

НАЦІОНАЛЬНИЙ ТЕХНІЧНИЙ УНІВЕРСИТЕТ УКРАЇНИ
«КИЇВСЬКИЙ ПОЛІТЕХНІЧНИЙ ІНСТИТУТ
імені ІГОРЯ СІКОРСЬКОГО»

факультет інформатики та обчислювальної техніки
(повна назва інституту/факультету)

кафедра інформаційних систем та технологій
(повна назва кафедри)

Курсова робота

з дисципліни «Програмування 3»

на тему: Система проведення аукціонів

Виконав : студент 1 курсу, групи ІА-23
(шифр групи)

Лядський Дмитро Сергійович
(прізвище, ім'я, по батькові) _____ (підпис)

Науковий керівник Колеснік В. М.
(посада, науковий ступінь, вчене звання, прізвище та ініціали) _____ (підпис)

Члени комісії _____
(посада, науковий ступінь, вчене звання, прізвище та ініціали) _____ (підпис)

_____ (посада, науковий ступінь, вчене звання, прізвище та ініціали) _____ (підпис)

Засвідчую, що у цій курсовій роботі немає запозичень з праць інших авторів без
відповідних посилань.

Студент
Лядський Д. С. _____
(підпис)

Київ – 2023

ЗМІСТ

| | |
|---|----|
| ВСТУП | 3 |
| 1 ВИМОГИ ДО СИСТЕМИ | 5 |
| 1.1 Функціональні вимоги до системи..... | 5 |
| 1.2 Нефункціональні вимоги до системи..... | 5 |
| 2 СЦЕНАРІЇ ВИКОРИСТАННЯ СИСТЕМИ..... | 6 |
| 2.1 Діаграма прецедентів..... | 6 |
| 2.2 Опис сценаріїв використання системи..... | 7 |
| 3 АРХІТЕКТУРА СИСТЕМИ..... | 15 |
| 4 РЕАЛІЗАЦІЯ КОМПОНЕНТІВ СИСТЕМИ | 17 |
| 4.1 Загальна структура проекту | 17 |
| 4.2 Компоненти рівня доступу до даних..... | 18 |
| 4.3 Компоненти рівня бізнес-логіки..... | 20 |
| 4.4 Компоненти рівня інтерфейсу користувача | 22 |
| ВИСНОВКИ..... | 23 |
| ПЕРЕЛІК ВИКОРИСТАНИХ ДЖЕРЕЛ..... | 24 |

ВСТУП

Аукціони є важливим інструментом в економіці, що дозволяють ефективно визначати ціни на товари і послуги, розподіляти ресурси та сприяти конкуренції. Система проведення аукціонів має актуальність як для державних установ, комерційних організацій, так і для приватних осіб. Нижче наведено аргументи, що підкреслюють актуальність такої системи:

1. Ефективне визначення цін: Аукціони дозволяють визначити справедливую ціну на товар або послугу шляхом конкуренції між учасниками. Це сприяє забезпеченню оптимальних цін на ринку.

2. Прозорість: Система проведення аукціонів надає можливість учасникам отримувати повну інформацію про умови та параметри аукціону. Це забезпечує прозорість процесу та довіру до системи.

3. Конкуренція: Аукціони стимулюють конкуренцію між учасниками, що дозволяє отримувати оптимальні результати. Це сприяє розвитку ефективного ринкового середовища та зниженню витрат.

4. Широкий спектр застосування: Система проведення аукціонів може бути застосована в різних галузях, таких як торгівля, закупівлі державних установ, нерухомість, інвестиції та інше. Це робить її універсальним інструментом для багатьох сфер діяльності.

5. Економія часу і ресурсів: Аукціони дозволяють ефективно розподілити товари та послуги, зменшуючи час і зусилля, необхідні для укладання угод. Вони дозволяють учасникам швидко зробити ставку або придбати товар, в результаті чого процес торгівлі стає більш ефективним.

6. Розширений ринок: Онлайн-системи проведення аукціонів відкривають доступ до більш широкого кола учасників, незалежно від їх географічного розташування. Це сприяє збільшенню конкуренції та можливостям отримання кращих пропозицій.

Метою створення системи проведення аукціонів є забезпечення ефективного та прозорого механізму для визначення цін на товари і послуги, розподілу ресурсів і

сприяння конкуренції. Головні задачі, які будуть вирішуватись такою системою, включають:

- Організація аукціонів: Система повинна надати можливість створювати та організовувати аукціони для різних типів товарів і послуг.

- Можливість робити пропозиції учасниками: Система повинна дозволяти учасникам робити пропозиції до лота.

- Публікація інформації: Система має надавати можливість публікації повної інформації про аукціони. Це допоможе залучити більше учасників і забезпечити прозорість процесу.

- Обробка ставок та визначення переможців: Система повинна автоматично обробляти ставки учасників та визначати переможців аукціонів відповідно до встановлених правил. Це забезпечить швидке та точне визначення результатів. Створення системи проведення аукціонів має на меті поліпшення ефективності, прозорості та конкурентоспроможності процесу визначення цін та розподілу ресурсів.

