ЗМІСТ

[Перелік позначень та скорочень 6](#_Toc471844210)

[Вступ 7](#_Toc471844211)

[1 Основні проблеми розробки сучасних баз даних. Аналіз предметної області та постановка задачі 8](#_Toc471844212)

[1.1 Основні проблеми розробки сучасних баз даних 9](#_Toc471844213)

[1.2 Аналіз предметної області 14](#_Toc471844214)

[1.2.1 Глосарій проекту 15](#_Toc471844215)

[1.2.2 Система бізнес-правил 15](#_Toc471844216)

[1.3 Постановка задачі 17](#_Toc471844217)

[2 Побудова концептуальної та логічної моделі даних предметної області 18](#_Toc471844218)

[2.1 Розробка концептуальної моделі даних 18](#_Toc471844219)

[2.2 Перетворення концептуальної моделі в логічну модель даних 20](#_Toc471844220)

[3 Проектування програмного забезпечення. Вибір системної архітектури. розробка необхідних UML діаграм. Вибір СКБД 22](#_Toc471844221)

[3.1 Розробка системної програмної архітектури 22](#_Toc471844222)

[3.2 Мотивований вибір СКБД та інструментальних програмних засобів для реалізації запропонованої системної архітектури 23](#_Toc471844223)

[3.2.1 Стислий огляд сучасних типів СКБД та критерії вибору СКБД для реалізації проекту 23](#_Toc471844224)

[3.2.2 Особливості інструментальних засобів програмної реалізації клієнтського додатку та бізнес-логіки системи 23](#_Toc471844225)

[3.3 Розробка прикладного програмного забезпечення 23](#_Toc471844226)

[3.4 Розробка візуального інтерфейсу користувачів системи 23](#_Toc471844227)

[4 Результати застосування розробленого програмного забезпечення 24](#_Toc471844228)

[4.1 Стислі відомості щодо розгортання системи 24](#_Toc471844229)

[4.2 Основні режими роботи із системою 24](#_Toc471844230)

[4.2.1 Режим покупця 24](#_Toc471844231)

[4.2.2 Режим адміністратора 24](#_Toc471844232)

[4.3 Результати тестування та рекомендації щодо удосконалення розробленої системи 24](#_Toc471844233)

[Висновки 25](#_Toc471844234)

[Список джерел інформації 26](#_Toc471844235)

# Перелік позначень та скорочень

БД - база даних.

ПС - програмна система.

MS - Microsoft.

ПрО - предметна область.

КМД – концептуальна модель даних.

CASE – Computer Aided Software Engineering(засоби автоматизацій розробки програм).

СА – системна архітектура.

СКБД – система керування базами даних.

ORM – Object-Relational Mapping(об’єктно-реляційне відображення).

CSS – Cascading Style Sheets(каскадні таблиці стилів).

# Вступ

Розвиток інформаційних та комунікаційних технологій призводить до збільшення масштабів діяльності підприємств, що плавно перетікає у глобалізацію бізнесу. Цей процес призводить до появи вимог зі збільшення оброблюваних об’ємів даних та виникнення нових, більш складних задач з їх обробки.

Для якісної реалізації цих вимог, дані треба організовувати та структурувати. Виконання цієї задачі забезпечує інформаційна (автоматизована) система. Інформаційна система (ІС) – це комунікаційна система, що забезпечує збирання, пошук, оброблення та пересилання інформації, метою якої являється забезпечення інформаційних потреб користувачів.

Для організації та зберігання даних в інформаційній системі використовуються бази даних. База даних (Database) – це сукупність даних, організованих відповідно до концепції, яка описує характеристику цих даних і взаємозв'язки між їх елементами. Використання баз даних є однією з характерних рис більшості сучасних інформаційних систем. По своїй суті бази даних є ядром інформаційних систем, що зумовлено первинністю їх проектування та розробки відносно інших компонентів ПС. Тому теорії створення та практиці використання баз даних приділяється велика увага.

Використання БД дозволяє усунути надмірність даних, що призводить до скорочення обсягу необхідної для зберігання цих даних пам’яті, та уникнення протиріч при зберіганні та обробки даних.

