLocateInMenuMode.Auto)
                                                                    .Location(ToolBarItemLocation.Before);

                            items.Add()
                                .Widget(w => w
                                    .Button()
                                        .FocusStateEnabled(false)
                                            .ID("addNewPriceList")
                                                .Text(Resources.strAddPriceList)
                                                    .OnClick("addNewPriceList")
                                                        )
                                                                .LocateInMenu(ToolBarItemLocateInMenuMode.Auto)
                                                                    .Location(ToolBarItemLocation.Before);

                            items.Add()
                                .Widget(w => w
                                    .Button()
                                        .FocusStateEnabled(false)
                                            .ID("duplicatePriceList")
                                                .Text(Resources.strDuplicatePriceList)
                                                    .OnClick("duplicatePriceList")
                                                        )
                                                                .LocateInMenu(ToolBarItemLocateInMenuMode.Auto)
                                                                    .Location(ToolBarItemLocation.Before);

                            items.Add()
                                .Widget(w => w
                                    .Button()
                                        .FocusStateEnabled(false)
                                            .ID("editPriceList")
                                                .Text(Resources.strEditPriceList)
                                                    .OnClick("editPriceList")
                                                        )
                                                                .LocateInMenu(ToolBarItemLocateInMenuMode.Auto)
                                                                    .Location(ToolBarItemLocation.Before);

                            items.Add()
                                .Widget(w => w
                                    .Button()
                                        .FocusStateEnabled(false)
                                            .ID("pricesPage")
                                                .Text(Resources.strPrices)
                                                    .OnClick("goToPricesPage")
                                                        )
                                                                .LocateInMenu(ToolBarItemLocateInMenuMode.Auto)
                                                                    .Location(ToolBarItemLocation.Before);
                }
            )
        )
    </div>
</header>

```

```

        .FocusStateEnabled(false)

.StylingMode(ButtonStylingMode.Text)
.ID("pricelistsColumnChooser")
.Icon("columnchooser")

.OnClick("showPriceListColumnChooser")
.Hint(Resources.strColumnChooser)
)

.LocateInMenu(ToolBarItemLocateInMenuMode.Auto)
.Location(ToolBarItemLocation.After);

items.Add()
.Widget(w => w
.Button()
.FocusStateEnabled(false)

.StylingMode(ButtonStylingMode.Text)
.ID("refreshTreeList")
.Icon("refresh")
.OnClick("refreshPriceListView")
.Hint(Resources.strRefresh)
)

.LocateInMenu(ToolBarItemLocateInMenuMode.Auto)
.Location(ToolBarItemLocation.After);

items.Add()
.Widget(w => w
.Button()
.FocusStateEnabled(false)

.StylingMode(ButtonStylingMode.Text)
.ID("clearSelectionOfTree")
.Icon("far fa-minus-square")
.OnClick("clearSelectionOfList")
.Hint(Resources.strClearSelection)
)

.LocateInMenu(ToolBarItemLocateInMenuMode.Auto)
.Location(ToolBarItemLocation.After);
})
)
</div>
<br />
</header>

<div class="main" id="pricelistMainView"
style="margin-top: -40px;">
<div style="color: #3c8dbc; background-color:
#3c8dbc; border-color: #3c8dbc; height: 3px;">
<hr />
</div>
<section id="treeListContainer">
<div class="treeListContainer dx-viewport"
style="height: 100%; position: page;">
@ (Html.DevExtreme().TreeList<PriceListData>()

.DataSource(
d => d.Mvc()
.Controller("PriceLists")
.LoadAction("Get")

.Key(nameof(PriceListData.PricelistId))
)
.ID("pricelists")

.KeyExpr(nameof(PriceListData.PricelistId))

.ParentIdExpr(nameof(PriceListData.Parent))
.RootValue(null)
.RowDragging(rd => rd
.AllowDropInsideItem(true)
.OnDragChange("onDragChange")
.OnReorder("onReorder")
.ShowDragIcons(true)
)
.AutoExpandAll(true)
.ColumnAutoWidth(true)
.ShowBorders(false)
.Selection(s => s
.Mode(SelectionMode.Single)
)
.ShowRowLines(true)
.ColumnHidingEnabled(false)
.Scrolling(scrolling => scrolling

.ShowScrollbar(ShowScrollbarMode.Always)
.UseNative(true)
.Mode(TreeListScrollingMode.Standard)
)
.Columns(columns =>
{
columns.AddFor(m => m.PricelistId)
.HeaderCellTemplate(@<text><b
style="color: black;">ID</b></text>)
.Caption("ID")

.Alignment(HorizontalAlignment.Left)
.Visible(false);

columns.AddFor(m => m.Name)
.HeaderCellTemplate(@<text><b
style="color:
black;">@Resources.strPriceList</b></text>);

columns.AddEnumColumnTL(m =>
m.IsActive, typeof(enumPriceListStatus),
EnumService)

.Alignment(HorizontalAlignment.Left)
.HeaderCellTemplate(@<text><b
style="color:
black;">@Resources.strActive</b></text>)
.Name("Status")
.Caption(Resources.strState);

columns.AddFor(m => m.ClientName)

