ЗМІСТ

[Вступ 6](#_Toc441220764)

[1 Основні проблеми розробки сучасних баз даних, аналіз предметної області та постановка задачі курсової роботи. 8](#_Toc441220765)

[1. Актуальність проблем розробки баз даних, основні поняття та визначення. 8](#_Toc441220766)

[1.2 Загальна схема процесу розробки інформаційної системи з застосуванням концепції БД 11](#_Toc441220767)

[1.3 Аналіз предметної області 13](#_Toc441220768)

[1.3.1 Система бізнес-правил 13](#_Toc441220769)

[1.3.2 Глосарій проекту 14](#_Toc441220775)

[1.4 Постановка задачі 14](#_Toc441220776)

[2. Моделювання данних предметної області 15](#_Toc441220777)

[2.1 Розробка концептуальної моделі даних 15](#_Toc441220778)

[2.2 Перетворення концептуальної моделі в логічну модель даних 18](#_Toc441220779)

[3 Програмна реалізація системи 20](#_Toc441220780)

[3.1 Розробка системної програмної архітектури 20](#_Toc441220781)

[3.2 Мотивований вибір СУБД та інструментальних програмних засобів для реалізації запропонованої системної архітектури 22](#_Toc441220782)

[3.2.1 Стислий огляд сучасних типів СУБД та критерії вибору СУБД для реалізації проекту 22](#_Toc441220783)

[3.2.2 Особливості інструментальних засобів програмної реалізації клієнтського додатку та бізнес-логіки системи 27](#_Toc441220784)

[3.3 Розробка прикладного програмного забезпечення 28](#_Toc441220785)

[3.4 Розробка візуального інтерфейсу користувачів системи 29](#_Toc441220786)

[4. Результати застосування розробленої системи 34](#_Toc441220787)

[4.1 Стислі відомості щодо розгортання системи 34](#_Toc441220788)

[4.2.1 Режим редагування даних 35](#_Toc441220789)

[Висновки 36](#_Toc441220790)

Вступ

Організація та структурованість даних дозволяє значно спростити сприйняття інформації та забезпечити адекватні та достовірні відповіді на безлічі запитань до інформаційного простору. Основною метою створення інформаційних систем (ІС) є задоволення інформаційних потреб користувачів шляхом надання необхідної їм інформації на основі збережених даних.

Для представлення певних структурованих знань про інформаційну систему, була винайдена база даних (БД) —  набір відомостей, що зберігаються деяким впорядкованим способом.

Для розроблювача ІС істотним моментом при використанні концепції баз даних є та обставина, що дані стають певним чином організовані, здобувають якусь упорядкованість і внутрішню структуру, а також те, що є деякий набір уніфікованих операцій обробки даних і декларативних засобів подання даних. До таких операцій варто віднести операції "Вставити" (Insert), "Додати" (Add), "Видалити" (Delete) і ряд інших. До декларативних засобів подання даних варто віднести мови визначення даних. Тобто використання даної концепції при створенні ІС припускає наявність мови визначення даних і мови маніпулювання даними, а також правил побудови інтерфейсів програм (додатків) із БД і користувачем.

Основна мета проектування баз даних - це скорочення надмірності збережених даних, а отже, економія обсягу використовуваної пам'яті, зменшення витрат на багаторазові операції відновлення надлишкових копій і усунення можливості виникнення протиріч через збереження в різних місцях зведень про один і той самий об'єкт.

Предметною областю даної курсової роботи є “ Автоматизація продажу квитків до кінотеатру ”. Ця тема є досить популярною в наш час, оскільки, тенденція обробляти всі замовлення в електронному варіанті, пов’язаних з економією часу, набуває все більшого масштабу. Розроблений додаток дозволить користувачеві замовляти квитки, не «підходячи до каси». Розробка даної інформаційної системи дозволить покращити ефективність та зручність покупки квитків.

Завданням курсової роботи є дослідження даної предметної області та створення програми, що працює з БД даної предметною області. Під час створення програми необхідно користуватися CASE-засобами для документування проекту, сучасними програмами та технологіями для його реалізації. Виконання даного проекту дозволить покращити навички програмування та роботи з базами даних.

# 1 ОСНОВНІ ПРОБЛЕМИ РОЗРОБКИ СУЧАСНИХ БАЗ ДАНИХ, АНАЛІЗ ПРЕДМЕТНОЇ ОБЛАСТІ ТА ПОСТАНОВКА ЗАДАЧІ КУРСОВОЇ РОБОТИ.

## Актуальність проблем розробки баз даних, основні поняття та визначення.

На сьогоднішній день застосування баз даних набуло дуже важливого значення в багатьох сферах організацій, які для спрощення своєї роботи застосовують комп'ютерні технології.

Бази даних стали основою інформаційних систем й у корені змінили методи роботи багатьох організацій. Зокрема, за останні роки, завдяки розвитку технологій баз даних, було створено багато дуже потужних і зручних в експлуатації систем. Завдяки цьому системи баз даних стали доступними для широкого кола користувачів.

Актуальність створення обумовлена цілим рядом факторів, серед яких особливо слід виділити інформатизацію, зростання ролі інформаційного забезпечення при прийнятті управлінських рішень, а також багатофакторність і різноманітність соціальних проблем у сільській місцевості, що потребують вирішення. Зазначена база даних дозволяє:

1. об'єднати і синхронізувати технологічні процеси контролю виконання документів, отримання звітів про отримані результати з їх одночасним документуванням в банках даних і сховищах;
2. використовувати сучасні технології з метою оперативного отримання інформації з корпоративних документованих сховищ і її аналітичної обробки для надання особам, які приймають рішення.

Наведемо визначення понять, що будуть широко застосовуватися при написанні даної курсової.

Предметна область (ПрО) - це певна (логічно виділена) частина оточуючого нас реального світу, яка містить в собі деякідані, завдання (бізнес-процеси) по їх обробці, і в якій існують різні користувачі, зацікавлені у використанні результатів пообробці цих даних.

Дані (data) - це будь-який набір символів, які стосуються певної формалізованої системі їх запису (напр., двійковій, десятковій і т.д.), які використовуються для опису фактів або гіпотез в деякій ПрО, і які допускають їх комп'ютерну обробку.

Інформація (information) - це прирощення знань (тобто нових фактів) про деякої ПрО, отриманих на підставі обробки відповідних даних.

Модель даних  — абстрактне представлення реального світу, що відображає тільки ті об'єкти, що безпосередньо стосуються програми. Це, як правило, визначає специфічну групу об'єктів, їх атрибутивне значення і відношення між ними. Вона не залежить від комп'ютерної системи і пов'язана тільки з структурою даних. Це фіксована система понять і правил для представлення даних структури, стану і динаміки проблемної області в базі даних.

БД (database - DB) - це структурована сукупність даних, яка відносяться до деякої ПрО, яка

* організована із застосуванням певної МД;
* створена програмними засобами деякої СУБД і постійно зберігається;
* на носіях пам'яті;
* знаходиться під управлінням цієї СКБД.