1 ВИМОГИ ДО СИСТЕМИ

1.1 Функціональні вимоги до системи

Система має відповідати наступним функціональним вимогам:

- незареєстрований користувач повинен мати можливість переглядати активні лоти;
- незареєстрований користувач повинен мати можливість переглядати інформацію про лот;
- незареєстрований користувач повинен мати можливість здійснювати пошук лотів по словам у назві та описі лота;
- незареєстрований користувач повинен мати можливість переглядати профіль власника лота;
- зареєстрований користувач повинен мати усі можливості, що є у незареєстрованого користувача, а також він повинен мати можливість робити пропозиції до лотів і створювати свої власні лоти;
- власник лота повинен мати усі можливості, що є у незареєстрованого і зареєстрованого користувача, а також робити певні маніпуляції з лотом, власником якого він є (змінювати назву, опис, початкову ставку лота, видаляти лот, запускати або зупиняти торги)

1.2 Нефункціональні вимоги до системи

Система має відповідати наступним нефункціональним вимогам:

- система повинна мати механізми захисту конфіденційності, цілісності та доступності даних;
- система повинна забезпечувати швидку відповідь на дії користувачів.
- система повинна мати відкриту архітектуру;
- система повинна мати веб-інтерфейс;
- інтерфейс користувача має бути зручним та інтуїтивно-зрозумілим;
- система повинна бути крос-платформною.

2 СЦЕНАРІЇ ВИКОРИСТАННЯ СИСТЕМИ

2.1 Діаграма прецедентів

Діаграма прецедентів системи представлена на рис. 2.1.

Акторами є користувачі системи: незареєстрований (гість) та зареєстровані (користувач та власник лота).

Зареєстрованому користувачу, який не є власником лота, доступна уся функціональність, що і незареєстрованому, а також можливість створювати лоти і робити ставки, зареєстрованому користувачу, який є власником лота, доступна уся функціональність, що і іншим користувачам, крім можливості робити ставку на свій лот, а також можливість змінювати назву, опис, початкову ставку лота, видаляти лот, запускати або зупиняти торги. Детально усі сценарії використання описані у наступному підрозділі.

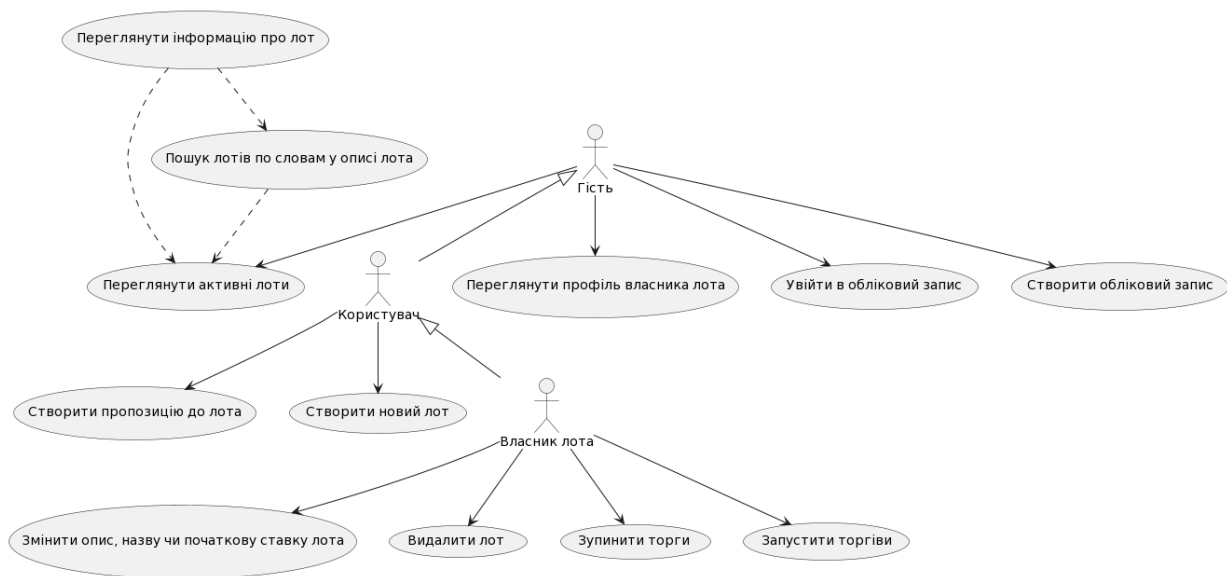


Рисунок 2.1 – Діаграма прецедентів

2.2 Опис сценаріїв використання системи

Детальні описи сценаріїв використання наведено у таблицях 2.1 – 2.8.

Таблиця 2.1 – Сценарій використання «Пошук лота за ключовими словами в описі»

| | |
|------------------------|--|
| Назва | Пошук лота за ключовими словами в описі |
| ID | 1 |
| Опис | Користувач, використовуючи поле для пошуку, шукає лот, що містить в назві або в описі ключове слово |
| Актори | Гість, Користувач |
| Вигоди компанії | Користувачі зможуть швидко знаходити потрібні їм лоти, тому не припинять користуватися сервісом через нудний та довгий пошук |
| Частота користування | Постійно |
| Тригери | Користувач вводить пошуковий запит у полі для пошуку |
| Передумови | Пошукове поле доступне у будь-якому вікні |
| Постумови | Користувач потрапляє на вікно з результатами пошуку |
| Основний розвиток | Користувач вводить запит у пошукову строку, натискає на кнопку Enter |
| Альтернативні розвитку | — |
| Виняткові ситуації | — |

В таблиці 2.2 представлений сценарій використання «Перегляд лоту»