.Alignment(HorizontalAlignment.Left)

```

```

        .HeaderCellTemplate(@<text><b
style="color:
black;">@Resources.strClient</b></text>)
        .Caption(Resources.strClient);

        columns.AddFor(m=> m.Parent)
        .Lookup(lookup => lookup
        .DataSource(d =>
d.Mvc().Controller("PriceLists").LoadAction("Get").
Key("Parent"))
        .DisplayExpr("Name")

        .ValueExpr(nameof(PriceListData.PricelistId))
        )
        .HeaderCellTemplate(@<text><b
style="color:
black;">@Resources.strParent</b></text>);

        columns.AddFor(m => m.ExternalId)
        .HeaderCellTemplate(@<text><b
style="color:
black;">@Resources.strExternalID</b></text>);

        columns.AddFor(m => m.Description)
        .HeaderCellTemplate(@<text><b
style="color:
black;">@Resources.strDescription</b></text>);

        columns.AddFor(m =>
m.MarkupPercent)
        .HeaderCellTemplate(@<text><b
style="color:
black;">@Resources.strMarkup</b></text>)

        .DataType(GridColumnDataType.Number);
    })
    .SearchPanel(sp => sp
        .Visible(true)
        .HighlightSearchText(true)
        .HighlightCaseSensitive(true)
    )
    .ColumnChooser(cc => cc
        .Enabled(false)

        .Mode(GridColumnChooserMode.Select)
        .Title(Resources.strColumnChooser)
        .Height(300)
    )
    .RowAlternationEnabled(true)
    .ShowColumnLines(true)
    .ColumnAutoWidth(true)
    .AllowColumnResizing(true)
    .AllowColumnReordering(true)
    .ColumnFixing(cf => cf
        .Enabled(true)
    )
    .FilterRow(fr => fr
        .Visible(true)
    )
    .Paging(pg => pg
        .Enabled(true)
        .PageIndex(0)

        .PageSize(20)
    )
    .Pager(pgr =>
    {
        pgr.Visible(true);
        pgr.AllowedPageSizes(new[] { 10, 20,
50, 100 });
        pgr.ShowNavigationButtons(true);
        pgr.ShowPageSizeSelector(true);
    })
    .ShowBorders(true)
    .FilterRow(f => f.Visible(true))
    .RemoteOperations(ro => ro
    .Grouping(true)
    .Filtering(true)
    )

    .OnSelectionChanged("onSelectionChangedPriceList
s")

    .OnInitialized("onInitializedPriceLists")
    .OnCellDbClick("goToPricesPage")
    .OnNodesInitialized(
        @<text>
        function(e) {
            $pricelists = e.component;
        }
        </text>
    )
    )
    @ (Html.DevExtreme().ContextMenu()
        .Width(200)
        .Target("#pricelists")

        .OnClick("priceListsCMenuItemClicked")
        .DisplayExpr("Text")
        .ItemsExpr("Items")
        .DataSource(new object[]
        {
            new { Text = Resources.strExpandAll},
            new { Text = Resources.strCollapsAll},
        })
    )
    </div>
</section>
</div>

    @ (Html.DevExtreme().Toast()
        .DisplayTime(2000)
        .ID("priceListToast")
        .Position(pos => pos
            .My(HorizontalAlignment.Center,
VerticalAlignment.Center)
            .At(HorizontalAlignment.Center,
VerticalAlignment.Top)
            .Of(new JS("treeListContainer")))
        )
        .Width(new JS("getWidthOfTreeList"))
    )
    <div id="pricelist-popup-div"></div>
    <div id="pricelist-component-popup-div"></div>

```

## Додаток Г

### Файл PriceListsController.cs

```
using DevExtreme.AspNet.Data;
using DevExtreme.AspNet.Mvc;
using Microsoft.AspNetCore.Mvc;
using Microsoft.EntityFrameworkCore;
using System;
using System.Collections.Generic;
using System.Linq;
using System.Threading.Tasks;
using NetinsightWebPortalExt.Models;
using DevExtreme.AspNet.Data.ResponseModel;
using NetinsightWebPortalExt.Utility;
using System.Linq.Dynamic.Core;
using NetinsightWebPortalExt.ModelHelper;
using log4net;
using log4net.Util;
using Microsoft.AspNetCore.Authorization;
using NetinsightWebPortalExt.Infrastructure;
using Microsoft.EntityFrameworkCore.Storage;
using System.Diagnostics;

namespace
NetinsightWebPortalExt.Controllers.Prices
{
    [Authorize(Policy = Policies.PortalSession)]
    public class PriceListsController : Controller
    {
        private readonly NiContext Context;
        private ISessionObject SessionObject { get; set; }

        private protected ILog Logger { get; set; }
        //private int PricelistIndex { get; set; }
        private int PricesIndex { get; set; }
        private int PricelistOverheadsIndex { get; set; }

        public PriceListsController(NiContext context,
ISessionObject sessionObject, ILog logger)
        {
            this.Context = context;
            this.SessionObject = sessionObject;
            this.Logger = logger;
        }

        public IActionResult Index()
        {
            return
View("Views/Prices/PriceLists.cshtml");
        }