Система управління базами даних (СКБД) – це програмні засоби, за допомогою яких можна створювати бази даних, заповнювати їх та працювати з ними.

Основна мета проектування бази даних - це скорочення надмірності збережених даних, а отже, економія об'єму використовуваної пам'яті, зменшення витрат на багаторазові операції відновлення надлишкових копій і усунення можливості виникнення протиріч через збереження в різних місцях відомостей про один й той самий об'єкт. Результатом проектування БД є перетворення опису предметної галузі у внутрішню схему БД.

Недоліками БД є :

* труднощі при обробці інформації з достатньо складними зв'язками;
* складність розуміння змісту звичайним користувачем, залежність від фізичної реалізації.

На початку 70-х років були сформовані теоретичні основи сучасних технологій БД і остаточно сформувався самостійний напрям інформаційних технологій – наука о базах даних. Головною подією цього періоду стало виникнення реляційних баз даних. В основі реляційної моделі лежить поняття відношення. Реляційна модель є простою і зрозумілою, а її фізичне представлення добре реалізується на комп'ютері. До недоліків реляційних моделей можна віднести складність реалізації ієрархічних і мережних зв'язків.

У 70-х роках почали формуватися підходи в БД, які пов'язані з використанням апарата логіки в якості моделі даних. Ці роботи привели до створення дедуктивних БД.

Останнім часом стали з'являтися об'єктно- орієнтовані та об'єктно- реляційні продукти. Ці продукти підтримують деякі технології, що реалізують об'єктно-орієнтований підхід: об'єкти, класи і спадкування реалізовані в структурі баз даних і мовою запитів. Найбільш популярними напрямки досліджень і розробок СУБД є:

1. Реляційні системи
   * стандартизація мови SQL;
   * використання мультипроцесорних організацій;
   * інтеграція.
2. Постреляційні системи
   * бази складних об'єктів, реляційна модель з відмовою від першої нормальної форми;
   * активні бази даних;
   * дедуктивні бази даних;
   * темпоральні бази даних;
   * інтегровані або федеративні системи та мультибази даних;
   * розподілені СКБД.

## 1.2 Загальна схема процесу розробки інформаційної системи з застосуванням концепції БД

У сучасних ІС реалізована 3-х рівнева схема подання даних, яка передбачає:

1. рівень концептуального представлення даних (conceptual data view) – на ньому інформація про деяку ПрО представляється у вигляді її МД без урахування будь-яких особливостей її подальшої комп'ютерної реалізації. Тобто при цьому не враховується, який тип СУБД чи мови програмування будуть використані для реалізації цієї МД, який обсяг збережених даних буде при цьому в ній створюватися і т.д. Це - т.зв. концептуальна МД, або КМД. Для побудови та аналізу КМД використовуються різні графічні схеми (нотації).
2. рівень логічного представлення даних (logical data view) - на цьому рівні отримана спочатку КМД представляється в тому вигляді, в якому вона може бути реалізована засобами мов обробки даних тієї чи іншої СУБД (мов запитів). Це - т.зв. логічна МД або ЛМД. Розрізняють такі види ЛМД: ієрархічна, мережева, реляційна, об'єктно-орієнтована та ін.
3. рівень фізичної реалізації тієї чи іншої МД (physical data view) – на цьому рівні відповідна логічна МД представляється у вигляді фізичних наборів збережених даних, які підтримує та чи інша файлова система відповідної ОС (напр., FAT32, NTFS Windows).

Наочно взаємодія цих 3-х рівнів подання даних в ІС, яка містить у своєму складі СКБД і кілька БД, показує схема:

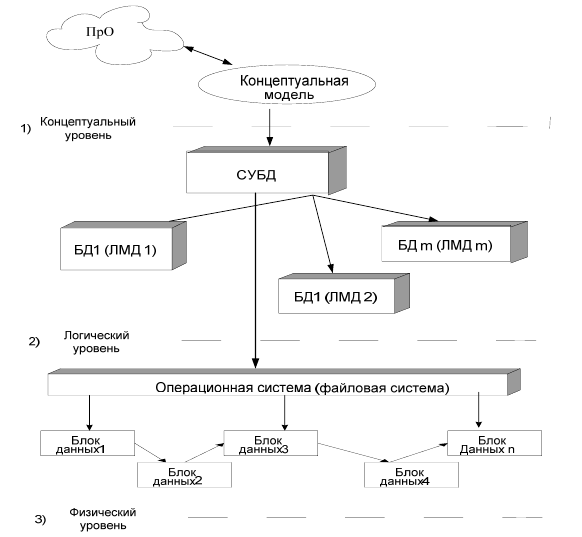


Рисунок 1.1 – Трирівнева схема представлення даних в ІС

При реалізації будь-якого проекту з розробки ІС для побудови КМД необхідно провести аналіз відповідної ПрО, інформація про яку, як правило, представлена у вигляді її текстового опису. У цьому процесі беруть участь різні категорії фахівців

* аналітики;
* замовники проекту;
* експерти з ПрО;
* кінцеві користувачі ІС.

В результаті цього процесу створюються 2 основних документа, на підставі яких і розробляється соответсвующіая КМД, а саме

* глосарій проекту (project glossary) – ДП;
* система бізнес-правил (business rules system) – СБП

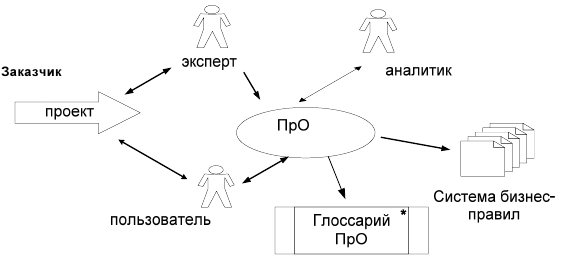


Рисунок 1.2 – Схема процесу аналізу ПрО

## 1.3 Аналіз предметної області

### 1.3.1 Система бізнес-правил

Аналізуємо предметну область проекту “ Автоматизація покупки квитків до кінотеатру ” та складаємо бізнес-правила.

### Кожен квиток характеризується наступними параметрами: унікальним ключем, сеансом, залом, назвою фільму та датою купівлі.

### Кожен сеанс характеризується датою, часом і вартістю обумовленою типом.

### Зал характеризується кількістю рядів і місць.

### Кожен фільм характеризується унікальним ключем, назвою фільму, тривалістю, режисером, роком виходу, жанром, обмеженням по віку, та описом.

### Вартість квитка може варіюватися лише від типу залу.

1. В одному залі можуть показувати кілька фільмів (у різний час) і один фільм може бути показаним в різних залах.
2. Один фільм може мати декілька жанрів, і в одному жанрі можуть знімати безліч фільмів.
3. За допомогою одного квитка можна відвідати тільки один сеанс.