Таблиця 2.2 – Сценарій використання «Перегляд лоту»

| | |
|------------------------|---|
| Назва | Перегляд лоту |
| ID | 2 |
| Опис | Користувач обирає лот, про який бажає дізнатися більш детальну інформацію |
| Актори | Гість, Користувач |
| Вигоди компанії | Користувач зможе зручно отримати інформацію про лот і, за бажанням, зробити ставку |
| Частота користування | Постійно |
| Тригери | Користувач натискає на кнопку “Перейти до лота” біля лота |
| Передумови | У вікні є список лотів, які можна переглянути |
| Постумови | Відкривається сторінка з лотом |
| Основний розвиток | Користувач натискає на кнопку “ Перейти до лота ” Лот завантажується з серверу Користувач дізнається інформацію про лот |
| Альтернативні розвитку | Користувач натискає на кнопку “ Перейти до лота ” Лот завантажується з кешу пристрою Користувач дізнається інформацію про лот |
| Виняткові ситуації | Якщо користувач намагається отримати за допомогою адресного рядка лот якого не існує, то його перенаправляє на головну сторінку з повідомленням про помилку |

В таблиці 2.3 представлений сценарій використання «Перегляд головної сторінки із лотами»

Таблиця 2.3 – Сценарій використання «Перегляд головної сторінки із лотами»

| | |
|------------------------|--|
| Назва | Перегляд головної сторінки із лотами |
| ID | 3 |
| Опис | Користувач переглядає головну сторінку із лотами |
| Актори | Гість, Користувач |
| Вигоди компанії | Головна сторінка важлива для зацікавлення нових користувачів, та актуальна інформація буде утримувати старих |
| Частота користування | Постійно |
| Тригери | Користувач переходить на сайт/запускає додаток |
| Передумови | Немає |
| Постумови | Користувач потрапляє на головну сторінку |
| Основний розвиток | Користувач запускає додаток/переходить на сайт |
| Альтернативні розвитку | — |
| Виняткові ситуації | — |

В таблиці 2.4 представлений сценарій використання «Створення лота»

Таблиця 2.4 – Сценарій використання «Створення лота»

| | |
|------------------------|--|
| Назва | Створення лота |
| ID | 4 |
| Опис | Користувач створює лот і стає його власником |
| Актори | Користувач |
| Вигоди компанії | Можливість створювати лоти є однією з основних можливостей системи проведення аукціонів |
| Частота користування | Постійно |
| Тригери | Користувач заповнює відповідну форму на сторінці створення лота |
| Передумови | Користувач натискає на кнопку «Створити лот» в шапці сайту |
| Постумови | Створюється лот і користувач опиняється на головній сторінці |
| Основний розвиток | Користувач натискає на відповідну кнопку у шапці сторінки. Потрапляє на сторінку створення лота. Створює лот |
| Альтернативні розвитку | — |
| Виняткові ситуації | Якщо користувач не заповнює поле з назвою чи описом лота, то він отримує повідомлення про помилку і лот не створюється |

В таблиці 2.5 представлений сценарій використання «Створення ставки»

Таблиця 2.5 – Сценарій використання «Створення ставки»

| | |
|------------------------|--|
| Назва | Створення ставки |
| ID | 5 |
| Опис | Користувач робить ставку |
| Актори | Користувач |
| Вигоди компанії | Можливість робити ставку є однією з основних можливостей системи проведення аукціонів |
| Частота користування | Постійно |
| Тригери | Користувач ввів у відповідне поле розмір ставки і натиснув кнопку «Зробити ставку» |
| Передумови | Користувач знаходиться на сторінці потрібного лота |
| Постумови | Поточна ставка змінюється на нову |
| Основний розвиток | Користувач змінює розмір ставки, на більший за поточну ставку, і натискає «Зробити ставку» |
| Альтернативні розвитку | — |
| Виняткові ситуації | Користувач змінює розмір ставки, на менший за поточну ставку, і натискає «Зробити ставку». При цьому поточна ставка не змінюється, а користувач повертається на сторінку лота з повідомленням про помилку. |

В таблиці 2.6 представлений сценарій використання «Редагування лота»

Таблиця 2.6 – Сценарій використання «Редагування лота»

| | |
|------------------------|---|
| Назва | Редагування лота |
| ID | 6 |
| Опис | Користувач змінює назву, опис або початкову ставку |
| Актори | Користувач, який є власником лота |
| Вигоди компанії | Якщо користувач зробив помилку при створенні лота, то може це легко виправити, тобто це є зручна функція для користувача |
| Частота користування | Інколи |
| Тригери | Користувач натискає на відповідну кнопку |
| Передумови | Користувач знаходиться на сторінці лота і натискає кнопку «Редагувати» |
| Постумови | Змінюються відповідні значення лота |
| Основний розвиток | Користувач натискає на відповідну кнопку на сторінці лота і редагує його |
| Альтернативні розвитку | - |
| Виняткові ситуації | Якщо власник лота намагається видалити назву лота чи опис, то він опиниться на сторінці лота і отримає повідомлення про помилку |

В таблиці 2.7 представлений сценарій використання «Зупинка/початок торгів»

Таблиця 2.7 – Сценарій використання «Зупинка/початок торгів»

| | |
|------------------------|--|
| Назва | Зупинка/початок торгів |
| ID | 7 |
| Опис | Користувач може зупинити або розпочати торги |
| Актори | Користувач, який є власником лота |
| Вигоди компанії | Зручна функція для користувача |
| Частота користування | Часто |
| Тригери | Користувач натискає відповідну кнопку на сторінці лота |
| Передумови | Користувач знаходиться на сторінці лота |
| Постумови | Торги зупиняються і лот стає неактивним або навпаки розпочинаються і лот стає активним |
| Основний розвиток | Користувач натискає відповідну кнопку і торги зупиняються/починаються |
| Альтернативні розвитку | — |
| Виняткові ситуації | - |

В таблиці 2.8 представлений сценарій використання «Видалення лота»

