

МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ
Національний університет «Запорізька політехніка»

Інститут інформатики та радіoeлектроніки,
Факультет комп'ютерних наук і технологій

(повне найменування інституту, назва факультету)

Кафедра програмних засобів

(повне найменування кафедри)

Пояснювальна записка

до дипломного проєкту (роботи)

магістр

(ступінь вищої освіти)

на тему ДОСЛІДЖЕННЯ ТА ПРОГРАМНА РЕАЛІЗАЦІЯ МЕТОДІВ
ПІДТРИМКИ ПРОВЕДЕННЯ АУКЦІОНУ

RESEARCH AND SOFTWARE IMPLEMENTATION OF METHODS FOR
AUCTION SUPPORT

Виконав: студент(ка) 2 курсу, групи КНТ-119м
Спеціальності 121 Інженерія програмного
забезпечення

(код і найменування спеціальності)

Освітня програма (спеціалізація)

Інженерія програмного забезпечення

Кузнецов В.В.

(прізвище та ініціали)

Керівник Скрупський С.Ю.

(прізвище та ініціали)

Рецензент Зеленьова І.Я.

(прізвище та ініціали)

МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ
Національний університет «Запорізька політехніка»
(повне найменування закладу вищої освіти)

Інститут, факультет ІРЕ, ФКНТ
Кафедра програмних засобів
Ступінь вищої освіти магістр
Спеціальність 121 Інженерія програмного забезпечення
(код і найменування)
Освітня програма (спеціалізація) Інженерія програмного забезпечення
(назва освітньої програми (спеціалізації))

ЗАТВЕРДЖУЮ

Завідувач кафедри ПЗ, д.т.н, проф.
С.О. Субботін
“ ” 2020 року

З А В Д А Н Н Я

НА ДИПЛОМНИЙ ПРОЄКТ (РОБОТУ) СТУДЕНТА(КИ)

Кузнецова Владислава Вікторівна

(прізвище, ім'я, по батькові)

1. Тема проєкту (роботи) Дослідження та програмна реалізація методів
підтримки проведення аукціону

Research and Software Implementation of Methods for Auction Support

керівник проєкту (роботи) Скрупський Степан Юрійович, к.т.н., доцент,
(прізвище, ім'я, по батькові, науковий ступінь, вчене звання)

затверджені наказом закладу вищої освіти від “3” листопада 2020 року № 301

2. Строк подання студентом проєкту (роботи) 10 грудня 2020 року

3. Вихідні дані до проєкту (роботи) рекомендована література, технічне
завдання

4. Зміст розрахунково-пояснювальної записки (перелік питань, які потрібно розробити) 1. Аналіз проблеми та постановка завдань дослідження.
2. Обґрунтування методу підтримки проведення аукціону. 3. Опис програми.
4. Розробка програми. 5. Експлуатація та тестування програми.
6. Економіко-організаційна частина, 7. Охорона праці та безпека в
надзвичайних ситуаціях.

5. Перелік графічного матеріалу (з точним зазначенням обов'язкових креслень) Слайди презентації

6. Консультанти розділів проєкту (роботи)

Розділ	Прізвище, ініціали та посада консультанта	Підпис, дата	
		завдання видав	прийняв виконане завдання
1-5 Основна частина	Скрупський С.Ю., доцент		
Економіко-організаційна частина	Пожуєва Т.О., професор		
Охорона праці та безпека в надзвичайних ситуаціях	Коробко О.В., ст. викладач		
Нормоконтролер	Калініна М.В., асистент		

7. Дата видачі завдання “ 7 ” вересня 2020 року.