        [HttpGet]
        public object
GetPriceListsForSelectBox(DataSourceLoadOptions
loadOptions)
        {
            var query =
                select new
                {
                    blpricelists.PricelistId,
                    blpricelists.Name,
                    blpricelists.IsActive,
                    blpricelists.Parent,
                    blpricelists.ExternalId,
                    blpricelists.Description,
                    blpricelists.MarkupPercent,
                    blpricelists.Favorite
                };

            return DataSourceLoader.Load(query,
loadOptions);
        }

        private List<PriceListData> GetPricelists()
        {
            List<PriceListData> pricelists = new
List<PriceListData>();

            try
            {
                var pricelistQuery =
                    this.Context.BIPricelist
                        .Include(cm => cm.Client)
                        .OrderBy(pricelist =>
pricelist.PricelistId)
                        .Select(pricelist => new
PriceListData(pricelist, pricelist.Client));

                pricelists = pricelistQuery.ToList();

                List<decimal> PricelistId =
pricelists.Select(price =>
(decimal)price.PricelistId).ToList();

                var query =
                    Context.BIPricelistOverheads
                        .Where(bl => PricelistId
                            .Contains(bl.PricelistId))
                        .GroupJoin(Context.BIComponents,
blPrice => blPrice.ComponentId, blComp =>
blComp.ComponentId, (blPriceOver, blComps) =>
new { blPriceOver, blComps })
                        .SelectMany(joinRes =>
joinRes.blComps.DefaultIfEmpty(), (JoinRes,
blComp) => new { JoinRes.blPriceOver, blComp })
                        .ToList();

                foreach (var priceList in pricelists)
                {
                    var v = query.Where(pric =>
pric.blPriceOver.PricelistId ==
priceList.PricelistId).ToList();
                }
            }
        }
    }
}
```



```

        priceList.PriceListOverheads = v.Select(r
=> new PriceListOverhead(r.blPriceOver,
r.blComp)).ToList();
    }
    catch (Exception ex)
    {
        this.Logger.ErrorExt(ex);
    };
    return pricelists;
}

[HttpGet]
public ActionResult<object>
Get(DataSourceLoadOptions loadOptions)
{
    LoadResult result = new LoadResult();
    try
    {
        result =
DataSourceLoader.Load(this.GetPricelists(),
loadOptions);
    }
    catch (Exception ex)
    {
        this.Logger.ErrorExt(ex);
        return this.Problem(ex.InnerException !=
null ? ex.InnerException.Message : ex.Message);
    };
    return this.Ok(result);
}

[HttpPut]
public async Task<IActionResult>
Saving(PriceListDataParam priceListDataParam)
{
    try
    {
        PriceListData sessionPriceList =
SessionObject.pricelistData;

        BIPracelist pracelist = null;

        this.Context.Database.BeginTransaction();

        var indexPriceList =
Context.BIIndex.FirstOrDefault();

        if (sessionPriceList.PricelistId != null)
        {
            pracelist = await
Context.BIPracelist.Where(prc => prc.PricelistId ==
sessionPriceList.PricelistId).FirstOrDefaultAsync();
        }
        else
        {
            pracelist = new Models.BIPracelist();

            indexPriceList.PricelistId++;

            pracelist.PricelistId =
indexPriceList.PricelistId.Value;

            priceListDataParam.PricelistId =
pracelist.PricelistId;

            Context.BIPracelist.Add(pracelist);
        }
        pracelist.Name =
priceListDataParam.Name;
        pracelist.IsActive =
(decimal)priceListDataParam.IsActive;
        pracelist.ClientId =
priceListDataParam.ClientId;
        pracelist.Description =
priceListDataParam.Description;
        pracelist.Parent =
priceListDataParam.Parent;
        pracelist.ExternalId =
priceListDataParam.ExternalId;
        pracelist.MarkupPercent =
priceListDataParam.MarkupPercent;
        pracelist.Favorite =
Convert.ToDecimal(priceListDataParam.Favorite);

        List<decimal> pracelistCmpIdsToDel =
sessionPriceList.PriceListOverheads.Where(priCmp
=> priCmp.IsDeleted == true).Select(priCmp =>
(decimal)priCmp.PricelistOverheadId).ToList();

        List<decimal> alldelIds = new
List<decimal>();

        alldelIds.AddRange(pracelistCmpIdsToDel);

        //Delete Discount/Surcharge components

Context.BIPracelistOverheads.RemoveRange(Context
.BIPracelistOverheads.Where(priCmp =>
alldelIds.Contains(priCmp.PricelistOverheadId)));

        //Save Discount/Surcharge components
        var sessionPriceListOverheads =
sessionPriceList.PriceListOverheads.Where(pover =>
pover.IsDeleted == false).ToList();

        List<decimal> pracelistoverheadids =
sessionPriceListOverheads.Where(pover =>
pover.IsDeleted == false &&
pover.PricelistOverheadId != null).Select(pover =>
(decimal)pover.PricelistOverheadId).ToList();