Таблиця 2.8 – Сценарій використання «Видалення лота»

| | |
|------------------------|---|
| Назва | Видалення лота |
| ID | 8 |
| Опис | Користувач може видалити лот видаляє лот |
| Актори | Користувач, який є власником лота |
| Вигоди компанії | Користувач зможе видаляти лоти, які йому непотрібні |
| Частота користування | Часто |
| Тригери | Користувач натискає на відповідну поряд із на сторінці лота |
| Передумови | Користувач заходить на сторінку лота |
| Постумови | Лот видаляється |
| Основний розвиток | Користувач натискає на кнопку «Видалити лот», лот видаляється, а користувач опиняється на головній сторінці |
| Альтернативні розвитку | — |
| Виняткові ситуації | — |

3 АРХІТЕКТУРА СИСТЕМИ

Загальна архітектура системи наведена на рис. 3.1

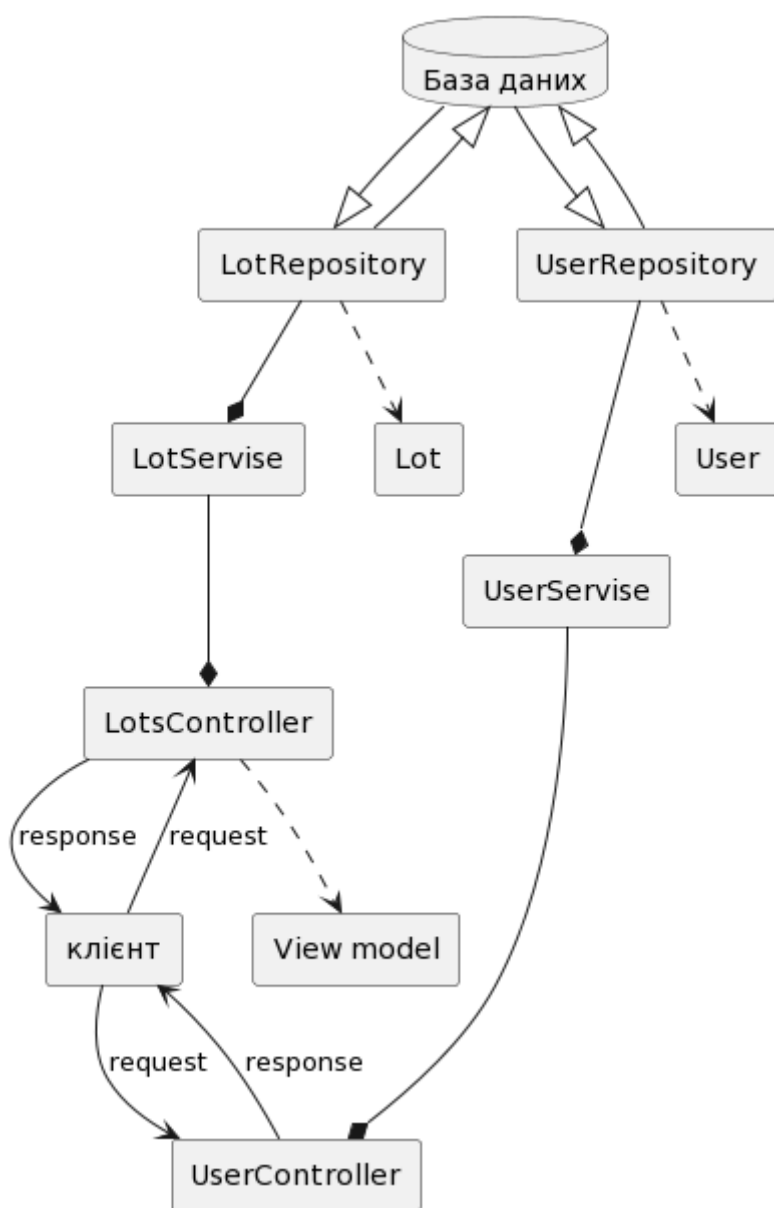


Рисунок 3.1 – Загальна архітектура системи

Система складається з наступних елементів:

- графічний інтерфейс;
- серверна частина;
- база даних.

Графічний інтерфейс складається з HTML-сторінок, CSS-стилів та JavaScript-скриптів, які забезпечують візуальне представлення та взаємодію з користувачем. Використовуючи певні елементи сторінки або адресного рядка, користувач надсилає запит за допомогою HTTP протокола до серверної частини, там запит оброблюється, виконуються певні перевірки, дістаються або зберігаються певні дані з бази даних. Далі серверна частина формує та відправляє відповідь користувачу, яку він отримує на HTML-сторінках.

Серверна частина складається з контролерів, сервісів, репозиторіїв і моделей. Контролер отримує дані, які користувач відправляє через графічний інтерфейс. Далі з контролера дані відправляються в сервіс, де виконуються основні операції над даними. За потреби сервіс звертається до репозиторію, який має можливість звертатися до бази даних, щоб отримати якусь інформацію або додати чи видалити щось. Якщо користувач відправив всі дані правильно, то після виконання операцій над даними він отримає необхідний йому результат.