Майже кожна організація має свою інформаційну систему та базу даних, що домагає прискорити роботу та частково автоматизувати процеси у цій системі.

Предметною областю даної курсової роботи є «Облік даних онлайн аукціону». Дана тема є достатньо актуальною, тому що роздрібна онлайн-торгівля, зокрема шляхом аукціону, стає все більш популярною у світі навіть серед користувачів, що не є приватними підприємцями у сфері торгівлі. Великі масштаби подібної діяльності потребують належного підходу до проектування додатків та баз даних відповідних систем, основними вимогами до яких є можливість зберігання великих об’ємів даних та одночасні швидкість та надійність їх обробки. Останні дві вимоги диктуються тим положенням, що для успішної роботи аукціону необхідно забезпечувати його учасників актуальною та коректною інформацією.

Завданням курсової роботи є дослідження даної предметної області та створення додатку, що працює з БД даної предметної 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 проектування, програмування та роботи з базами даних.

# 1 Основні проблеми розробки сучасних баз даних. Аналіз предметної області та постановка задачі

## Основні проблеми розробки сучасних баз даних

На сьогоднішній день, дуже широкого розповсюдження здобула автоматизація роботи організацій майже в усіх сферах людської діяльності. Частіше за все, автоматизація досягається шляхом застосування інформаційних систем, які у корені змінили методи роботи багатьох організацій, у яких вони були впроваджені. ІС дозволяють значно підвищити ефективність, швидкодію, надійність та зручність роботи.

Актуальність створення зумовлена інформатизацією, зростанням ролі інформаційного забезпечення при прийнятті управлінських рішень, а також багатофакторністю і різноманітністю проблем господарства та бізнесу, що потребують вирішення.

Для організації роботи з даними в інформаційних системах використовуються бази даних, які представляють собою основу цих ІС. За останні роки, завдяки розвитку технології баз даних, було створено багато різноманітних інформаційних систем, як внутрішніх, так і доступних широкому колу користувачів. Майже усі сучасні програмні системи, які працюють з деякими даними використовують бази даних.

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.

Відомі два підходи до організації інформаційних масивів: файлова організація та організація у вигляді бази даних. Файлова організація передбачає спеціалізацію та збереження інформації, орієнтованої, як правило, на одну прикладну задачу, та забезпечується прикладним програмістом. Така організація дозволяє досягнути високої швидкості обробки інформації, але характеризується рядом недоліків. Характерна риса файлового підходу - вузька спеціалізація як обробних програм, так і файлів даних, що служить причиною великої надлишковості, тому що ті самі елементи даних зберігаються в різних системах. Оскільки керування здійснюється різними особами, відсутня можливість виявити порушення суперечливості збереженої інформації [1].

Розроблені файли для спеціалізованих прикладних програм не можна використовувати для задоволення запитів користувачів, які перекривають дві і більше області. Крім того, файлова організація даних внаслідок відмінностей структури записів і форматів передання даних не забезпечує виконання багатьох інформаційних запитів навіть у тих випадках, коли всі необхідні елементи даних містяться в наявних файлах. Тому виникає необхідність відокремити дані від їхнього опису, визначити таку організацію збереження даних з обліком існуючих зв’язків між ними, яка б дозволила використовувати ці дані одночасно для багатьох застосувань. Вказані причини обумовили появу баз даних. База даних може бути визначена як структурна сукупність даних, що підтримуються в активному стані та відображає властивості об'єктів реального світу. В базі даних містяться не тільки дані, але й описи даних, і тому інформація про форму зберігання вже не схована в сполученні "файл-програма", вона явним чином декларується в базі.

База даних орієнтована на інтегровані запити, а не на одну програму, як у випадку файлового підходу, і використовується для інформаційних потреб багатьох користувачів. В зв'язку з цим бази даних дозволяють в значній мірі скоротити надлишковість інформації. Перехід від структури БД до потрібної структури в програмі користувача відбувається автоматично за допомогою систем управління базами даних (СКБД).

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

База даних (БД) - це структурована сукупність даних, відносяться до деякої ПрО, яка:

1. Організована із застосуванням певної МД;
2. Створена програмними засобами деякої СКБД і постійно зберігається на носіях пам'яті;
3. Знаходиться під управлінням цієї СКБД.