КАЛЕНДАРНИЙ ПЛАН

№ з/п	Назва етапів дипломного проєкту (роботи)	Строк виконання етапів проєкту (роботи)	Примітка
1	Постановка завдання роботи.	1 тиждень	Завдання, ТЗ
2	Аналіз предметної області.	2–3 тижні	Розділ 1
3	Розробка методу підтримки проведення аукціону.	4–5 тижні	Розділ 2
4	Розробка архітектури програми.	6 тиждень	Розділ 3
5	Розробка програми.	7–8 тижні	Розділ 4
6	Тестування та відлагодження програмного забезпечення.	9 тиждень	Розділ 5
7	Розробка організаційно-економічної частини	10 тиждень	Розділ 6
8	Розробка заходів з охорони праці та цивільної безпеки.	11 тиждень	Розділ 7
9	Оформлення пояснювальної записки та документів до неї.	12-13 тиждень	Додатки
10	Нормоконтроль та рецензування.	14–15 тижні	
11	Захист роботи.	16 тиждень	

Студент(ка)

(підпис) Кузнецов В.В.
(прізвище та ініціали)

Керівник проєкту (роботи)

(підпис) Скрупський С.Ю.
(прізвище та ініціали)

РЕФЕРАТ

Пояснювальна записка до дипломної кваліфікаційної роботи магістра:
121 с., 32 рис., 8 табл., 3 дод., 30 джерел.

ЕЛЕКТРОННИЙ АУКЦІОН, ВИЗНАЧЕННЯ ВАРТОСТІ ЛОТУ,
ПОДІБНІ ЗОБРАЖЕННЯ, ЗГОРТКОВІ НЕЙРОННІ МЕРЕЖІ, МОВА C#.

Об'єкт дослідження – процес проведення аукціону. Предмет дослідження – програмне забезпечення підтримки проведення аукціону.

Мета роботи – розроблення програмного забезпечення підтримки проведення аукціону для забезпечення обґрунтованого прийняття рішень щодо вибору початкових параметрів лотів.

Матеріали, методи та технічні засоби: об'єктно-орієнтоване програмування, мова програмування C#, система керування базами даних Microsoft SQL Server.

Результати. Виконано аналіз проблеми зі шляхами її вирішення, розроблено власний метод вирішення проблеми, виконано проєктування та реалізацію програмного забезпечення для вирішення розглядаємої проблеми на практиці.

Висновки. Наукова новизна роботи полягає в запропонованому методі підтримки проведення аукціону, проведення якого ґрунтується на моделі відкритого багатораундового аукціону, який використовує згорткову нейронну мережу на основі моделі архітектури VGG-19 для виділення подібних зображень лотів, що дозволяє узгоджувати початкову вартість лоту з подібними раніше проданими лотами, отримувати оцінку фактичної вартості лоту продавцям та знаходити подібні лоти для забезпечення синхронності аукціону для покупців.

Галузь використання – електронні аукціони товарів, проведення закупок товарів і послуг.

ABSTRACT

Explanatory note to the diploma qualifying work of the master: 121 pages, 32 figures, 8 tables, 3 appendixes, 30 sources.

ELECTRONIC AUCTION, LOT VALUE DETERMINATION, SIMILAR IMAGES, CONVOLUTIONAL NEURAL NETWORKS, LANGUAGE C#.

The object of research is the auctioneering process. The subject of research is the software for auction support.

The purpose of the work is to develop software to support the auction to ensure informed decision-making on the choice of initial parameters of lots.

Materials, methods and technical means: object-oriented programming, C# programming language, Microsoft SQL Server database management system.

Results. The analysis of a problem with ways of its decision is executed, the own method of the solution of the problem is developed, design and implementation of the software for the decision of the solution of the problem in practice is executed.

Conclusions. The scientific novelty of the work lies in the proposed method of supporting the auction, which is based on the open multi-round auction model, which uses a convolutional neural network based on the architecture model VGG-19 to select similar images of lots, which allows to match the initial price of the lot, obtain an estimate of the actual price of the lot to sellers and find similar lots to ensure the synchronicity of the auction for buyers.

Field of use is electronic auctions of goods, tenders of goods and services.