4 РЕАЛІЗАЦІЯ КОМПОНЕНТІВ СИСТЕМИ

4.1 Загальна структура проекту

Загальна структура проекту представлена на рис.4.1

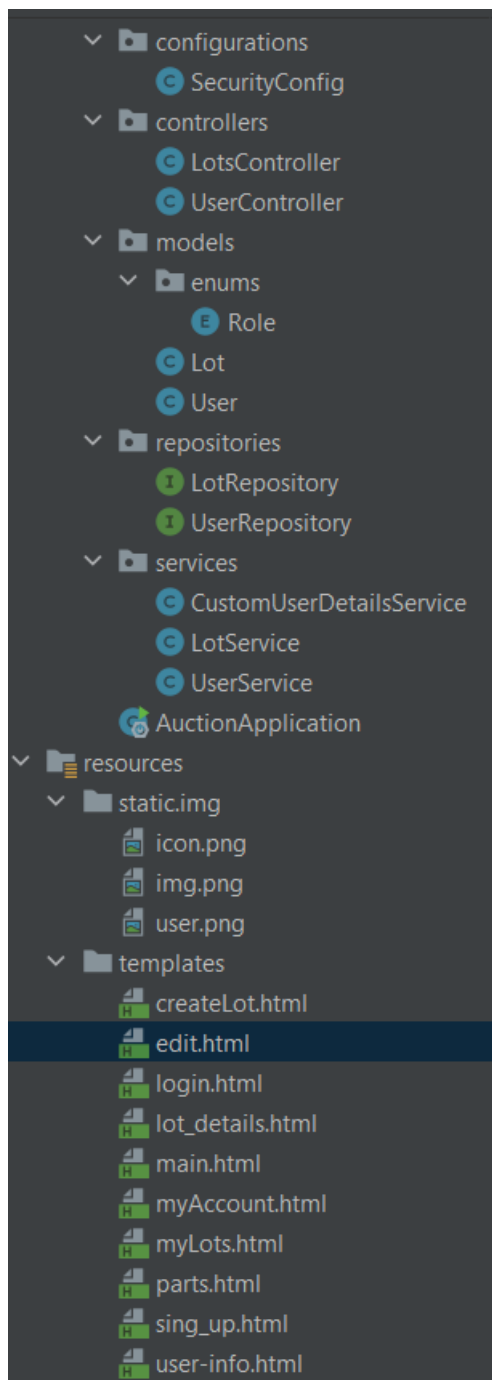


Рисунок 4.1 – Загальна структура проекту

Проект складається з веб-ресурсів, бібліотек, та вихідного коду, який в свою чергу можна поділити на компоненти рівня доступу до даних, компоненти бізнес-логіки та веб-компоненти.