        List<BIPracelistOverheads>
pracelistOverheads =
Context.BIPracelistOverheads.Where(pover =>
pracelistoverheadids.Contains(pover.PricelistOverhea
dId)).ToList();
        var indexOver =
Context.BIIndex.FirstOrDefault();

        foreach (var priceOverhead in
sessionPriceListOverheads)
        {

```

```

        BIPricelistOverheads
        blPricelistOverheads = null;

        if (priceOverhead.PricelistOverheadId <
0)
        {
            blPricelistOverheads = new
            BIPricelistOverheads()
            {
                Pricelist = pricelist,
            };
        }
        else
        {
            blPricelistOverheads =
            pricelistOverheads.Where(pover =>
            pover.PricelistOverheadId ==
            priceOverhead.PricelistOverheadId).SingleOrDefault
            ();
        }
        blPricelistOverheads.ComponentId =
        priceOverhead.ComponentId;
        blPricelistOverheads.ValuePercent =
        priceOverhead.Percent / 100;
        blPricelistOverheads.Description =
        priceOverhead.Description;
        blPricelistOverheads.Value =
        priceOverhead.Value;

        if (priceOverhead.PricelistOverheadId <
0)
        {
            indexOver.PricelistOverheadId++;

            blPricelistOverheads.PricelistOverheadId =
            indexOver.PricelistOverheadId.Value;

            priceOverhead.PricelistOverheadId =
            blPricelistOverheads.PricelistOverheadId;

            Context.BIPricelistOverheads.Add(blPricelistOverhe
            ads);
        }

        if (sessionPriceList.PricelistId == null)
        {
            this.Context.BIPricelist.Add(pricelist);
        }
        this.Context.SaveChanges();

        this.Context.Database.CommitTransaction();
    }
    catch (Exception ex)
    {
        this.Logger.ErrorExt(ex);
        return this.Problem(ex.InnerException !=
        null ? ex.InnerException.Message : ex.Message);
    }
}

        finally
        {
            if
            (this.Context.Database.CurrentTransaction != null)
            this.Context.Database.RollbackTransaction();
        }
        return this.Ok();
    }

    [HttpGet]
    public IActionResult EditMode(decimal?
    PriceListId)
    {
        try
        {
            PriceListData pricelistData = null;
            if (PriceListId != null)
            {
                BIPricelist pricelist = Context.BIPricelist
                .Include(cl => cl.Client)
                .Where(prc => prc.PricelistId ==
                PriceListId)
                .FirstOrDefault();

                pricelistData = new
                PriceListData(pricelist);

                var client =
                Context.BIPricelist.FirstOrDefault(cl => cl.ClientId
                == pricelist.ClientId);
                pricelistData.ClientId =
                pricelist.ClientId;

                var queryForDiscount =
                Context.BIPricelistOverheads
                .GroupJoin(Context.BIComponents,
                blPrice => blPrice.ComponentId, blComp =>
                blComp.ComponentId, (blPriceOver, blComps) =>
                new { blPriceOver, blComps })
                .SelectMany(joinRes =>
                joinRes.blComps.DefaultIfEmpty(), (JoinRes,
                blComp) => new { JoinRes.blPriceOver, blComp })
                .ToList();

                var v = queryForDiscount.Where(pric =>
                pric.blPriceOver.PricelistId ==
                pricelist.PricelistId).ToList();
                pricelistData.PriceListOverheads =
                v.Select(r => new PriceListOverhead(r.blPriceOver,
                r.blComp)).ToList();
            }
            else
            {
                pricelistData = new PriceListData();
            }
            SessionObject.pricelistData = pricelistData;

            return
            PartialView("~/Views/Prices/PriceListPopup.cshtml"
            , pricelistData);
        }
    }
}

```

```

    }
    catch (Exception ex)
    {
        this.Logger.ErrorExt(ex);
        return this.Problem(ex.InnerException !=
null ? ex.InnerException.Message : ex.Message);
    }
}

[HttpPost]
public ActionResult
changePriceListHierarchy(decimal
sourceDataPriceListId, decimal
targetDataPriceListId)
{
    try
    {
        var listWithParentToBeChanged =
this.Context.BIPricelist.FirstOrDefault(cgr =>
cgr.PricelistId == sourceDataPriceListId);
        listWithParentToBeChanged.Parent =
targetDataPriceListId;
        Context.SaveChanges();
    }
    catch (Exception ex)
    {
        this.Logger.ErrorExt(ex);
        return this.Problem(ex.InnerException !=
null ? ex.InnerException.Message : ex.Message);
    }
    return this.Ok();
}

[HttpGet]
public object
getClient(DataSourceLoadOptions loadOptions)
{
    return
DataSourceLoader.Load(Context.BIClients,
loadOptions);
}