ЗМІСТ

	С.
Перелік скорочень та умовних позначок	8
Вступ.....	9
1 Аналіз проблеми та постановка завдань дослідження	11
1.1 Електронні аукціони	11
1.2 Огляд програмних засобів для проведення аукціонів.....	16
1.3 Постановка завдань дослідження	20
1.4 Висновки за розділом 1	20
2 Обґрунтування методу підтримки проведення аукціону.....	21
2.1 Методи підтримки аукціонної діяльності.....	21
2.2 Методи пошуку подібних зображень.....	23
2.3 Згорткові нейронні мережі	24
2.4 Нейронна мережа VGG-19	26
2.5 Метод підтримки проведення аукціону.....	27
2.6 Висновки за розділом 2	29
3 Опис програми.....	30
3.1 Вибір мови програмування	30
3.2 Моделювання функціональних вимог до програмного забезпечення...	31
3.3 Проєктування структури бази даних.....	34
3.4 Проєктування структури програмного застосунку	38
3.5 Висновки за розділом 3	40
4 Розробка програми	41
4.1 Алгоритм функціонування програми.....	41
4.2 Опис розроблених модулів.....	42
4.3 Висновки за розділом 4	50
5 Експлуатація та тестування програми	51
5.1 Призначення програми	51
5.2 Умови виконання програми	51
5.3 Виконання програми.....	51

6 Економіко-організаційна частина.....	59
6.1 Планування розробки програмного продукту.....	59
6.2 Визначення витрат на розробку програмного продукту.....	62
6.3 Розрахунок витрат на оплату праці.....	62
6.4 Визначення витрат на матеріали	65
6.5 Визначення витрат на спеціальне устаткування.....	66
6.6 Визначення накладних витрат	71
6.7 Оцінка економічної ефективності	73
7 Охорона праці та безпека в надзвичайних ситуаціях.....	74
7.1 Аналіз потенційних небезпек.....	74
7.2 Заходи по забезпеченню безпеки	74
7.3 Заходи з виробничої санітарії і гігієни праці	76
7.4 Заходи безпеки у надзвичайних ситуаціях	82
7.4.1 Заходи з пожежної безпеки.....	82
7.4.2 Заходи з цивільного захисту	83
Висновки	84
Перелік джерел посилання	86
Додаток А Технічне завдання	90
Додаток Б Текст програми	94
Додаток В Слайди презентації.....	114

ПЕРЕЛІК СКОРОЧЕНЬ ТА УМОВНИХ ПОЗНАК

- ОЦ – обчислювальний центр;
ПК – персональний комп'ютер;
ПУЕ – правила улаштування електроустановок.

ВСТУП

Загально відомо, що електронна торгівля в даний момент переживає свій бурхливий розвиток, особливо в країнах, що розвиваються і зокрема в умовах пандемії.

Ключові фактори, що сприяють розвитку електронної торгівлі, – зниження бізнес-витрат, підвищення конкурентоспроможності, в тому числі в умовах глобального поділу праці: пріоритетними сферами розвитку електронного бізнесу є сегменти електронної торгівлі за схемою «підприємство-приватна особа» та «системи електронної комерції між приватними особами» [1].

Інтернет дав можливість приєднатися до торгів величезній кількості людей, практично будь-кому, і кожен бажаючий отримав можливість виставити на торги будь-який товар, починаючи від особистих не потрібних йому речей до колекційних, ця можливість привернула багато людей на аукціони, відповідно аукціони стали бурхливо розвиватися [2].

Мета інтернет-аукціону – не продати, а звести разом покупців і продавців, онлайн-аукціони представляють зручний канал продажів для індивідуальних торговців і дрібних компаній, великі компанії також використовують аукціони для того, щоб швидко збути товари, які накопичилися на складі, інтернет-аукціони можна використовувати для тестових продажів, щоб визначити величину початкового попиту і ринкову ціну для нового продукту [3].

Тому можна говорити про актуальність проблеми підтримки проведення аукціону, адже окрім безпосередньо створення майданчика для реалізації аукціону важливою є і підтримка торгів на рівні методологічного апарату, адже результат торгів залежить не тільки від учасників, але і від того, які параметри лотів були початково виставлені.