4.2 Компоненти рівня доступу до даних

Основні сутності та інтерфейси рівня доступу до даних наведені на рис. 4.2

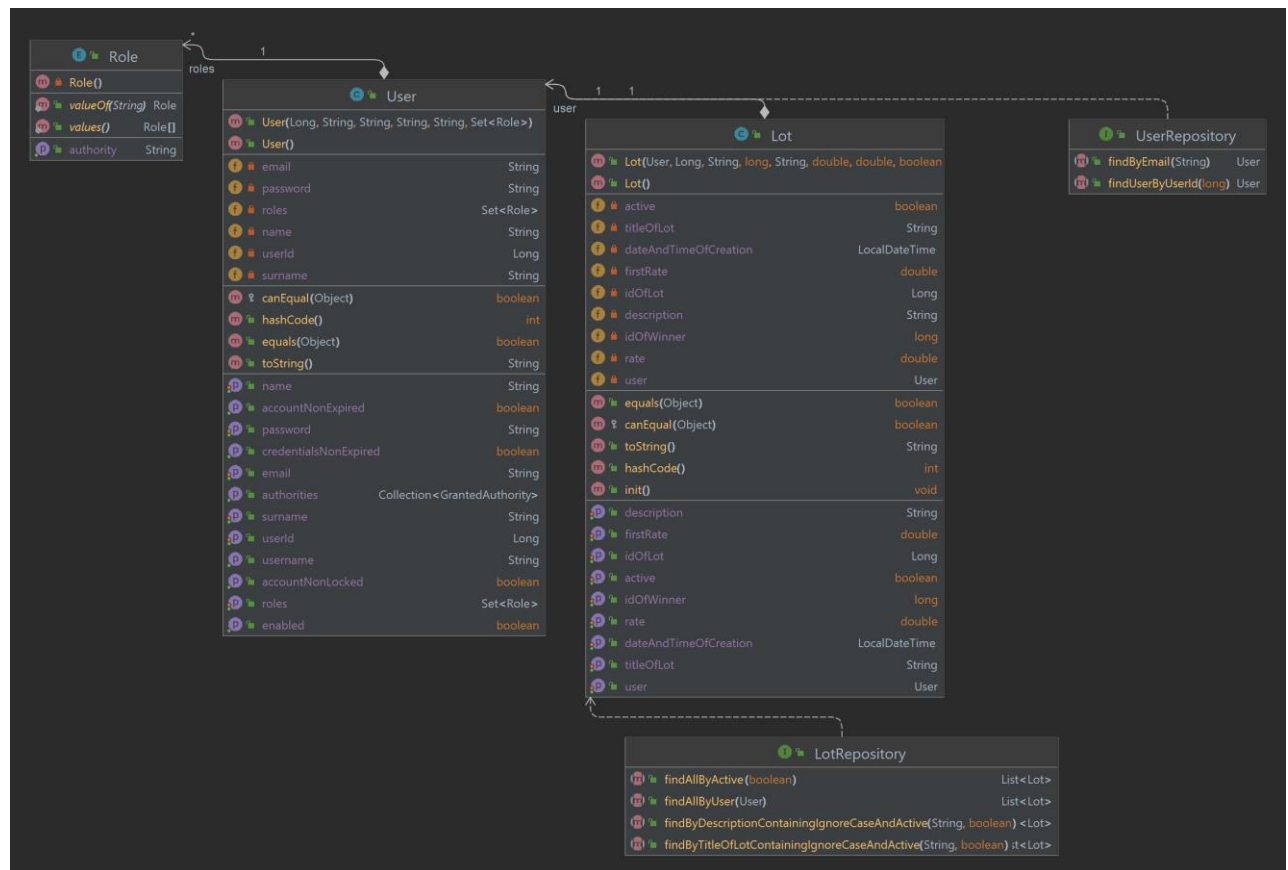


Рисунок 4.2 – Основні сутності та інтерфейси рівня доступу до даних

Клас "User" є сутністю, що представляє користувача в системі. Він містить такі поля, як "userId", "email", "password", "name", "surname" та "roles". Цей клас є сутністю (entity), означеною анотацією "@Entity". Анотація "@Data" генерує стандартні методи "equals()", "hashCode()", "toString()", "getters" та "setters" для всіх полів класу. "@NoArgsConstructor" та "@AllArgsConstructor" генерують конструктори без аргументів і конструктор, що приймає всі поля класу в якості аргументів відповідно. Поля класу описуються за допомогою анотацій "@Id", "@GeneratedValue", "@Column", "@ElementCollection". "@Id" вказує, що поле "userId" є ідентифікатором об'єкта. "@GeneratedValue" вказує стратегію генерації значення для поля "userId". У даному випадку використовується "GenerationType.IDENTITY", що означає, що значення генерується автоматично

базою даних. "@Column" визначає назву колонки бази даних, що відповідає полю. "@ElementCollection" означає, що поле "roles" є колекцією елементів, що зберігаються в окремій таблиці бази даних. "@CollectionTable" вказує назву таблиці, в якій зберігатимуться елементи колекції, а "@JoinColumn" визначає зовнішній ключ для зв'язку з об'єктом "User". "@Enumerated(EnumType.STRING)" вказує, що елементи перерахування "Role" зберігатимуться як рядки в базі даних.

Клас "**Lot**" використовується для представлення даних про певний лот. Цей клас також є сутністю, означеною анотацією "@Entity". Він представляє лот, що пропонується для аукціону. "@ManyToOne" вказує, що поле "user" є зв'язком багато до одного з класом "User". За допомогою анотації "@JoinColumn" визначається колонка, що зберігає зовнішній ключ для зв'язку. Поля класу описуються за допомогою анотацій "@Id", "@GeneratedValue", "@Column", "@PrePersist". "@PrePersist" вказує на метод "init()", який буде викликаний перед збереженням об'єкта в базі даних. У даному випадку, метод встановлює поточний час і дату в поле "dateAndTimeOfCreation".

Перерахування "**Role**": Це перерахування, яке імплементує інтерфейс "GrantedAuthority". Воно представляє ролі користувача системи. Метод "getAuthority()" повертає назву ролі.

Клас "LotRepository": Цей публічний інтерфейс розширяє "JpaRepository<Lot, Long>". Він надає додаткові методи для роботи з репозиторієм, який зберігає об'єкти типу "Lot". Декілька методів, таких як "findAllByActive", "findAllByUser", "findByTitleOfLotContainingIgnoreCaseAndActive" і "findByDescriptionContainingIgnoreCaseAndActive", дозволяють здійснювати пошук лотів за різними критеріями, такими як активність, користувач або ключове слово в описі.

Клас "UserRepository" також є публічним інтерфейсом, який розширяє "JpaRepository<User, Long>". Цей інтерфейс надає методи для роботи з репозиторієм, який зберігає об'єкти типу "User". Методи "findByEmail" і "findUserByUserId" дозволяють знаходити користувачів за їх електронною поштою або ідентифікатором користувача.