[HttpDelete]
public ActionResult Delete(int[] keys)
{
    List<decimal> priceListKeys = new
List<decimal>();
    List<BIPricelist> listsToBeDeleted = new
List<BIPricelist>();

    foreach (var key in keys)
    {
        var childKeys =
findAllChildsOfPriceLists(priceListKeys, key);

        foreach (var childkey in childKeys)
        {
            listsToBeDeleted.Add(Context.BIPricelist.FirstOrDef
ault(pl => pl.PricelistId == childkey));
        }

        foreach (var list in listsToBeDeleted)
        {
            this.Context.BIPricelist.Remove(list);
        }
    }
    this.Context.SaveChanges();
    return this.Ok();
}

private List<decimal>
findAllChildsOfPriceLists(List<decimal>
priceListKeys, decimal priceListId)
{
    priceListKeys.Add(priceListId);
    var query =
        from pricelists in Context.BIPricelist
        where pricelists.Parent == priceListId
        select new
        {
            pricelists.PricelistId
        };

    foreach (var listInList in query)
    {
        if (listInList != null)
        {
            findAllChildsOfPriceLists(priceListKeys,
listInList.PricelistId);
        }
    }
    return priceListKeys;
}

[HttpPost]
public List<decimal>
DuplicatePriceList(decimal[] keys)
{
    List<decimal> PricelistIdOfNewPricelists =
new List<decimal>();

    var indexPriceList =
Context.BIIndex.FirstOrDefault();

    foreach (var key in keys)
    {
        BIPricelist pricelist =
Context.BIPricelist.FirstOrDefault(prl =>
prl.PricelistId == key);

        List<BIPricelistOverheads>
pricelistOverhead =
Context.BIPricelistOverheads.Where(pricOvr =>
pricOvr.PricelistId == pricelist.PricelistId).ToList();

        indexPriceList.PricelistId++;

        int idolPricelistId =
(int)pricelist.PricelistId;
    }
}

```



```

        Price = priceNew,
        PriceId = priceNew.PriceId,
        // ClientId = pricecomp.ClientId,
        PricecompId =
indexPriceList.PricecompId.Value,
        Portion = pricecomp.Portion,
        ComponentId =
pricecomp.ComponentId

    };

this.Context.BIPricecomp.Add(newPriceComp);
    }
    }
    Context.SaveChanges();
    transaction.Commit();
    }
    catch (Exception ex)
    {
        transaction.Rollback();
    }
    }

private void CopyComponents(BIPrice blPrice,
decimal IdolPriceId)
    {
        BIIndex index =
Context.BIIndex.FirstOrDefault();

        var priceComponentQuery =
this.Context.BIPricecomp.Where(pr => pr.PriceId ==
IdolPriceId);
        if (priceComponentQuery != null)
        {
            List<BIPricecomp>
pricecompsOfIdolPriceList =
priceComponentQuery.ToList();
            if (pricecompsOfIdolPriceList != null &&
pricecompsOfIdolPriceList.Count > 0)
            {
                for (int i = 0; i <
pricecompsOfIdolPriceList.Count(); i++)
                {
                    BIPricecomp pricecomp =
pricecompsOfIdolPriceList[i];
                    pricecomp.PriceId = blPrice.PriceId;
                    pricecomp.PricecompId =
(decimal)index.PricecompId + 1;
                    pricecomp.Price = blPrice;

this.Context.BIPricecomp.Add(pricecomp);
                    index.PricecompId++;
                }
            }
        }
    }
}
}}

```

## Додаток Д

### Файл Pricelists.js

```
var SelectedpricelistId = 1;

window.onload = function () {

    $modal = $('#pricelist-popup-div');

    $modal.dxPopup(
    {
        toolbarItems: [
            {
                toolbar: "bottom",
                location: "after",
                widget: "dxButton",
                options: {
                    text: Resources.Buttons.Cancel,
                    width: "125px",
                    onClick: function () {
                        $modal.dxPopup("instance").hide(); }
                }
            },
            {
                toolbar: "bottom",
                location: "after",
                widget: "dxButton",
                options: {
                    type: "default",
                    text: Resources.Buttons.Save,
                    width: "125px",
                    onClick: Saveform
                }
            }
        ],
        showTitle: true,
        closeOnOutsideClick: true,
        height: 725,
        width: 1100,
        position: { my: "center", at: "center", of:
"window" },
        onHiding: clearSelectionOfTree,

    });

    resizePriceListView();
//
}

function goToPricesPage(e) {
    var treeList =
$('#pricelists').dxTreeList("instance");
    var id = treeList.getSelectedRowKeys()[0];
    window.location.href = "Prices?PricelistId=" + id;
}

function addNewPriceList() {

var $modal = $('#pricelist-popup-div');

$.ajax({
    url: 'PriceLists/EditMode',
    type: 'Get'
}).done(function (data) {
    $modal.dxPopup("option", "contentTemplate",
data);
    $modal.dxPopup("option", "title",
Resources.DialogHeaders.AddPriceList);
    $modal.dxPopup("instance").show();
    resetFields();

    var e = {};
    e.value = 0;
    ActivatePriceListTab(e);
});
}

function editPriceList() {
    var tree = $('#pricelists').dxTreeList("instance");
    var id = tree.getSelectedRowKeys()[0];
    var params = decodeURIComponent($.param({
PriceListId: id }, true));
    var rowData = tree.getSelectedRowsData();
    var $modal = $('#pricelist-popup-div');

    $.ajax({
        url: 'PriceLists/EditMode?' + params,
        type: 'Get'
    }).done(function (data) {