У результаті в якості мети роботи було обрано розроблення програмного забезпечення підтримки проведення аукціону для забезпечення обґрунтованого прийняття рішень щодо вибору початкових параметрів лотів.

1 АНАЛІЗ ПРОБЛЕМИ ТА ПОСТАНОВКА ЗАВДАНЬ ДОСЛІДЖЕННЯ

1.1 Електронні аукціони

Електронний аукціон – нова форма проведення державних і комерційних тендерів, яка витіснила торги в звичайних аукціонних залах: електронні аукціони створюються для зручності учасників і головне – для усунення корупції, всесвітні торгові майданчики та аукціони являють собою нове покоління багатопрофільних електронних торговельних майданчиків, які об'єднують десятки різних видів товарів і послуг як для експортно-імпортних компаній, так і для оптових і роздрібних клієнтів в різних регіонах світу, відносно яких можна виділити наступні факти:

- аукціон eBay, який заснований в 1995 році і має оборот в декілька мільйонів угод в день, є найбільшим інтернет-аукціоном в світі: інтернет-аукціон («онлайновий аукціон») – аукціон, який проводиться за допомогою Інтернету на спеціалізованих сайтах (або сайт-аукціон, або електронна торговельна площадка), на відміну від звичайних аукціонів, інтернет-аукціони проводяться на відстані (дистанційно) і в них можна брати участь не перебуваючи в місці проведення, тобто віддалено, роблячи ставки через інтернет-сайт або комп'ютерну програму аукціону;

- проведення електронних аукціонів є досить поширеним явищем: так, зокрема, в 2007 році корпорація «Fozzy Group», згідно зі стратегією поліпшення послуг для своїх клієнтів, оголосила конкурсний відбір банку-партнера для надання йому ексклюзивних прав з надання фінансових послуг на території магазинів корпорації, для відбору кращої пропозиції було проведено відкритий електронний онлайн-аукціон серед усіх зацікавлених банків [1].

Існують наступні типи аукціонів.

Закритий аукціон з одним раундом, коли учасники не знають про розмір ставки (або ряд ставок) інших учасників, результати оголошуються

тільки після закінчення торгів, і лот видається учаснику, який запропонував максимальну ціну, відзначається наступним:

- проста, швидка, легко керована форма торгів;
- аукціони можуть використовуватися для продажу як одного, так і декількох товарів (ліцензій);
- немає необхідності збирати учасників в одному місці, використовувати складні паперові або електронні процедури проведення торгів;
- результати аукціону просто інтерпретувати;
- запобігається можливість змови учасників;
- недоліком є те, що в загальному випадку у учасників немає можливості аналізувати дії один одного – це призводить до того, що учасникам складно прийняти рішення, на який з лотів робити ставку, коли на торги виставлено декілька рівнозначних товарів (ліцензій);
- недоліком є відсутність впевненості у виграші того чи іншого лота, якщо на торги виставляється декілька лотів, що доповнюють один одного [4].

Багатораундовий синхронний аукціон з підвищенням ставок дуже широко використовується при продажу лотів, до його особливостей належать наступні умови проведення: на торги, які проходять в декілька раундів, виставляються декілька товарів (ліцензій), що одночасно продаються, в кожному раунді учасники підвищують свої ставки, і аукціон завершується, коли припиняється оголошення нових ставок, лот віддається учаснику, який запропонував максимальну ціну, при організації відкритого багатораундового синхронного аукціону з підвищенням ставок можуть застосовуватися різні правила, стандартний варіант такої форми торгів передбачає продаж декількох лотів, ціна на кожен з яких піднімається тільки тоді, коли будь-хто з учасників робить на нього ставку, таким чином, в процесі торгів ціни на окремі лоти змінюються, при цьому учасники можуть коригувати свої ставки, тобто почати робити ставки на той лот, ціна якого нижче, відкритий синхронний аукціон з підвищенням ставок ефективний,