### Глосарій проекту

* 1. Фільм — об'єкт перегляду, заради якого відвідувачі приходять в кінотеатр.
  2. Сеанс — період безперервного показу кінофільму у встановлений час і дату.
  3. Зал — приміщення, призначене для розташування глядачів і показу фільму.
  4. Глядач — фізична особа, що переглядає фільм у кінотеатрі та є користувачем даного ПЗ.
  5. Квиток —  документ, що засвідчує право перегляду фільму в кінотеатрі.
  6. Жанр —  рід творів у галузі кіномистецтва, що характеризує фільм.

## 1.4 Постановка задачі

Під час виконання даної курсової роботи потрібно створити програмну систему для предметної області “ Автоматизація покупки квитків до кінотеатру ”, послідовно застосовуючи принцип об’єктно-орієнтованого аналізу, концептуального і реляційного моделювання даних, а також використовуючи програмні засоби СУБД і технології програмування застосувань БД в операційній системі MS Windows.

Необхідно реалізувати наступні можливості програми:

* підключення до БД та використання її інформації;
* надання користувачу зручного інтерфейсу редагування даних БД;
* відображення всіх таблиць ІС за БД та їх редагування;
* забезпечити можливість покупки квитка;

Інтерфейс користувача передбачає створення вікна з необхідними кнопками, таблицями для ведення та редагування даних, рядками введення скалярних даних.

До основних задач курсової роботи відносяться:

* дослідження предметної області;
* отримання практичних навичок у об’єктно-орієнтованому аналізі, застосування CASE засобів;
* вивчення технології програмування застосувань з доступом до БД, програмних засобів СКБД;
* створення БД та застосування з GUI для роботи з ним.

# МОДЕЛЮВАННЯ ДАННИХ ПРЕДМЕТНОЇ ОБЛАСТІ

## 2.1 Розробка концептуальної моделі даних

Концептуальна модель – це модель, представлена множиною понять і зв’язків між ними, що визначають змістовну структури розглянутої предметної галузі. Її розробляють перед створенням БД. Зазвичай при цьому використовують спеціальні нотації. Найбільш поширеними є канонічна ER нотація Пітера Чена та UML-діаграма класів. Наведемо концептуальну модель предметної області

“Автоматизація покупки квитків до кінотеатру” в UML-нотації.

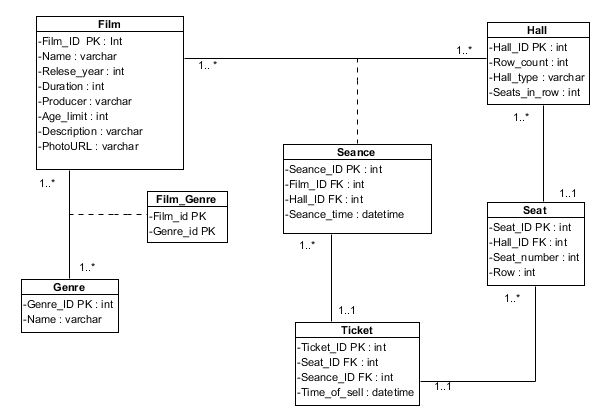


Рисунок 2.1 – Концептуальна модель даних в UML-нотації

Усі сутності надані на діаграмі необхідні для генерування квитка.

Сутність Film відображає інформацію про доступні для перегляду фільми, а також його характеристику, що допоможе при виборі фільму.

Сутність Genre – жанр фільму.

Cутність Film\_Genre є ассоціативною сутністю між Film та Genre.

Hall – зали, відображає інформацію про кількість рядів в залі, кількість місць в ряді, тип залу, та ціну на сеанс.

Сутність Séance – є ідентифікуючою сутністю між фільмами та залами.

Атрибут Seamce\_time визначає дату та час сеансу.

Сутність Seat – відображає місце та ряд глядача при перегляді фільму.

Сутність Ticket – узагальнює вибрані дані користувача, та містить поле Date\_of\_sell – час продажу квитка.

AllFusion ERwin Data Modeler (ERwin) дозволяє наочно відображати складні структури даних. Зручна у використанні графічна середовище AllFusion ERwin Data Modeler спрощує розробку баз даних і автоматизує багато трудомістких завдань, зменшуючи терміни створення високоякісних та високопродуктивних транзакційних баз даних і сховищ даних. Дане рішення покращує комунікацію організації, забезпечуючи спільну роботу адміністраторів і розробників баз даних, багаторазове використання моделі, а також наочне представлення комплексних активів даних в зручному для розуміння і обслуговування форматі.

Реляційна модель даних (РМД) - логічна модель даних, прикладна теорія побудови баз даних, яка є додатком до завдань обробки даних таких розділів математики як теорії множин і логіка першого порядку. На основі реляційної моделі даних будуються реляційні бази даних.

Реляційна модель даних включає наступні компоненти:

Структурний аспект (складова) - дані в базі даних представляють собою набір відносин.

Аспект (складова) цілісності - відносини (таблиці) відповідають певним умовам цілісності. РМД підтримує декларативні обмеження цілісності рівня домену (типу даних), рівня відносини і рівня бази даних.

Аспект (складова) обробки (маніпулювання) - РМД підтримує оператори маніпулювання відносинами (реляційна алгебра, реляційне числення).

Для кращого розуміння РМД слід відзначити три важливих обставини:

* модель є логічною, тобто відносини є логічними (абстрактними), а не фізичними (збереженими) структурами;
* для реляційних баз даних вірний інформаційний принцип: все інформаційне наповнення бази даних представлено одним і тільки одним способом, а саме – явним завданням значень атрибутів у кортежі відносин; зокрема, немає ніяких покажчиків (адрес), що пов'язують одне значення з іншим; наявність реляційної алгебри дозволяє реалізувати декларативне програмування і декларативний опис обмежень цілісності, на додаток до навігаційного (процедурним) програмування і процедурної перевірці умов.

Перед тим як приступити до розробки інформаційної системи всі необхідні елементи й структури даних повинні бути об'єднані у відповідну модель даних. Така модель даних являє собою інформаційну модель певної предметної області й служить для рішення наступних завдань:

- створення відповідної БД;

- модель даних виступає засобом спілкування різних категорій осіб, що приймають участь у створенні інформаційної системи (розроблювачі, експерти, по предметній області, прикладні програмісти і. т. д.);

- наявність моделі даних (її реалізація й документація) дозволяє надалі адаптувати існуючу інформаційну систему до змін, що виникають у предметній області.

З погляду класичної теорії моделювання даних у СКБД логічна модель системи ERwin являє собою так званий концептуальний рівень опису даних (Conceptual level). Концептуальний рівень являє собою абстрактний погляд на дані. На цьому рівні дані представляються так, як вони виглядають у реальному світі. Концептуальна модель даних є універсальною й ніяк не пов'язана з конкретною реалізацією СКБД.

Нижче наведено концептуальну модель предметної області курсової роботи, вона має не має сутності subject\_lector, оскільки її автоматично створить ERWin при переході до логічної моделі даних.