Головним завданням БД є гарантоване збереження значних обсягів інформації та надання доступу до неї користувачеві або ж прикладній програмі. Таким чином БД складається з двох частин: збереженої інформації та системи управління нею. З метою забезпечення ефективності доступу записи даних організовують як множину елементів даних [2].

Предметна область - це певна, логічна виділена, частина оточуючого нас реального світу, яка містить в собі деякі дані, завдання по їх обробці, і в якій існують різні користувачі, зацікавлені у використанні результатів по обробці цих даних.

Дані - це будь-який набір символів, які стосуються певної формалізованої системі їх запису, які використовуються для опису фактів або гіпотез в деякій предметній області, і які допускають їх комп'ютерну обробку.

Знання - сукупність даних про деякі об'єкти, явища та процеси в предметній області, про їхні зв'язки і правилах інтерпретації, які можуть бути використані для отримання нових фактів у даній предметній області.

Інформація - це прирощення знань про деяку предметну область, отриманих на підставі обробки відповідних даних.

Модель даних - це формалізоване представлення даних про деяку предметну область, що задається трійкою множин:

MD = {S, О, C}, де:

S - непорожнє множина певних структур даних;

О - непорожнє множина операцій (функцій), дозволених для обробки цих

структур даних;

C - множина обмежень, які повинні бути дотримані при роботі з цією моделлю даних (МД).

СКБД - це сукупність мовних і програмних засобів, що дозволяють створювати і використовувати БД для вирішення завдань користувачів у певній предметній області [3].

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

У сучасних ІС реалізована трирівнева схема подання даних, яка передбачає:

1 Рівень концептуального представлення даних (conceptual data view) – на ньому інформація про деяку предметну область 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 тощо. Це – так звана 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 – так звана 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:

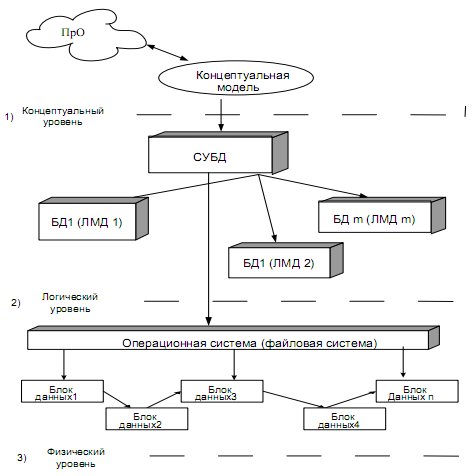


Рисунок 1.1 – Трирівнева схема представлення даних в ІС

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

* аналітики;
* замовники проекту;
* експерти з предметної області;
* кінцеві користувачі ІС.

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

* глосарій проекту (project glossary);
* система бізнес-правил (business rules system).

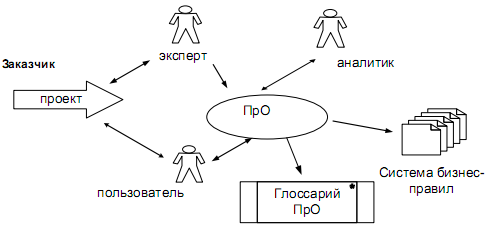


Рисунок 1.2 – Схема процесу аналізу предметної області

## 1.3 Аналіз предметної області

### 1.3.1 Вихідний опис предметної області

На сайті можуть реєстровані рядові користувачі та адміністратори, кожен з яких має власний аккаунт. Користувач може створювати власний набір лотів для майбутніх аукціонів. Кожен лот має власну обов’язкову категорію, що входить до ієрархічної системи категорій. Далі користувач може висунути певний товар з власного списку на аукціон, об’явивши початкову ціну, дату закінчення аукціону. На основі початкової ціни визначається крок аукціону. Будь-який користувач(окрім власника лоту) має право на участь в аукціоні, анонімно встановлюючи власну ставку(багаторазово за необхідності). Користувачі можуть залишати відгуки один про одного та відсилати скарги на поведінку інших користувачів адміністрації. Адміністратор розглядає скарги та пропозиції користувачів, фіксуючи статус заявки. Адміністратор має право накладати перманентний бан на користувача. Також він може редагувати набір категорій лотів та керує повноваженнями інших адміністраторів, тобто має право призначати нових та усувати вже існуючих. Член адміністрації має доступ до функціоналу рядового користувача.