4.3 Компоненти рівня бізнес-логіки

Основні компоненти рівня бізнес-логіки наведені на рис. 4.3

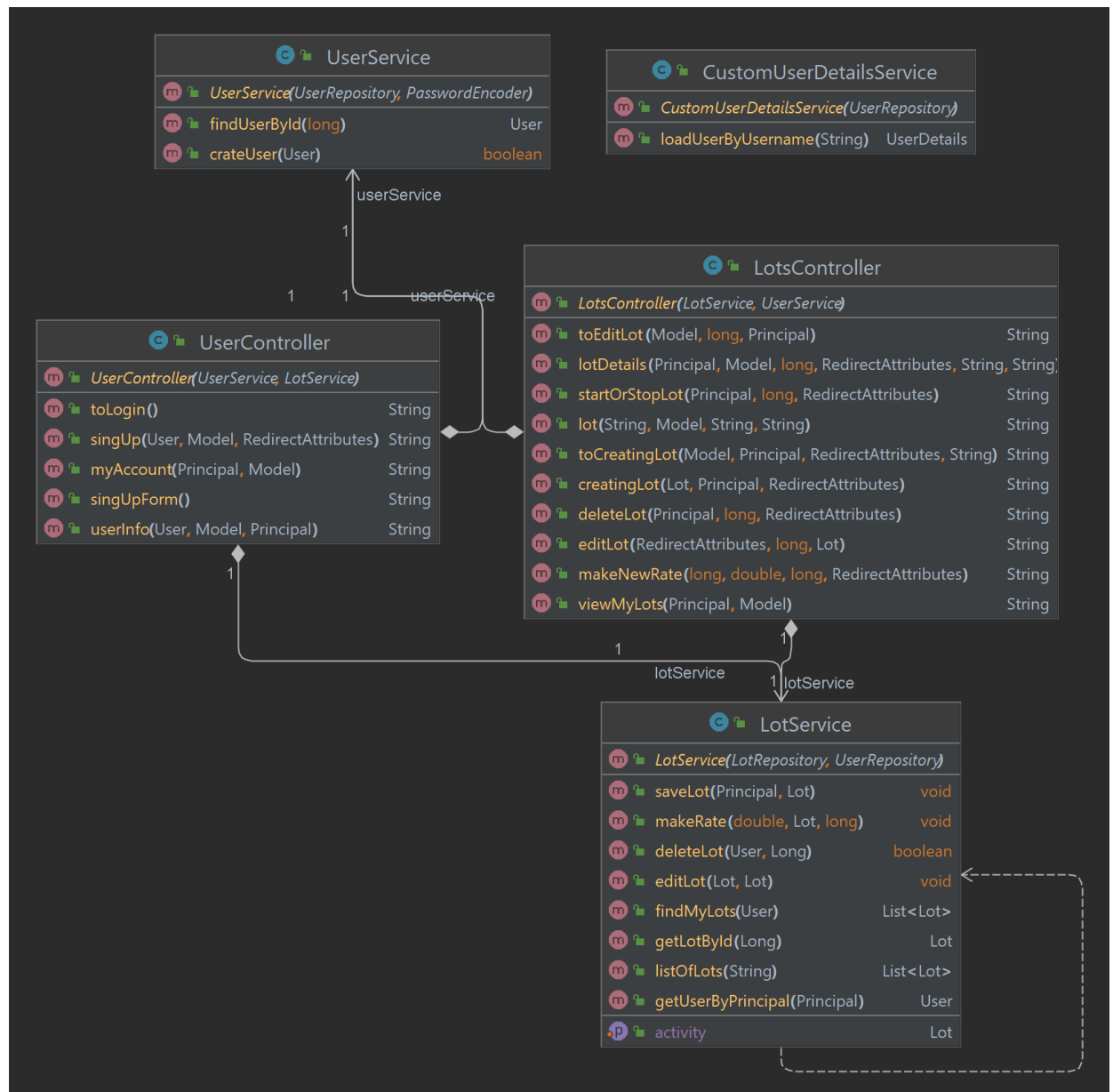


Рисунок 4.3 – Основні компоненти рівня бізнес-логіки

Клас “LotsController” є класом-контролером і містить методи для обробки запитів, пов'язаних з лотами. Він має анотацію “@Controller”, що позначає його як контролер веб-сторінок. У класі “LotsController” ми бачимо наступні елементи: `private final LotService lotService;` і `private final UserService userService;` - приватні поля, які представляють сервіси (“LotService” і “UserService”), які використовуються для виконання бізнес-логіки пов'язаної з лотами та користувачами.

Клас “UserController” є класом-контролером і містить методи для обробки запитів, пов'язаних з користувачами. Він також має анотацію “@Controller” і містить наступний елемент “private final UserService userService;” - приватне поле, яке представляє сервіс (“UserService”), використовується для роботи з користувачами.

У цих класах використовуються анотації “@RequiredArgsConstructor”, які генерують конструктор, який автоматично встановлює залежності за допомогою введених фінальних полів.

Клас “CustomUserDetailsService”: Цей клас реалізує інтерфейс “UserService” та відповідає за завантаження деталей користувача за його електронною поштою. Має залежність від репозиторію “UserRepository”, який використовується для отримання користувачів з бази даних. Метод “loadUserByUsername” використовує репозиторій для отримання користувача за його електронною поштою. Якщо користувач не знайдений, викидається виняток “UsernameNotFoundException”.

Клас “LotService”: Цей клас містить методи, які виконують операції пов'язані з лотами. Має залежності від репозиторію “LotRepository” та “UserRepository”. Метод “listOfLots” повертає список лотів з можливістю фільтрації за ключовим словом. Метод “saveLot” зберігає новий лот в базі даних, встановлюючи необхідні поля, такі як користувач, активність, початкова ставка. Метод “makeRate” встановлює нову ставку та ідентифікатор переможця для вказаного лоту. Метод “editLot” редагує існуючий лот, змінюючи заголовок, опис та початкову ставку, якщо вони відрізняються від поточних значень. Також змінюється ставка та ідентифікатор переможця, якщо необхідно. Метод “setActivity” змінює статус активності лоту. Методи “getUserByPrincipal”, “deleteLot”, “getLotById” та “findMyLots” надають додаткову функціональність для роботи з користувачами, зображеннями та лотами.

Клас “UserService”: Цей клас містить методи, пов'язані з операціями над користувачами. Має залежність від репозиторію “UserRepository”, класу “LotService” та “PasswordEncoder”. Метод “findUserById” повертає користувача за його ідентифікатором. Метод “createUser” створює нового користувача, зберігає його в базі даних, кодує пароль та надає роль користувача