коли на торги виставляється декілька ліцензій і існує невизначеність в оцінці їх вартості, відзначається наступним:

- можливість учасників аналізувати поведінку конкурентів і коригувати свої ставки в залежності від зміни ціни товару (ліцензії);
- менше ризик переплати переможця;
- менший ризик в неотриманні учасником потрібного товару (ліцензії);
- менший ризик в неотриманні одного з товарів (або ліцензії) з пакета, який необхідний учаснику;
- можливість змови учасників: кількість учасників торгів може бути зафіксовано ще до початку аукціону, крім того, діючі компанії можуть перешкодити участі в торгах новачків;
- складність: правила і процедури такого виду аукціону більш складні, ніж в разі проведення простого закритого аукціону, незважаючи на те, що розвиток електронних систем торгів зробив проведення синхронних аукціонів дешевше і простіше;
- нерівні умови для учасників: при такій формі проведення аукціону деякою перевагою володіють учасники, яким необхідний тільки один лот, в учасників, яким для ефективної роботи мережі потрібно декілька конкретних лотів, вище ризик не виграти їх;
- торги можуть займати багато часу [3]-[4].

Багатораундовий аукціон з пониженням ставок, до особливостей проведення якого відноситься наступне: перед початком аукціону аукціоніст оголошує стартову ціну лота, яка свідомо завищена, в разі відсутності учасників, готових придбати лот за вказаною ціною, аукціоніст знижує ціну; торги припиняються, коли знаходиться учасник, готовий придбати лот, відзначається наступним:

- простота реалізації і прозорість процедури аукціону;
- в порівнянні з закритим форматом у учасників є можливість аналізувати поведінку конкурентів, що підвищує ефективність їх рішень;

– аукціон може бути ефективний, коли метою аукціону є продаж лота за мінімальною ціною;

– недоліком є те, що організатор торгів може втратити частину доходів, якщо невідомо, яку максимальну ціну готові заплатити учасники, а мета аукціону – продати лот за максимально можливою ціною [4].

«Тактовий» аукціон – варіант відкритого багатораундового синхронного аукціону з підвищенням ставок, використовується для продажу лотів, які вважаються рівними за вартістю і при цьому учасники не можуть віддати перевагу одному лоту над іншим, в даному форматі на лоти встановлюється єдина ціна, яка піднімається з кожним раундом до тих пір, поки загальна кількість учасників торгів не знизиться до кількості лотів, в одному з варіантів такого виду торгів лоти об'єднуються в групи так, щоб в одній групі були рівнозначні для учасника лоти, у цьому випадку ціна встановлюється для одного лота кожної групи, «тактовий» аукціон – це циклічна процедура, при якій аукціоніст на початку раунду оголошує ціну за один лот кожної групи лотів, потім учасники оголошують, скільки лотів кожної групи вони готові придбати за вказаною ціною, для групи лотів, на які попит більше пропозиції, ціна за один лот підвищується, потім учасники знову оголошують, скільки лотів вони готові придбати за вказаними цінами, процес повторюється, поки попит по всіх групах лотів не зрівняється з пропозицією, відзначається наступним:

– є простим для учасників: у кожному раунді учасник повідомляє, який обсяг спектра він готовий придбати за вказаною ціною;

– виключена можливість змови учасників: обмеження доступної для учасників інформації про різницю між попитом і пропозицією не дозволяє розробляти стратегії змови, виключається подача сигналів за допомогою ставок, тому що учасники не володіють інформацією про ставки окремих учасників;

– велика ефективність: з кожним раундом учасники все більше володіють інформацією про ціну тієї чи іншої ділянки спектра, що

використовується при прийнятті рішень, в результаті витрати на участь в торгах знижуються, а ефективність зростає;

– недоліком є те, що лінійне ціноутворення в кінці торгів, лінійне ціноутворення стимулює зниження попиту учасників, що призводить до зниження ефективності результатів аукціону [3]-[4].