        $modal.dxPopup("option", "contentTemplate",
data);
        $modal.dxPopup("option", "title",
Resources.DialogHeaders.EditPriceList +
rowData[0].Name);
        $modal.dxPopup("instance").show();
    });
}

function Saveform() {
    var unindexed_frmGeneral =
$("#frmGeneralDate").serializeArray();
    var unindexed_frmExtra =
$("#frmExtra").serializeArray();
    var PriceListJson = {};

    $.map(unindexed_frmGeneral, function (n, i) {
        PriceListJson[n['name']] = n['value'];
    });

    $.map(unindexed_frmExtra, function (n, i) {
        PriceListJson[n['name']] = n['value'];
    });
}
```

```

});

$('#selectBoxForParentPriceList').dxSelectBox();
var cboParentSelectBox =
$('#selectBoxForParentPriceList').dxSelectBox("instance");
PriceListJson.PricelistId =
cboParentSelectBox.option("value");

$('#ClientSelectBox').dxSelectBox();
var cboClientSelectBox =
$('#ClientSelectBox').dxSelectBox("instance");
PriceListJson.ClientId =
cboClientSelectBox.option("value");

$('#checkBoxForFavourite').dxCheckBox();
var cboFavoriteCheckBox =
$('#checkBoxForFavourite').dxCheckBox("instance");
;
PriceListJson.Favorite =
cboFavoriteCheckBox.option("value");

var popup = $('#pricelist-popup-
div').dxPopup("instance");

$.ajax({
  url: 'PriceLists/Saving',
  type: 'Put',
  data:
  {
    'priceListDataParam': PriceListJson
  },
  error: function () {
    var toast =
$('##pricelistPopupToast').dxToast("instance");
    toast.option("message",
Resources.Alerts.CantCreateWoDate);
    toast.option("type", "error");
    $('##pricelistPopupToast').dxToast("show");
  }
}).done(function () {
  var tree = $('#pricelists').dxTreeList("instance");
  tree.refresh();
  popup.hide();
});
}

function ActivatePriceListTab(e) {
  if (e.value == 0) {
    $('#tabHeaderGeneral').removeClass("hide");
    $('#tabHeaderDiscount').removeClass("hide");
    $('#tabHeaderExtra').removeClass("hide");

  } else {
    $('#tabHeaderGeneral').addClass("hide");
    $('#tabHeaderDiscount').addClass("hide");
    $('#tabHeaderExtra').addClass("hide");
  }
}
}

```

```

function buildDeleteCompButtons(e) {
  var toolbarItems = e.toolbarOptions.items;

  toolbarItems.unshift({
    //html: 'Resources.strPriceComponent',
    html: 'Component',
    location: 'before'
  });

  toolbarItems.push(
  {
    widget: 'dxButton',
    options:
    {
      icon: "trash",
      type: "",
      onClick: function () { deleteComponents();
    }
  },
  location: 'after'
  )
);

}

function buildDeleteGnrlButtons(e) {
  var toolbarItems = e.toolbarOptions.items;

  toolbarItems.unshift({
    //html: " + Resources.strPriceComponent. + ",
    html: 'General',
    location: 'before'
  });

  toolbarItems.push(
  {
    widget: 'dxButton',
    options:
    {
      icon: "trash",
      type: "",
      onClick: function () { deleteGeneral(); }
    }
  },
  location: 'after'
  )
);

}

function deleteComponents() {
  var dataGrid =
$('##priceListsComponent').dxDataGrid("instance");
  var priceCompToBeDeleted =
dataGrid.getSelectedRowsData();
  var deleteCompArray = new Array();

  for (var i = 0; i < priceCompToBeDeleted.length;
i++) {

```

```

deleteCompArray.push(priceCompToBeDeleted[i].PricelistOverheadId);
    }

    var userResult =
    DevExpress.ui.dialog.confirm(Resources.Dialogs.DeletePriceComponent,
    Resources.DialogHeaders.DeletePriceComponent);

    userResult.done(function (dialogResult) {
        if (dialogResult) {
            $.ajax({
                url:
'DataPriceLists/deletePriceComponentInDataGridFor
Popup',
                type: 'Delete',
                traditional: 'true',
                data:
                {
                    'deleteCompsArray': deleteCompArray
                }
            }).done(function () {
                dataGrid.refresh();
            });
        }
    })
}

function deleteGeneral() {
    var dataGrid =
    $("#priceListsGeneral").dxDataGrid("instance");
    var priceCompToBeDeleted =
    dataGrid.getSelectedRowsData();
    var deleteCompArray = new Array();

    for (var i = 0; i < priceCompToBeDeleted.length;
    i++) {

        deleteCompArray.push(priceCompToBeDeleted[i].PricelistOverheadId);
    }

    var userResult =
    DevExpress.ui.dialog.confirm(Resources.Dialogs.DeleteGeneralComponent,
    Resources.DialogHeaders.DeleteGeneralComponent)
    ;

    userResult.done(function (dialogResult) {
        if (dialogResult) {
            $.ajax({
                url:
'DataPriceLists/deletePriceComponentInDataGridFor
Popup',
                type: 'Delete',
                traditional: 'true',
                data:
                {
                    'deleteCompsArray': deleteCompArray
                }
            }).done(function () {
                dataGrid.refresh();
            });
        }
    });
}

```