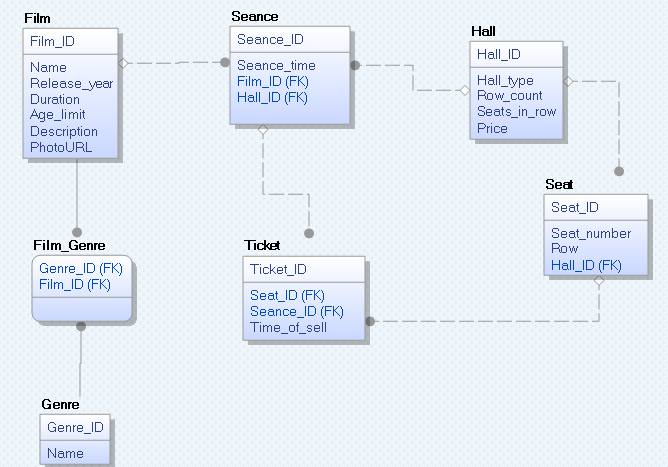


Рисунок 2.2 – Логічний рівень КМД у Case системі ERWin

## 2.2 Перетворення концептуальної моделі в логічну модель даних

Фізична модель даних залежить від конкретної СКБД. У фізичній моделі втримується інформація про всі об'єкти. В фізичній моделі важливо описати всю інформацію про конкретні об'єкти - таблицях, колонках, індексах, процедурах і т.д. Компонентами фізичної моделі даних є таблиці, стовпці і відносини. Сутності логічної моделі, ймовірно, стануть таблицями у фізичній моделі. Логічні атрибути стануть стовпцями. Логічні відносини стануть обмеженнями цілісності зв’язків.

З приводу атрибутів КМД даної курсової роботи не зазнала змін після перетворення у логічний рівень. Були вибрані типи даних, що найбільше співвідносяться з інформацією стовпців. В ході аналізу визначено, що модель даних знаходиться в 3-й нормальній формі, оскільки не має транзитивних залежностей від потенційних ключів.

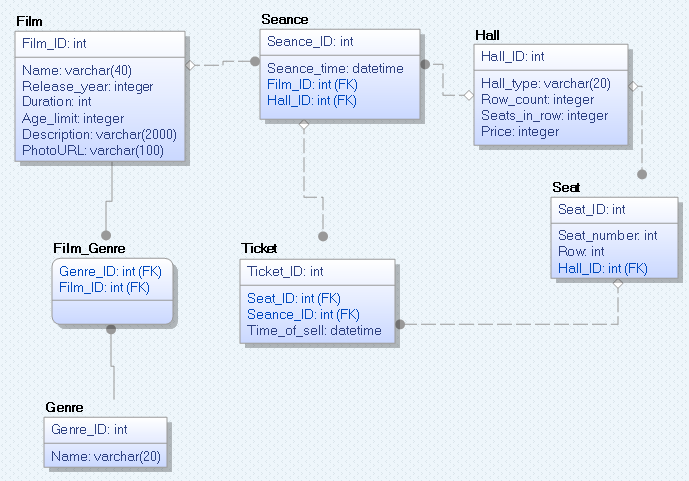


Рисунок 2.3 – Логічна модель даних

# 3 ПРОГРАМНА РЕАЛІЗАЦІЯ СИСТЕМИ

## 3.1 Розробка системної програмної архітектури

Архітектура системи – принципова організація системи, втілена в її елементах, їх взаєминах один з одним і з середовищем, а також принципи, направляючі її проектування та еволюцію.

Логічна архітектура підтримує функціонування системи протягом усього її життєвого циклу на логічному рівні. Вона складається з набору пов'язаних технічних концепцій і принципів. Логічна архітектура представляється за допомогою методів, що відповідають тематичними групами описів, і як мінімум, включає в себе функціональну архітектуру, поведінкову архітектуру і тимчасову архітектуру.

Функціональна архітектура. Функціональна архітектура являє собою набір функцій та їх підфункцій, що визначають перетворення, здійснювані системою при виконанні свого призначення.

Поведінкова архітектура. Поведінкова архітектура - угода про функції та їх підфункція, а також інтерфейсах (входи і виходи), які визначають послідовність виконання, умови для управління або потоку даних, рівень продуктивності, необхідний для задоволення системних вимог. Поведінкова архітектура може бути описана як сукупність взаємопов'язаних сценаріїв, функцій та / або експлуатаційних режимів.

Тимчасова архітектура. Тимчасова архітектура є класифікацією функцій системи, яка отримана відповідно до рівня частоти її виконання. Тимчасова архітектура включає в себе визначення синхронних і асинхронних аспектів функцій. Моніторинг рішень, який відбувається всередині системи, слідує тієї ж тимчасової класифікації

При виконанні курсової роботи визначаємо архітектуру за розміщенням 3-х фізичних та логічних компонентів програмного комплексу: рівня доступу та зберігання даних (data access layer), рівня обробки даних (business logic layer), рівня представлення даних (data presentation layer). Рівнем зберігання даних за завданням курсової роботи буде БД. Отже СА може бути наступних типів:

* 2-х рівнева клієнт-серверна СА з “товстим” клієнтом і сервером БД (бізнес логіка зосереджена на клієнті);
* 2-х рівнева клієнт-серверна СА з “тонким” клієнтом і сервером БД (бізнес логіка зосереджена на сервері);
* 3-х рівнева клієнт-серверна СА з “тонким” клієнтом і сервером застосунком (для бізнес-логіки виділено окремий сервер);

Для того, щоб визначити системну програмну архітектуру розроблено UML-діаграму основних варіантів використання.

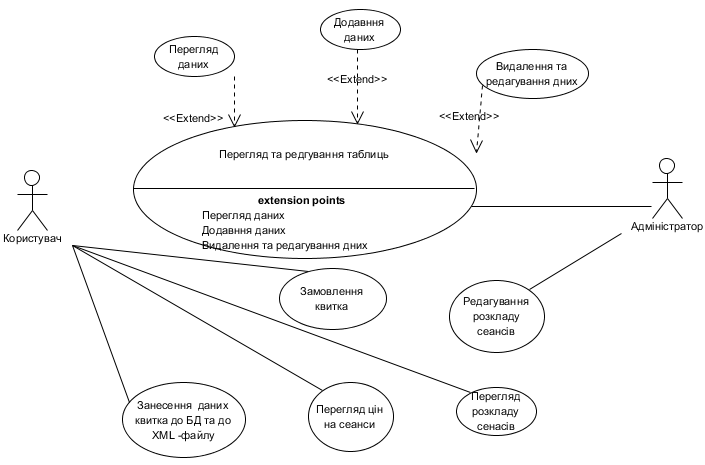


Рисунок 3.1 – Діаграма використання

Основними задачами системи є робота з таблицями БД: внесення, змінення, видалення інформації. Навіть складання розкладу та перегляд є роботою з інформацією БД. Імпорт розкладу в формат зручний для друку є роботою рівня клієнта архітектури.