4.4 Компоненти рівня інтерфейсу користувача

Основні компоненти рівня інтерфейсу користувача наведені на рис. 4.4

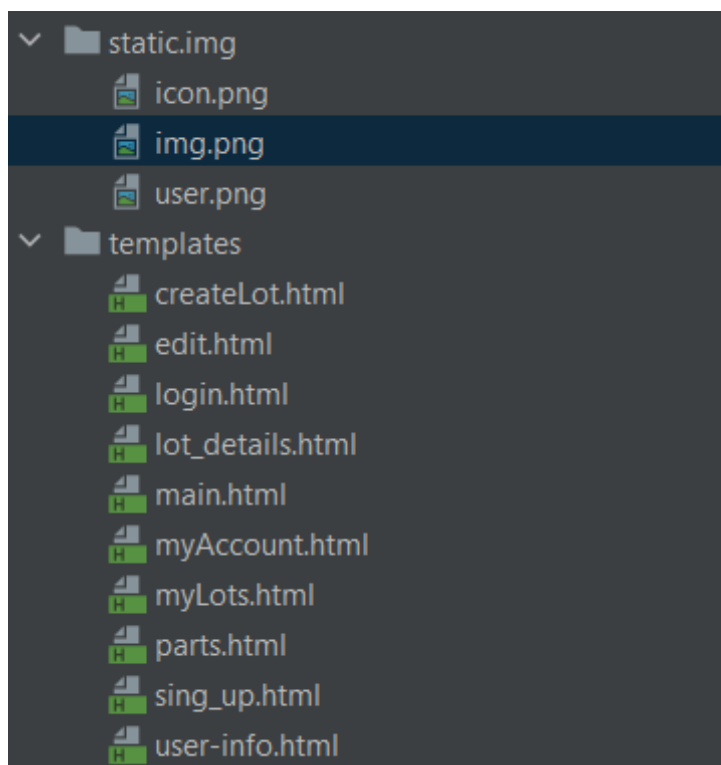


Рисунок 4.4 – Основні компоненти рівня інтерфейсу користувача

“createLot.html” – сторінка створення лота.

“edit.html” – сторінка редагування лота.

“login.html” – сторінка входу.

“lot_details” – сторінка, на якій відображена інформація про певний лот.

“main.html” – сторінка, на якій відображені всі лоти.

“myAccount.html” – сторінка авторизованого користувача.

“myLots.html” – сторінка, на якій відображені лоти авторизованого користувача.

“parts.html” – header and footer.

“sing_up.html” – сторінка реєстрації.

“user-info.html” – сторінка на якій відображено дані користувача і його лоти.

Також мною було використано шаблони з Bootstrap[1].

ВИСНОВКИ

Під час розробки системи проведення аукціонів було проведено аналіз ринку CRM-систем в Україні, вивчено переваги та недоліки існуючих систем[2, 3]. На основі цього аналізу було прийнято рішення про створення власної системи, яка надаватиме користувачам можливість проводити аукціони.

У процесі розробки системи було сформульовано функціональні та нефункціональні вимоги, які визначили очікувану поведінку системи. Для написання системи були обрані такі технології: Java як мова програмування, IntelliJ IDEA як середовище розробки, PostgreSQL як система управління базами даних, Hibernate[4] та Spring Data[5, 6, 7] для роботи з базою даних, Spring Security[6, 7, 8] для налаштування безпеки, HTML та Thymeleaf[9] для графічного інтерфейсу.

Також були описані сценарії використання системи, включаючи перегляд активних лотів, перегляд інформації про лот, можливість робити ставки, створення/видалення лота, редагування лота, зупинка/початок торгів.

ER-модель була розроблена для встановлення відповідності між базою даних і Java-кодом за допомогою Hibernate[4]. Були створені класи-сутності з необхідними полями та зв'язками з іншими сутностями.

У підсумку, система проведення аукціонів має декілька переваг, які роблять її привабливою для використання. По-перше, вона забезпечує механізми захисту конфіденційності, цілісності та доступності даних, гарантуючи безпеку і надійність інформації. Крім того, система демонструє швидку відповідь на дії користувачів, що забезпечує плавне та ефективне взаємодію з нею. Для зручного користування системою, вона має веб-інтерфейс, який є інтуїтивно зрозумілим та зручним для користувачів. Графічний інтерфейс є простим у налаштуванні, що сприяє швидкому впровадженню та використанню системи без додаткових зусиль. Крім того, система є крос-платформною, що дозволяє її використання на різних пристроях та операційних системах. Враховуючи ці переваги, система проведення аукціонів є конкурентоспроможною та має потенціал для подальшого розвитку, задовольняючи вимоги щодо захисту даних, швидкої відповіді, відкритої архітектури, зручного інтерфейсу та крос-платформовості.

ПЕРЕЛІК ВИКОРИСТАНИХ ДЖЕРЕЛ

1. Bootstrap [Електронний ресурс] – Режим доступу до ресурсу: <https://getbootstrap.com/docs/5.3/examples/>
2. Аукціони системи Прозорро [Електронний ресурс] – Режим доступу до ресурсу: <https://sale.uub.com.ua/>
3. Державні публічні аукціони в Україні [Електронний ресурс] – Режим доступу до ресурсу: <https://ubiz.ua/>
4. Односторонние и двусторонние отношения в Hibernate [Електронний ресурс] – Режим доступу до ресурсу: <https://habr.com/ru/articles/542328/>
5. Разработка сайта на Java Spring Boot [Електронний ресурс] – Режим доступу до ресурсу: https://www.youtube.com/playlist?list=PL0lO_mIqDDFUYDRzvocu5EsFGBqPM7CIw
6. Spring Boot. Разработка веб-приложения с нуля [Електронний ресурс] – Режим доступу до ресурсу: <https://www.youtube.com/playlist?list=PLMN3dELi3-VdVo8NGOYqaOLC2alq6ZmSh>
7. Spring Guides [Електронний ресурс] – Режим доступу до ресурсу: <https://spring.io/guides>
8. Spring Boot User Registration and Login Example Tutorial [Електронний ресурс] – Режим доступу до ресурсу: <https://www.javaguides.net/2018/10/user-registration-module-using-springboot-springmvc-springsecurity-hibernate5-thymeleaf-mysql.html>
9. Учебник: Использование Thymeleaf [Електронний ресурс] – Режим доступу до ресурсу: <https://habr.com/ru/articles/350862/>