### 1.3.2 Глосарій проекту

Глосарій проекту – перелік основних термінів предметної області з їх стислим смисловим описом (розширений словник), перекладанням іншою мовою та прикладами коректних значень.

Аналізуючи ПрО, можна привести наступний глосарій:

1. Звичайний користувач – зареєстрована особа, що має право на участь у торговому процесі онлайн-аукціону та має базовий функціонал.
2. Адміністратор – користувач, що має доступ до розширенного функціоналу, необхідного для адміністрування та модерації.
3. Аукціон – публічний продаж товару, що відбувається за раніше встановленими правилами(сутність);
4. Ставка – грошова сума, що заявлена учасником аукціону як ціна, яку він готовий заплатити за лот у даний момент часу(сутність);
5. Лот – об’єкт торгу на аукціоні(сутність);
6. Категорія лоту – вид предмету, до якого належить лот(сутність);
7. Відгук – думка одного користувача про іншого, заснована на враженні першого від співробітництва з другим(сутність);
8. Скарга – висловлене невдоволення поведінкою іншого користувача(сутність);
9. Роль – право на користування певним функціоналом системи, що йменується назвою тих користувачів, що мають це право. Приклад: «звичайний користувач», «адміністратор»(сутність);
10. Повноваження – відповідність користувача його ролі. Передбачена наявність у одного користувача цілого списку повноважень(сутність);
11. Крок ставки – сума, на яку учасник аукціону може підвищити поточну ціну лоту. Обчислюється в залежності від початкової ціни лоту(атрибут сутності).

### 1.3.3 Система бізнес-правил

Бізнес-правило – логічно коректний, однозначний, ненадлишковий вислів про деякі інформаційні об’єкти та їх зв’язки в даній предметній області.

Аналізуючи ПрО, можна привести наступну систему бізнес-правил:

1. кожний користувач може мати безліч ролей і кожна роль може бути присвоєна багатьом користувачам;
2. кожний користувач може створити безліч лотів, в той час як кожний лот може мати лише одного власника, причому існують такі користувачі, які не створили жодного лоту;
3. користувачі залишають відгуки один про одного, причому:

* кожний користувач може написати безліч відгуків, але кожний відгук може мати лише одного автора, причому існують такі користувачі, що не написали жодного відгуку;
* кожний користувач може отримати безліч відгуків, але кожний відгук може бути отриманий лише одним користувачем;

1. кожний користувач може написати безліч скарг, але кожна скарга може бути написана лише одним користувачем, причому існують такі користувачі, що не написали жодної скарги;
2. кожний лот має одну категорію, але до кожної категорії може бути віднесено безліч лотів, причому існують такі категорії, до яких не віднесено жодного лоту;
3. кожна категорія може підпорядковуватись в ієрархії лише одній категорії, і кожна категорія може підпорядковувати безліч категорій, причому існують категорії, що не підпорядковуються жодній з існуючих категорій, та категорії, що не підпорядковують жодної категорії;
4. лот може бути виставлений на безліч аукціонів, і на кожний аукціон може бути виставлений лише один лот, причому існують такі лоти, що не виставлялись на жодному з аукціонів;
5. кожен користувач може висувати ставки на багатьох аукціонах багаторазово, і на кожному аукціоні можуть виставляти ставки безліч користувачів, причому існують такі користувачі, що не робили ставки у жодному аукціоні, та аукціони, на яких ніколи не виставляли ставки.

## 1.4 Постановка задачі

Під час виконання даної курсової роботи потрібно створити програмну систему для предметної області “Облік даних онлайн-аукціону”, 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:

1. підключення до БД та використання її даних;
2. надання зручного інтерфейсу користувача для редагування та пошуку необхідних даних БД;
3. відображення всіх таблиць ІС та можливість їх редагування;
4. інтерфейс користувача передбачає створення вікна з необхідними кнопками, таблицями для ведення та редагування даних, рядками введення даних для пошуку.