Оскільки предметна область не передбачає великої кількості даних, складних запитів, багатьох підключень до БД та володіє потенціалом для збільшення функціональності - система не потребує потужного апаратного забезпечення та задовільним вибором СА є “товстий клієнт”.

## 3.2 Мотивований вибір СКБД та інструментальних програмних засобів для реалізації запропонованої системної архітектури

Для розробки даної ІС була вибрана системна архітектура «Товстий клієнт»,

оскільки сервер в цьому випадку є лише сховищем даних, а вся робота з обробки та подання цих даних переноситься на машину клієнта.

Діаграма компонентів знаходиться на рисунку 3.2

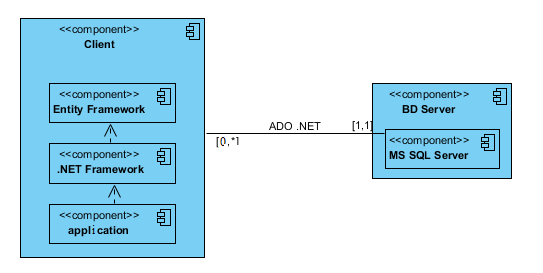


Рисунок 3.2 - Діаграма розгортання ПЗ

### 3.2.1 Стислий огляд сучасних типів СКБД та критерії вибору СКБД для реалізації проекту

Класифікація СКБД.

За моделлю даних

* ієрархічні;
* мережеві;
* реляційні;
* об'єктно-орієнтовані;
* об'єктно-реляційні.

При виборі СКБД слід враховувати вимоги до апаратного забезпечення та системного програмного забезпечення, необхідного для установки й роботи СКБД. СКБД, установка й експлуатація яких може викликати проблеми, обирати не рекомендується. Обрана СКБД повинна бути орієнтована на підтримку архітектури «клієнт-сервер». До таких СКБД, зокрема, відносяться: Oracle, Microsoft SQL Server, InterBase, Sybase Adaptive Server Anywhere, MySQL і т.п.

Програмно-апаратні платформи, використані при розробці та застосуванні баз даних, справляють істотний вплив на ефективність їх функціонування. Проблеми вибору апаратно-програмних платформ для баз даних можна розділити на наступні складові:

1. вибір СКБД;
2. вибір апаратних засобів обробки баз даних.

Підходи до вибору СКБД. Основним принципом вибору СКБД логічно вважати визначення програмного продукту, в найбільшій мірі відповідного пропонованим вимогам. Практично вирішити це завдання не дуже просто. По-перше, до СКБД пред'являється велика кількість вимог і, головне, вони з часом змінюються. По-друге, СКБД мають велике число параметрів, що ускладнює їх порівняння.

Розглянемо технологію оцінки характеристик СКБД і визначення ступеня їх відповідності пропонованим вимогам. Вибір СКБД найкраще робити з позицій особи, приймає рішення при неповній чи суперечливої інформації. Програмні продукти зазвичай супроводжує наступна інформація:

1. відомості розробників і рекламна інформація продавців;
2. інформація кінцевих користувачів, розробників і адміністраторів, які мають досвід роботи з продуктом;
3. інформація аналітиків та експертів.

Процедуру вибору СКБД зручно проводити в три етапи. Спочатку на якісному рівні оцінити пропоновані програмні продукти на предмет придатності, звузивши область вибору. Потім оцінити технічні характеристики відібраних систем більш детально. І, нарешті, оцінити продуктивність залишилися продуктів для прийняття остаточного рішення.

До числа основних показників придатності програмних продуктів можна віднести наступні:

1. види програмного продукту;
2. категорії користувачів;
3. зручність і простота використання;
4. модель представлення даних;
5. якість засобів розробки;
6. якість засобів захисту та контролю коректності бази даних;
7. якість комунікаційних засобів;
8. фірма-розробник;

У конкретній організації є своя розкладка показників на основні та додаткові. Пошук потрібного продукту рекомендується почати з вивчення потреб і можливостей. Важливо визначитися, для чого потрібен пакет: для розробки прикладних систем професійними програмістами або для роботи кінцевих користувачів в інтерактивному режимі, щось інше або кілька цілей. Визначальними параметрами на першому етапі відбору є вид програмного продукту і категорія користувачів.

Microsoft Access - це найпопулярніша сьогодні настільна система управління базами даних. Її успіх багато в чому пов'язаний з включенням її в сімейство Microsoft Office. Цей програмний продукт розрахований як на початківця, так і на кваліфікованого користувача.

СКБД Access для роботи з даними використовує процесор баз даних Microsoft Jet, об'єкти доступу до даних і засобу швидкої побудови інтерфейсу - Конструктор форм. Для отримання роздруківок використовуються Конструктори звітів.

Microsoft SQL Server призначений виключно для підтримки систем, що працюють у середовищі клієнт-сервер. Він підтримує широкий спектр засобів розробки і максимально простий в інтеграції з додатками, що працюють на ПК .

SQL Server може тиражувати інформацію в БД інших форматів включаючи Oracle, DB2, Sybase, Microsoft Access і інші СКБД (при наявності ODBC драйвера, що відповідає визначеним вимогам).

SQL Server дозволяє розробникам управлятися з даними будь-яких типів - від традиційних до географічних. Це відкриває дорогу до створення додатків нового покоління з урахуванням інформації про розташування і можливістю управління документами. SQL Server здатний формувати звіти і виконувати аналіз будь-якого обсягу та складності, одночасно полегшуючи користувачам доступ до даних за рахунок більш тісної інтеграції з Microsoft Office.

У SQL Server поліпшені можливості дзеркалювання БД, що вперше з'явилися в SQL Server 2005. Збої в обладнанні здатні пошкодити сторінки даних БД. За допомогою даної СКБД головний і дзеркальний комп'ютери можуть прозоро для користувачів і додатків відновлювати працездатність після помилок. Для зеркалирования БД потрібно обмін даними між усіма учасниками. SQL Server забезпечує стиск вихідного потоку даних, оптимізуючи продуктивність і скорочуючи навантаження на мережу, пов'язану з дзеркалюванням БД. Стиснення даних дозволяє більш ефективно зберігати їх і зменшити витрати на накопичувальні пристрої. При цьому істотно знижуються робочі навантаження, підвищується продуктивність, а обсяг введення -виведення, наприклад у сховищах даних, збільшується.

MySQL є ідеальним рішенням для малих і середніх додатків. Выхыдны файли сервера компілюються на безлічі платформ. Найбільш повно можливості сервера проявляються на Unix-серверах, де є підтримка багатопоточності, що дає значний приріст продуктивності.

На поточний момент MySQL усе ще в стадії розробки, хоча версії 3.22 повністю працездатні.

Короткий перелік можливостей MySQL:

* підтримується необмежена кількість користувачів, що одночасно працюють з базою даних;
* кількість рядків у таблицях може досягати 50 млн;
* швидке виконання команд. Можливо MySQL найшвидший сервер з існуючих;
* проста і ефективна система безпеки.