```

    });
    }
}

function showPriceListColumnChooser(e) {
    if (e !== undefined) {
        var priceListTree =
        $("#pricelists").dxTreeList("instance");
        if (priceListTree !== undefined) {
            priceListTree.showColumnChooser();
        };
    };
}

function refreshPriceListView(e) {
    if (e !== undefined) {
        var priceListTree =
        $("#pricelists").dxTreeList("instance");
        if (priceListTree !== undefined) {
            priceListTree.refresh();
            priceListTree.clearSelection();
        };
    };
}

function clearSelectionOfList() {
    $("#pricelists").dxTreeList("instance").clearSelection();
    ;
}

function onDragChange(e) {
    var treeList =
    e.component,
    visibleRows = treeList.getVisibleRows(),
    sourceNode =
    treeList.getNodeByKey(e.itemData.PricelistId),
    targetNode = visibleRows[e.toIndex].node;

    while (targetNode && targetNode.data) {
        if (targetNode.data.PricelistId ===
        sourceNode.data.PricelistId) {
            e.cancel = true;
            break;
        }
        targetNode = targetNode.parent;
    }
}

function onReorder(e) {
    var treeList =
    e.component,
    visibleRows = treeList.getVisibleRows(),
    targetData = visibleRows[e.toIndex].data;

    if (e.dropInsideItem) {
        $.ajax({
            url: 'PriceLists/changePriceListHierarchy',
            type: 'Post',
            data:
            {

```



```

        'sourceDataPriceListId':
e.itemData.PricelistId,
        'targetDataPriceListId':
targetData.PricelistId
    }
    }).done(function () {
        treeList.refresh();
    });
}

function resizePriceListView() {

$('#pricelists').dxTreeList("instance").option("height", getPriceListViewHeight());
}

function getPriceListViewHeight() {
    var height = window.innerHeight;
    height = height - (40 + 110 + 10);
    return height;
}

function resizePriceViewAndListenForChanges(e) {
    if (e !== undefined) {
        resizePriceListView();
        window.addEventListener("resize",
resizePriceListView);
    };
}

function enableSelectBoxForParentPriceList(e) {
    if (e.value == 1) {

$('#selectBoxForParentPriceList').dxSelectBox("instance").option("disabled", false)
    } else {

$('#selectBoxForParentPriceList').dxSelectBox("instance").reset()

$('#selectBoxForParentPriceList').dxSelectBox("instance").option("disabled", true)

    }
}

function
enableOrDisableToolbarButtonsBasedOnSelection()
{
    var selection =
$('#pricelists').dxTreeList("instance").getSelectedRowKeys();
    if (selection.length == 1) {
        if (selection[0] == 0) {

$('#deletePriceListsDDB').dxButton("instance").option("disabled", true)

$('#duplicatePriceList').dxButton("instance").option("disabled", false)

$('#editPriceListToolbarButton').dxButton("instance").option("disabled", false)

$('#pricesToolbarButton').dxButton("instance").option("disabled", false)
        } else {

$('#deletePriceListsDDB').dxButton("instance").option("disabled", false)

$('#duplicatePriceList').dxButton("instance").option("disabled", false)

$('#editPriceListToolbarButton').dxButton("instance").option("disabled", false)

$('#pricesToolbarButton').dxButton("instance").option("disabled", false)
        }
    } else {
        if (selection.length > 1) {

$('#pricesToolbarButton').dxButton("instance").option("disabled", true)
        } else {

$('#deletePriceListsDDB').dxButton("instance").option("disabled", true)

$('#duplicatePriceList').dxButton("instance").option("disabled", true)

$('#editPriceListToolbarButton').dxButton("instance").option("disabled", true)

$('#pricesToolbarButton').dxButton("instance").option("disabled", true)
        }
    }
}

function clearSelectionOfTree() {

$('#pricelists').dxTreeList("instance").clearSelection();
}

function deletePriceList() {
    var treeList =
$('#pricelists').dxTreeList("instance");
    var priceListData =
treeList.getSelectedRowsData();
    var priceListId = new Array();
    var priceListNames = new Array();
    for (var i = 0; i < priceListData.length; i++) {
        priceListId.push(priceListData[i].PricelistId);
    }

    for (var i = 0; i < priceListData.length; i++) {
        if (i == 0) {

```

```

        var name = priceListData[i].Name
        priceListNames.push("\"" + name + "\"");
    }
    else {
        var name = priceListData[i].Name
        priceListNames.push(" " + "\"" + name +
"\"");
    }
}

if (priceListNames.length > 3) {
    var userResult =
DevExpress.ui.dialog.confirm(Resources.Dialogs.DeletePricelistsWithAmounts + priceListNames.length +
"), Resources.DialogHeaders.DeletePricelists);
} else {
    var userResult =
DevExpress.ui.dialog.confirm(Resources.Dialogs.DeletePricelists + priceListNames + " ?",
Resources.DialogHeaders.DeletePricelists);
}