До основних задач курсової роботи відносяться:

1. дослідження предметної області;
2. отримання практичних навичок у об’єктно-орієнтованому аналізі та застосуванні CASE засобів;
3. вивчення технології програмування застосувань з доступом до БД, програмних засобів СКБД;
4. створення БД та застосування клієнтської прикладної програми для роботи з ним.

# 2 Побудова концептуальної та логічної моделі даних предметної області

## 2.1 Розробка концептуальної моделі даних

Концептуальна модель даних – це модель, представлена множиною понять і зв’язків між ними, що визначають змістовну структури розглянутої предметної галузі. Ця модель не повинна відображати конкретні потреби окремих користувачів або застосувань. Вона має фіксувати, чим є предметна область в цілому, а не з точки зору інтересів або потреб користувачів. Її розробляють перед створенням БД. Зазвичай при цьому використовують спеціальні нотації. Найбільш поширеними є канонічна ER нотація Пітера Чена та UML-діаграма класів[4].

На рисунку 2.1 зображена КМД у нотації UML.

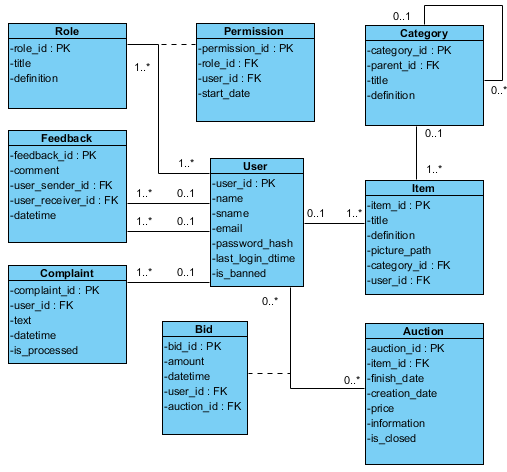


Рисунок 2.1 - КМД у нотації UML

Наведемо перелік сутностей концептуальної моделі:

1. User – відображає користувача системи;
2. Role – відображає можливі види повноважень у системі;
3. Permission – є асоціативною сутністю та відображає наявність у користувача певного виду повноважень;
4. Feedback – відображає відгук користувача про співробітництво з іншим користувачем;
5. Complaint – відображає скаргу на поведінку користувача до адміністрації
6. Item – відображає лот, що створює користувач;
7. Category – відображає видову приналежність предмету до конкретної групи речей;
8. Auction – відображає аукціон;
9. Bid – відображає ставку, що ставиться користувачем під час аукціону;

AllFusion ERwin Data Modeler (ERwin) дозволяє наочно відображати складні структури даних. Зручна у використанні графічна середовище ERwin спрощує розробку баз даних і автоматизує багато трудомістких завдань, зменшуючи терміни створення високоякісних та високопродуктивних трансакційних 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[5].

На рисунку 2.2 зображена КМД у нотації IDEF1X.

## 

Рисунок 2.2 - КМД у нотації IDEF1X

## 2.2 Перетворення концептуальної моделі в логічну модель даних

На основі раніше розробленої КМД засобами CASE-системи ERwin було перетворено КМД в логічну модель даних.

Логічна модель даних залежить від конкретної СКБД. У логічній моделі втримується інформація про всі об'єкти. В логічній моделі важливо описати всю інформацію про конкретні об'єкти в таблицях, колонках, індексах, процедурах тощо. Компонентами логічної моделі даних є таблиці, стовпці і відносини. Сутності концептуальної моделі, ймовірно, стануть таблицями у логічній моделі. Атрибути стануть стовпцями. 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.

Логічна модель даних у нотації IDEF1X зображена на рисунку 2.3.