Перші версії MySQL мали низьку фукціональність, але оскільки розробник удосконалював СКБД на замовлення ліцензованих користувачів – сучасні версії MySQL такого недоліку не мають.

Oracle Database – СКБД від компанії Oracle мають багато різних версій для різного бюджету і потреб.

Розглянемо одну з версій, що може бути використана в даній роботі. Oracle Database 11g Express Edition (Oracle Database XE) - це СУБД початкового рівня, заснована на програмному коді СУБД Oracle 11g Database Release 2. Дана СУБД безкоштовна для розробки, розгортання та продажу, швидко скачується і проста в адмініструванні.

Oracle Database XE - чудовий стартовий СКБД для:

* розробників, що працюють з PHP, Java, .NET, XML і додатками з відкритим кодом;
* адміністраторів баз даних, яким потрібна невелика безкоштовна СКБД;
* незалежних вендорів, які хочуть безкоштовну СКБД стартового рівня для продажу без зайвих витрат;
* навчальних закладів і студентів, яким необхідна безкоштовна СКБД.

Для виконання даної курсової роботи використається СКБД реляційного типу. Важливими критеріями для її вибору є надійність, швидкість, безкоштовність, наявність безкоштовної локалізованної програми зі зручним інтерфейсом.

### 3.2.2 Особливості інструментальних засобів програмної реалізації клієнтського додатку та бізнес-логіки системи

Наведемо інструментальні засоби, що можуть бути використані для реалізації клієнтського додатку.

Microsoft Visual Studio - лінійка продуктів компанії Microsoft, що включають інтегроване середовище розробки програмного забезпечення і ряд інших інструментальних засобів. Дані продукти дозволяють розробляти як консольні додатки, так і додатки з графічним інтерфейсом, в тому числі з підтримкою технології Windows Forms, а також веб-сайти, веб-додатки, веб-служби як в рідному, так і в керованому кодах для всіх платформ, підтримуваних Windows, Windows Mobile, Windows CE, .NET Framework, Xbox, Windows Phone .NET Compact Framework і Silverlight.

Visual Studio включає в себе редактор вихідного коду з підтримкою технології IntelliSense і можливістю найпростішого рефакторінга коду. Вбудований відладчик може працювати як відладчик рівня вихідного коду, так і як відладчик машинного рівня. Решта вбудовуються інструменти включають в себе редактор форм для спрощення створення графічного інтерфейсу додатку, веб-редактор, дизайнер класів і дизайнер схеми бази даних. Visual Studio дозволяє створювати і підключати сторонні додатки (плагіни) для розширення функціональності практично на кожному рівні, включаючи додавання підтримки систем контролю версій вихідного коду (як наприклад, Subversion і Visual SourceSafe), додавання нових наборів інструментів (наприклад, для редагування і візуального проектування коду на предметно-орієнтованих мовах програмування або інструментів для інших аспектів процесу розробки програмного забезпечення (наприклад, клієнт Team Explorer для роботи з Team Foundation Server).

Завдяки програмі Microsoft Dreamspark існує можливість отримати останні версії даного програмного продукту безкоштовно.

Курсова робота згідно з завданням повинна бути створена під операційну систему MS Windows. Платформа .NET та мова С# ефективніше працюють на даній ОС замість Java.

## 3.3 Розробка прикладного програмного забезпечення

Для доступу к даним БД можна обрати декілька технологій, враховуючи раніше вибрану середу розробки та СКБД:

* ADO;
* ADO.NET;
* NHibernate.

ADO (від англ. ActiveX Data Objects – “об'єкти даних ActiveX”) - інтерфейс програмування додатків для доступу до даних, розроблений компанією Microsoft (MS Access, MS SQL Server) і заснований на технології компонентів ActiveX. ADO дозволяє представляти дані з різноманітних джерел (реляційних баз даних, текстових файлів і т. д.) в об'єктно-орієнтованому вигляді.

Останньою версією ADO є версія 2.8. В рамках платформи Microsoft .NET інтерфейс ADO замінений ADO.NET.

ADO.NET - технологія, що надає доступ до даних для додатків, заснованих на Microsoft .NET. Є не розвитком більш ранньої технології ADO, а самостійної технологією, частиною фреймворку .NET.В відміну від класичної ADO, яка була в основному призначена для тісно пов'язаних клієнт-серверних систем, ADO.NET більше націлена на автономну роботу за допомогою об'єктів DataSet. Ці типи представляють локальні копії будь-якої кількості взаємопов'язаних таблиць даних, кожна з яких містить набір рядків і стовпців. Об'єкти DataSet дозволяють викликає збірці (на зразок веб-сторінки або програми, що виконується на настільному комп'ютері) працювати з вмістом DataSet, змінювати його, не вимагаючи підключення до джерела даних, і відправляти назад блоки змінених даних для обробки за допомогою відповідного адаптера даних. Але, мабуть , саме фундаментальна відмінність між класичною ADO і ADO.NET полягає у тому, що ADO.NET є керованою кодової бібліотекою, і, значить, підкоряється тим же правилам, що і будь-яка керована бібліотека. Типи, складові ADO.NET, використовують протокол управління пам'яттю CLR, належать до тієї ж системи типів (класи, інтерфейси, перерахування, структури і делегати), і доступ до них можливий за допомогою будь-якої мови .NET. Класи ADO.NET знаходяться в збірці System.Data.dll.

NHibernate - ORM-рішення для платформи Microsoft .NET, портоване з Java. Це безкоштовна бібліотека з відкритим кодом, поширюється під ліцензією GNU Lesser General Public License.

NHibernate дозволяє відображати об'єкти бізнес-логіки на реляційну базу даних. По заданому XML-опису сутностей і зв'язків NHibernate автоматично створює SQL-запити для завантаження і збереження об'єктів.

Очевидним вибором для даної курсової роботи є ADO.NET. Ця технологія найбільш співвідноситься з логікою “товстого клієнту” (локальна робота з даними), розроблена під платформу .NET та у MS Visual Studio 2010 є простий інтерфейс для її використання.

Рівні бізнес логіки, окрім тригерів, та представлення даних зосереджено на клієнті. Застосування буде виконуватися у середі CLR .NET використовувати технологію ADO.NET для доступу до даних БД MySQL, яку розгорнуто на сервері.

## 3.4 Розробка візуального інтерфейсу користувачів системи

Візуальний інтерфейс сформований стандартними засобами бібліотеки Win Forms. Загальну схему виконання програми наведено нижче:

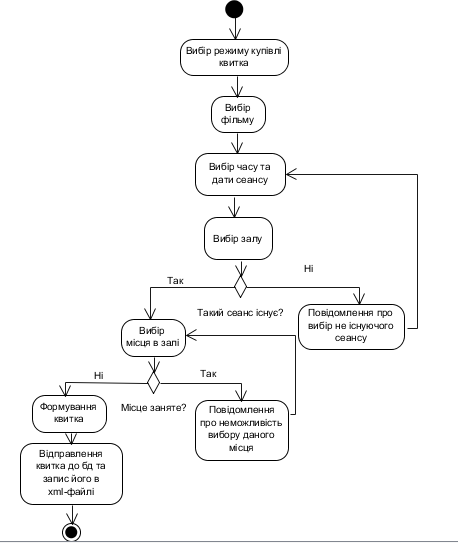


Рисунок 3.4 – Діаграма діяльності купівлі квитка користувачем

При вході до програми користувачеві пропонується купити квиток або ж для редагування таблиць перейти до режиму адміністратора.