Комбінаторний відкритий багатораундовий аукціон: такий тип аукціону використовується, коли на торги виставляються лоти, що доповнюють один одного, на відміну від стандартного аукціону, учасники можуть робити декілька ставок, по одній на кожен комбінацію лотів, іншими словами, ставки можуть робитися як на окремий лот, так і на довільну комбінацію лотів, переможці визначаються наступним чином: за всіма учасниками за всіма лотами визначається комбінація ставок, сума яких максимальна, переможцями визнаються учасники, які зробили ставки, які увійшли в таку комбінацію, торги можуть бути організовані як в один раунд, так і в декілька раундів, такий підхід може бути доповненням до будь-якого формату проведення аукціону, відзначається наступним:

– перевага: можливість знизити ризики для учасників: можливість робити ставки на ряд лотів дозволяє уникнути ситуації, коли учасники не отримають частину необхідного пакета, це забезпечує інтереси як тих, хто хоче отримати декілька лотів (товарів, ліцензій), так і тих, хто хоче отримати один лот (товар, ліцензію);

– високі витрати і складність алгоритму визначення переможця: комбінаторний аукціон може бути значно складнішим для розуміння як організаторів, так і учасників торгів, коли на торги виставлено понад 4 лотів і в них бере участь значна кількість учасників, то для визначення переможця може знадобитися комп'ютерний алгоритм, це пов'язано з великою кількістю можливих комбінацій переможців і лотів, які необхідно розглядати, навіть при малому обсязі лотів визначення переможця – непросте завдання;

- низька прозорість: результати такого типу аукціонів не завжди прозорі для учасників і спостерігачів, особливо коли для визначення переможця необхідний комп'ютерний алгоритм;

- складність у прийнятті рішень: для учасників, які хочуть отримати конкретний лот або малий пакет лотів, така форма аукціону може привести до виникнення нового ризику – цим учасникам складно приймати правильні рішення, щоб конкурувати з тими, хто претендує на великий пакет лотів і може робити ставки таким чином, щоб обмежити можливості перших [4].

Комбінований аукціон: комбінованим називається будь-який аукціон, який об'єднує параметри декількох типів аукціонів, наприклад, англо-голандський, тактовий з можливістю робити ставки на пакет лотів, англо-голандський формат об'єднує стандартний одночасний багатораундовий аукціон і закритий аукціон [3]-[4].

1.2 Огляд програмних засобів для проведення аукціонів

eBay – це компанія, що займається онлайн-аукціоном, заснована П'єром Омідьяром в Каліфорнії 3 вересня 1995 року, при цьому для засновника спочатку цей проект був всього лише хобі, яке в підсумку приніс йому значний прибуток, незабаром проект перетворився в популярний сайт для продажу і покупки речей онлайн, на сьогоднішній день eBay користуються близько 171 мільйонів чоловік, за статистикою, за останні 8 років кількість користувачів збільшилася на 100 мільйонів чоловік, вартість товарів, проданих на цьому майданчику за рік, зараз перевищує 17.9 мільярдів доларів, а на рекламі eBay заробив ще 3 мільярди доларів, при цьому програмний застосунок сам по собі є достатньо простим і щодо його функціонування можна виділити наступні особливості:

- продавець перераховує товари на eBay, при чому до цих товарів може входити майже все – від антикваріату до автомобілів, від книг до спортивних товарів, а далі вирішує, чи приймати лише заявки на товар

(перелік у стилі аукціону), чи запропонувати опцію «Купуй зараз», що дозволяє покупцям придбати товар відразу за фіксованою ціною (рис. 1.1);

– на онлайн-аукціоні торги відкриваються за ціною, визначеною продавцем, і залишаються на eBay протягом певної кількості днів, покупці роблять заявки на товар, коли лістинг закінчується, виграє покупець із найбільшою ставкою;

– у списку «Купуй зараз» («Buy It Now») перший товар купує той, хто бажає заплатити ціну продавця [5]-[6].