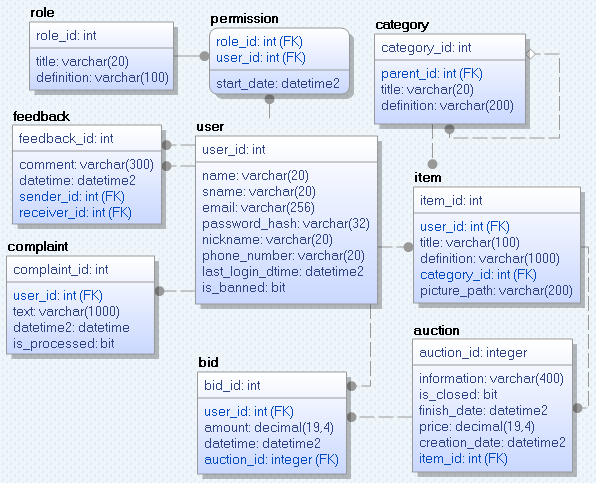


Рисунок 2.3 - ЛМД у нотації IDEF1X

# 3 Проектування програмного забезпечення. Вибір системної архітектури. розробка необхідних UML діаграм. Вибір СКБД

## 3.1 Розробка системної програмної архітектури

Для того, щоб визначити системну програмну архітектуру розроблено UML-діаграму основних варіантів використання, яка зображена на рисунку 3.1.

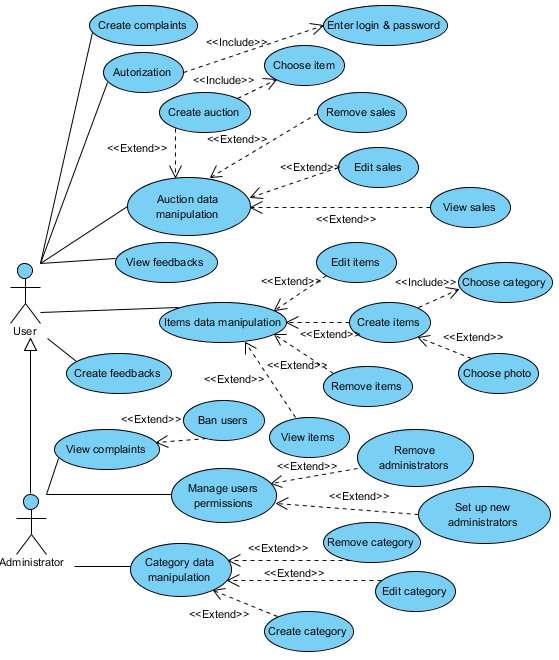


Рисунок 3.1 - UML-діаграма основних варіантів використання

Основними задачами системи є робота з таблицями БД: внесення, вибірка, даних є самими масовими видами запитів.

Система архітектура (СА) – абстракція моделювання в програмній інженерії, яка задається кортежом з трьох множин:

* кінцева непуста множина програмних компонентів / модулів, які реалізують необхідний функціонал;
* кінцева непуста множина структурних форм, в які можуть бути поєднані програмні компоненти;
* кінцева непуста множина інтерфейсів, які система надає зовнішнім модулям, підсистемам тощо.

Майже кожна програмна система, що проектується має володіти системною архітектурою. В результаті аналізу багатьох реальних проектів, з’явилось поняття про еталонну системну архітектуру, яка представляє собою певну множину типових, стандартних архітектурних рішень.

Еталонна системна архітектура – це набір певних правил побудови та засобів використання деякої множини програмних компонентів, виконання яких забезпечує наявність в цільовій програмній системі певних значень показників якості.

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- додатком (для бізнес-логіки виділено окремий сервер).

Оскільки предметна область передбачає велику кількість даних, складних запитів, багатьох підключень до БД - система потребує потужного апаратного забезпечення серверу БД и додатку, тому необхідним варіантом серед усіх СА є 3-х рівнева клієнт-серверна СА з “тонким” клієнтом і сервером - додатком.

Даний вибір передбачає, що для бази даних виділяється окремий сервер, а вся робота з обробки та подання цих даних переноситься на сервер додатку. Подібна архітектура володіє такими перевагами СА с тонким клієнтом, як низька вартість модернізації та більш висока масштабованість у порівнянні з СА з товстим клієнтом, та сама по собі є більш продуктивною у роботі, ніж СА з тонким клієнтом. Висока продуктивність зумовлена розвантаженням серверу за стосунку та переносом частини обчислювальної роботи на окремий сервер БД. Недоліком 3-х рівневої архітектури є більша вартість проектування та розробки у порівнянні з іншими СА.

Діаграма розміщення, що відображає архітектуру спроектованої ПС, зображена на рисунку 3.2.