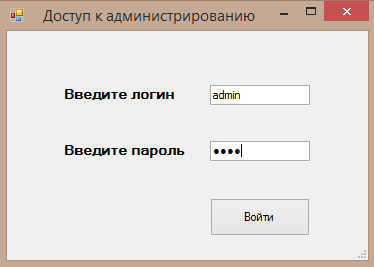


Рисунок 3.5 – Вигляд вікна для входу в режим адміністратора

У режимі редагування можна вибрати таблицю для редагування та за допомогою стандартних елементів інтерфейсу DataGridView здійснити редагування. При натисненні на кнопку "Зберегти". Дані відсилаються до БД. Якщо вони коректні вони заносяться до таблиці.

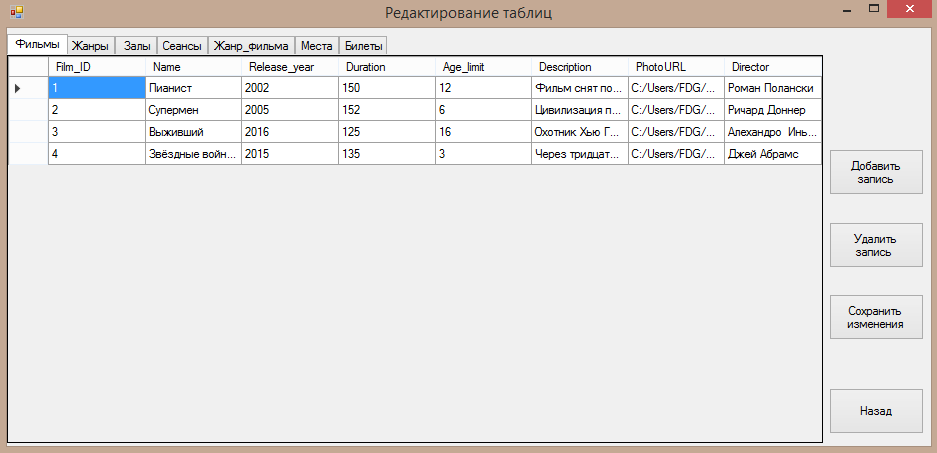


Рисунок 3.6 – Вікно редагування таблиць

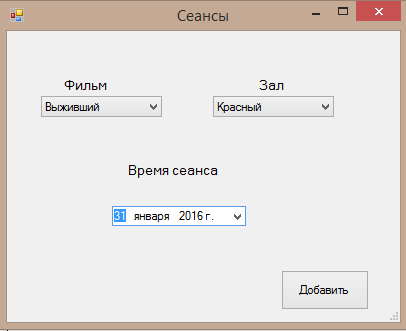


Рисунок 3.7 – Вікно додавання сеансів у розклад

При виборі режиму покупки квитка користувачеві пропонується вибрати фільм, для цього створені елементи що містять головну інформацією про фільм.

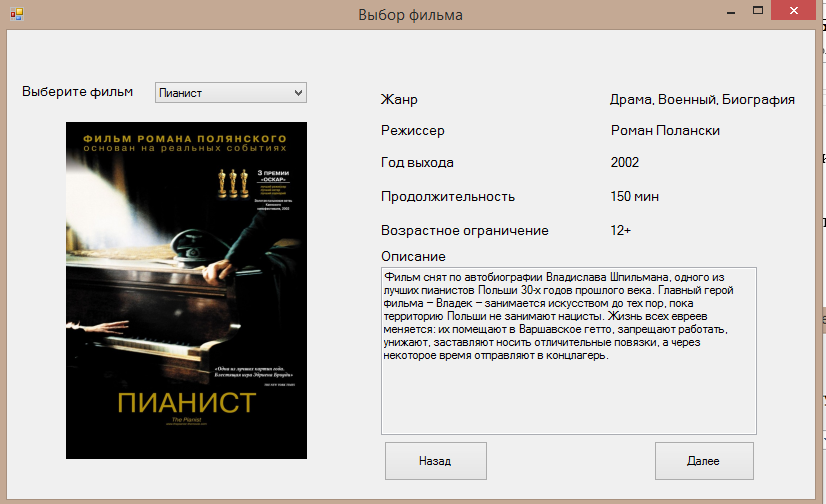


Рисунок 3.7 – Вікно вибору фільму

Далі користувачеві пропонується вибрати тип залу, час та дату сеансу, місце та ряд.

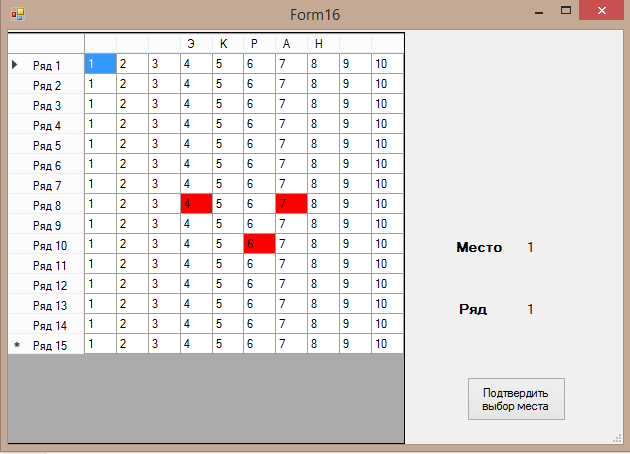


Рисунок 3.7 – Вікно вибору місця на сеанс

Згенерований квиток зображений на рисунку 3.8. Користувач має можливість зберегти його до xml-файлу. Квиток також автоматично добавляється до БД.

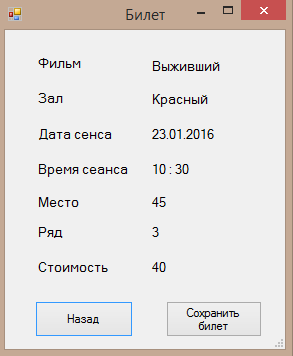


Рисунок 3.8 – Вікно вибору місця, залу, дати та типу сеансу,

Для зручності купівлі квитка користувачеві також пропонується переглянути ціни на сеанси

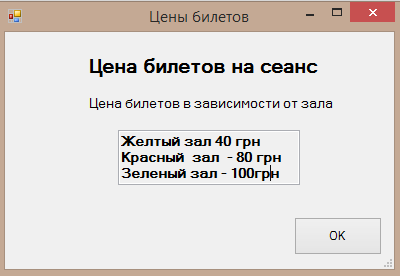


Рисунок 3.9 – Вікно перегляду цін на сеанси

# РЕЗУЛЬТАТИ ЗАСТОСУВАННЯ РОЗРОБЛЕНОЇ СИСТЕМИ

## 4.1 Стислі відомості щодо розгортання системи

Оскільки бібліотека графічного інтерфейсу Windows Forms, що була застосована в проекті виконується у середовище .NET та останньою версією .NET, що вносила зміни до Win Forms, є .NET 3.5, то мінімальними системними вимогами до клієнтського застосунку будуть вимоги до .NET 3.5.