userResult.done(function (dialogResult) {
    if (dialogResult) {
        $.ajax({
            url: 'PriceLists/Delete',
            type: 'Delete',
            data:
            {
                'keys': priceListId
            },
            success: function () {
                treeList.refresh();
                treeList.clearSelection();
            },
            error: function () {
                var toast =
$("#priceListToast").dxToast("instance");
                toast.option("message",
Resources.Alerts.CantRemoveList);
                toast.option("type", "error");
                $("#priceListToast").dxToast("show");
            }
        })
    }
})
}

function getWidthOfTreeList() {
    var size =
document.getElementById("treeListContainer").offsetWidth
    return size/2;
}

function duplicatePriceList() {
    var treeList =
$("#pricelists").dxTreeList("instance");
    var priceListKeys =
treeList.getSelectedRowKeys();

    $.ajax({
        url: 'PriceLists/DuplicatePriceList',
        type: "Post",
        data: {
            'keys': priceListKeys
        }
    }).done(function (response) {
        treeList.refresh();
        treeList.clearSelection();
        treeList.navigateToRow(response[0]);
        treeList.selectRows(response, false);
    });
}

function priceListsCMenuItemClicked(e) {
    if (e != null && e !== undefined) {
        if (e.itemIndex == 0) {
            expandAll();
        } else if (e.itemIndex == 1) {
            collapsAll();
        };
    }
}

function collapsAll() {
    $pricelists.forEachNode(function (node) {
        $pricelists.collapseRow(node.key);
    });
}

function expandAll() {
    $pricelists.forEachNode(function (node) {
        $pricelists.expandRow(node.key);
    });
}

/*
 *
 * Events for Treelist
 *
 */

function onSelectionChangedPriceLists() {
    //.. other functions for on selection changed

enableOrDisableToolBarButtonsBasedOnSelection();
}

function onInitializedPriceLists(e) {
    //.. other functions for on initialized

    resizePriceViewAndListenForChanges(e);
}

function
clearSelectionOfGroupsAndRefreshGridCmp() {
    $('#ComponentGrid').dxDataGrid("instance").refresh(
);
}

```

```

function getWidthOfFileManagerHalf() {
    var size =
document.getElementById("fileManagerComponent"
).offsetWidth
    return size / 2;
}

function customizeFileManagerGrid(columns) {
    var fileSizeColumn = columns.filter(function (c) {
return c.dataField === "size"; })[0];

    if (fileSizeColumn !== undefined) {
        fileSizeColumn.visibleIndex = 1;
    };

    var nameColumn = columns.filter(function (c) {
return c.dataField === "name"; })[0];

    if (nameColumn !== undefined) {
        nameColumn.caption =
Resources.Strings.strAttachment;
        fileSizeColumn.visibleIndex = 2;
    };

    var modifiedColumn = columns.filter(function (c)
{ return c.dataField === "dateModified"; })[0];
    if (modifiedColumn !== undefined) {
        modifiedColumn.caption =
Resources.Strings.strModified;

columns.splice(columns.indexOf(modifiedColumn),
1);
    };

    columns.push({
        caption: Resources.Strings.strExternalID,
        dataField: "externalID",
    });

    columns.push({
        caption: Resources.Strings.strValidAfter,
        dataField: "validFrom",
        dataType: 'date',
        displayFormat: "dd.MM.yyyy",
        format: "dd.MM.yyyy",
    });

    columns.push({
        caption: Resources.Strings.strValidTo,
        dataField: "validTo",
        dataType: 'date',
        displayFormat: "dd.MM.yyyy",
        format: "dd.MM.yyyy",
    });

    columns.push({
        caption: Resources.Strings.strInternal,
        dataField: "internal",
        dataType: 'boolean',
    });

    columns.push({
        caption: Resources.Strings.strDescription,
        dataField: "description",
    });

    return columns;
}

function uploadPopupCancelButtonClicked() {
    if (document.forms['form'] !== undefined &&
document.forms['form'].reset() !== undefined)
        document.forms['form'].reset();

    document.getElementById("fileNameInput").value
= "";

    closePopup();
}

function resetUploadForm() {
    if (document.forms['form'] !== undefined &&
document.forms['form'].reset() !== undefined)
        document.forms['form'].reset();

    document.getElementById("fileNameInput").value
= "";
    resetUrlInput(undefined);
}

function resetUrlInput(e) {
    if ($('#urlInput').dxTextBox('instance') !==
undefined &&
$('#urlInput').dxTextBox('instance')._options !==
undefined &&
$('#urlInput').dxTextBox('instance')._options.value
!== undefined) {
        $('#urlInput').dxTextBox('instance')._options.value =
"https://";
        $('#urlInput').dxTextBox('instance').repaint();
    };
}

function resetFields() {
    $('#textBoxForPriceListName').dxTextBox("instance
").reset();

    $('#textBoxForPriceListExternalId').dxTextBox("inst
ance").reset();

    $('#textAreaForPriceListDescription').dxTextArea("i
nstance").reset();

    $('#numberBoxForMarkup').dxNumberBox("instance
").reset();
}

```