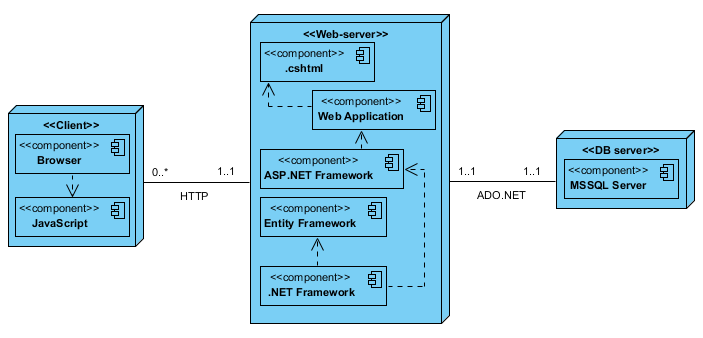


Рисунок 3.2 – Діаграма розміщення

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

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

Системи керування базами даних (СКБД) - це програмні засоби, за допомогою яких можна створювати бази даних, заповнювати їх та працювати з ними [6]. У світі існує багато різноманітних систем управління базами даних. Багато з них насправді є не закінченими продуктами, а спеціалізованими мовами програмування, за допомогою яких кожний, хто вивчить дану мову, може сам створювати такі структури, які йому потрібні, і вводити в них необхідні елементи управління.

Класифікація СКБД за моделлю даних:

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

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

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 відносяться: Oracle, Microsoft SQL Server, MySQL, PostgreSQL, Sybase Adaptive Server Anywhere і т.п.

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:

* вид програмного продукту;
* категорії користувачів;
* зручність і простота використання;
* модель представлення даних;
* якість засобів розробки;
* якість засобів захисту та контролю коректності бази даних;
* якість комунікаційних засобів;
* фірма-розробник.

Для реалізації онлайн-аукціону була обрана СКБД MS SQL Server 2012 Express, бо вона найбільш інтегрована з системою Windows і продуктами MS Visual Studio та фрейморкамі на базі .NET.

MS 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 та інші (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 для 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.

У SQL Server поліпшені можливості віддзеркалювання 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 дзеркалювання БД потрібно обмін даними між усіма учасниками. SQL Server забезпечує стискування 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 збільшується[7].

### 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[8].

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).

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 забезпечення доступу до БД було обрано технологію ADO.NET Entity Framework. Entity Framework - об'єктно-орієнтована технологія доступу до даних, є object-relational mapping (ORM) рішенням для .NET Framework від Microsoft [9].

ADO.NET надає можливість взаємодії з об'єктами як за допомогою LINQ, так і з використанням Entity SQL. Для полегшення побудови рішень використовується зв'язка з ASP.NET MVC Framework.

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.

## 3.4 Розробка візуального інтерфейсу користувачів системи

В ході виконання курсової роботи був розроблений візуальний інтерфейс користувача на мові C# з використанням ASP.NET MVC 5 Framework.

Сам інтерфейс користувача був зроблений на HTML з використанням Razor[10], що представляє собою рушій відображень(view engine) з більш лаконічним синтаксисом, ніж попередній рушій Web Forms. Razor не є новою мовою програмування, бо є лише способом формування відображень, що має синтаксис для переходу від розмітки HTML до коду C#.

Для стилізації структури відображень та окремих її елементів був використаний CSS фреймворк Bootstrap, що дозволяє створювати веб-сторінки з адаптивним дизайном[11].

# 4 Результати застосування розробленого програмного забезпечення

## 4.1 Стислі відомості щодо розгортання системи

Щоб використовувати розроблену систему необхідно встановити MS SQL Server 2012, IIS Server 10 та .NET Framework 4.5.2.

Нижче приведені мінімальні характеристики, які необхідно мати комп’ютеру.

Пам'ять - 2 ГБ ОЗУ.

Швидкодія процесора - 1,5 ГГц;

Жорсткий диск: 4 ГБ вільного місця.

Операційна система: Windows 7 SP1 і вище.

Після встановлення IIS Server 10 треба розгорнути проект та налаштувати підключення до MS SQL Server 2012.

## 4.2 Основні режими роботи із системою

Для роботи з системою потрібно реєструватися. Для цього необхідно ввести особисті дані для взаємодії з іншими користувачами, логін та пароль, як зображено на рисунку 4.1.