Таблиця 3.1 – Вимоги .NET 3.5 до обладнання

|  |  |  |
| --- | --- | --- |
| Процесор | | 400 МГц |
| ОЗП | | 96 МБ |
| Об’єм пам’яті жорсткого диску | 32 розрядна ОС | 280 МБ |
| 64 розрядна ОС | 610 МБ |

Таблиця 3.2 – Підтримувані .NET 3.5 клієнтські ОС

|  |  |  |
| --- | --- | --- |
| Операційна система | Підтримувані випуски | Додаткові відомості |
| Windows 8.1 | 32- та 64-розрядна | Включає .NET Framework 4.5.1 |
| Windows 8 | 32- та 64-розрядна | Включає .NET Framework 4.5 |
| Windows 7 SP1 | 32- та 64-розрядна | - |
| Windows Vista SP2 | 32- та 64-розрядна | - |

Вимоги до серверної машини на яку потрібно встановити сервер MySQL. При написанні програмного забезпечення застосовувався MySQL 5.1 ОС родини Windows на які він може бути встановлений: Microsoft Windows XP, Microsoft Windows 2003 Server, Microsoft Windows Vista, Microsoft Windows 2008 Server, Microsoft Windows 7, як на 64, так і на 32-розрядні. Щодо вимог до апаратної частини, якщо припустити, що сервер бути використовувати тільки завуч, або максимум 5 підключень, то 512 МБ оперативної пам’яті, процесору 2.4 ГГц та 35 Гб дискового простору буде достатньо.

Для розгортання системи потрібно запустити MySQL сервер. Скопіювати дам бази даних, якщо він існує, в іншому разі БД можна заповнити з клієнтського застосунку. Скопіювати клієнтський застосунок на комп’ютер клієнта. У файлі з розширенням config та назвою програми записати користувача, сервер, пароль до БД та назви БД. Також для коректної роботи програми на машині клієнта повинні бути встановлені: платформа .NET та MS Excel для функції друкування розкладу.

### 4.2.1 Режим редагування даних

В даному режимі здійснюється редагування даних таблиць БД на основі яких створюється розклад. Інтерфейс передбачає зрозумілі для користувача вікна вибору таблиць та редагування даних .

**4.3 Результати тестування та рекомендації щодо удосконалення розробленої системи**

При тестуванні основною проблемою виявилося обробка виняткових ситуацій викликаних тригерами БД та обмеженнями зв'язків. Потрібно було надати користувачу правильну інформацію про проблему та по різному у програмі оброблялись ці два типа винятків. При більш детальному вивчені бібліотеки win forms та технології ado.net entity model ця проблема була вирішена.

Також проблемою було оновлення даних після некоректного вводу у таблицю. Некоректні дані не приймалися базою даних, але залишались у локальній копії таблиці, заважаючи створювати будь-які інші транзакції. Ця проблема була вирішена перезавантаженням таблиці.

Розроблену систему можна розвивати у декількох напрямках. Розширити її функціональність веденням розкладу сеансу. Для цього потрібно додати таблиці, що відповідатимуть розширеній предметній області до БД, забезпечити підтримання бізнес правил новими тригерами та збереженими процедурами. Також у такому випадку доведеться ввести систему паролів та покращити апаратне забезпечення серверу, оскільки система потребуватиме більше підключень та зберігатиме більше даних.

Також можна додати до системи можливості зберігання таблиць у інші формати крім XML, наприклад Excel - формат.

# Висновки

В ході написання курсової роботи були отримані практичні навички аналізу предметної області. Складання бізнес правил та глосарію проекту. Побудування концептуальних схем даних та на основі них створення БД. Було ознайомлено з класифікацією та найбільш поширеними СУБД та технологіями зв’язку з ними клієнтського програмного забезпечення. Покращені навички ведення проектної документації за допомогою CASE-засобів.

Було розроблене програмне забезпечення автоматизації продажу квитка із застосуванням бібліотеки графічного інтерфейсу windows forms. Для зв’язку з БД використовується технологія доступа до даних Entity Framework. Розроблене програмне забезпечення дозволяє редагувати дані БД за допомогою засобів графічного інтерфейсу користувача. Генерувати квиток та зберігати його.

# Список джерел інформації

1. Пасічник В.В., Резниченко В.А. Організація баз даних та знань (підручник). – К.: BHV, 2006.
2. Коноллі Т. та ін. Бази даних: проектування, реалізація і супровід. - М. «Вільямс», 2001. (2003) Хомоненко А.А., Циганков В.М., Мальцев М.Т. Бази даних. - "Корона", СПб, 2002. (2003).
3. Чекалов А.П. Бази даних: від проектування до розробки додатків. - СПб. : БХВ-Петербург, 2003.
4. Batini, C., Ceri, S., Navathe, Sh. Conceptual Database Design. – Benjamin / Cummings Publishing Company, Inc., 1992, - 496 p.p. (it is available at the ASU- Dept library).
5. Мюллер, Р. Бази даних та UML. - М .: "Лорі", 2002.
6. Когаловскій М.Р. Енциклопедія технологій баз даних. - М., «Фінанси і статистика», 2002.
7. Архітектури, моделі і технології програмного забезпечення інформаційно-керуючих систем / Ткачук Н.В., Шеховцов В.А., Кукленко Д.В., Сокіл В.Є. - Харків: НТУ «ХПІ», 2005.
8. Маклаков С.В. BpWin і ErWin: CASE-засобу розробки інформаційних систем. - М .: «Діалог-МІФІ», 2000.
9. Методичні вказівки до проведення лабораторних занять з курсу "Організація баз даних і знань ", частина 1:" Моделювання даних у системі ERwin "для студентів напрями підготовки фахівців 0804 - "Комп'ютерні науки" / Складали Н.В. Ткачук, Є.В. Борщ Є.Ю. Климова, А.В. Костанян, Е.А. Матат. - Харків: НТУ «ХПІ», 2007.
10. Методичні вказівки до проведення лабораторних занять з курсу "Організація баз даних і знань ", частина 2:" Технологія роботи з СУБД MySQL "для студентів напрями підготовки фахівців 0804 - "Комп'ютерні науки" / Складали Н.В. Ткачук, Є.В. Борщ Є.Ю. Климова, А.В. Костанян, Е.А. Матат. - Харків: НТУ «ХПІ», 2007.