Розроблена система має два режими: непривілейований користувач, адміністратор.

### 4.2.1 Режим непривілейованого користувача

### 4.2.2 Режим адміністратора

## 4.3 Результати тестування та рекомендації щодо удосконалення розробленої системи

Після створення бази даних, необхідно провести ряд тестувань. Від результатів тестування буде дуже багато залежати експлуатація бази даних, оскільки, якщо запити будуть працювати не правильно, то користувач бази даних не зможе дізнатися правильну інформацію про належні об’єкти.

Тестуванню підлягали майже всі створені таблиці, вони перевірялись на правильність та відповідність атрибутів до інформації яку вони зберігають.

Під час розробки виникли наступні проблеми:

1. Проблеми з налаштування з’єднання з базою даних;
2. Проблеми з відображенням складних даних;

На основі тестування були виправлені всі проблеми і зараз система працює коректно.

# Висновки

В ході написання курсової роботи були закріплені практичні навички аналізу предметної області, складання бізнес правил та глосарію проекту, побудування концептуальних схем даних та на основі них створення БД.

Був проведений аналіз нині існуючих еталонних системних архітектур, їх переваг та недоліків, на основі чого була обрана цільова архітектура для її подальшої реалізації з оглядом на обрані СКБД та інструментальні засоби програмної реалізації клієнтського додатку та бізнес-логіки системи.

Було ознайомлено з класифікацією та найбільш поширеними СКБД та технологіями зв’язку з ними клієнтського програмного забезпечення. Покращені навички ведення проектної документації за допомогою CASE-засобів.

Була розроблена та задокументована програмна система онлайн-аукціону, що відповідає вимогам до неї.

На основі обраної СКБД MSSQL Server 2012 та ORM-технології з’єднання з БД ADO.NET Entity Framework, створених раніше глосарію, системи бізнес-правил та результатів аналізу предметної області була створена база даних , та шар бізнес-логіки, що були протестовані та розміщені згідно з обраною 3-х рівневою системною архітектурою з виділеним сервером додатку та тонким клієнтом.

Візуальний інтерфейс користувачів системи, що реалізований за допомогою технології ASP.NET MVC 5, створювався з оглядом на результати аналізу предметної області та дозволяє проводити користувачам маніпуляції з базою даних у тих межах, що були передбачені реалізованою системою повноважень користувачів системи.

# Список джерел інформації

1. Хомоменко А.А., Цыганков В.М., Мальцев М.Т. Базы данных. - СПб.: Корона, 2002 - 672 с.
2. Пасічник В.В., Резниченко В.А. Організація баз даних та знань (підручник МОНУ). - К.: BHV, 2006. - 384 с.
3. Конолли Т. и др. Базы данных: проектирование, реализация и сопровождение. - М.: Вильямс, 2003. - 1120 с.
4. Ларман К. Применение UML 2 и шаблонов проектирования. Третье издание. - М.: Вильямс, 2013. – 736 с.
5. Назарова О.Б., Масленникова О. Е. Разработка реляционных баз данных с использованием CASE-средства ALL FUSION ErWin Data Modeler - М.: 2013 - 74с.
6. Методические указания к проведению лабораторных занятий по курсу “Организация баз данных и знаний”, часть 2: “Технология работы с СУБД MySQL” для студентов направления подготовки специалистов 0804 - "Компьютерные науки" / Сост. Ткачук Н.В., Борщ Е.В., Климова Е.Ю., Костанян А.В., Матат Е.А. - Харьков: НТУ "ХПИ", 2007. - 60 c.
7. Опис функціональності, можливостей та переваг MS SQL Server. // https://www.microsoft.com/ru-ru/server-cloud/products/sql-server/
8. Опис середовища розробки MS Visual Studio 2015. // https://www.visualstudio.com/
9. Стиллмен Э., Грин Дж. Изучаем C#. 3-е изд. - СПб.: Питер, 2014. - 816 с.
10. Razor. Призначення та синтаксис. // https://docs.microsoft.com/en-us/aspnet/core/mvc/views/razor
11. Офіційний сайт проекту Bootstrap. // getbootstrap.com