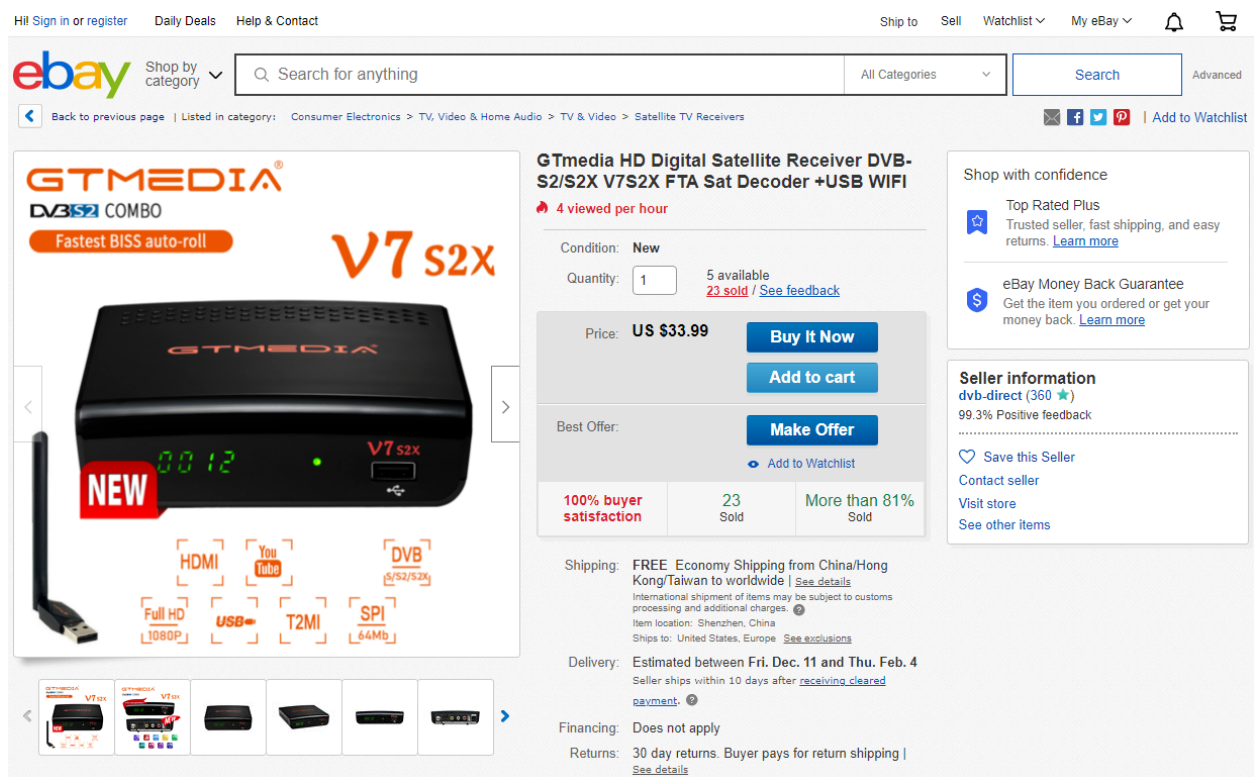


Рисунок 1.1 – Сторінка перегляду товару аукціону eBay

Є багато аналогічних програмних засобів, які за основною своєю функціональністю і головне – за способом організації онлайн-аукціону – повторюють логіку роботи eBay. Зокрема в Україні одним з таких майданчиків є SkyLots (рис. 1.2).

Violity – майданчик для продажу і покупки антикваріату та інших ексклюзивних предметів колекціонування, що був заснований в Україні в 2005 році, щодня на Violity продаються і купуються тисячі лотів, на сьогодні

на Viology, за даними адміністрації аукціону, більше 450 000 активних продавців і більше 400 000 активних покупців, зростання становить близько 1 % в місяць (це 10 000 нових користувачів), в день на Viology виставляється більше 200 000 лотів [7].

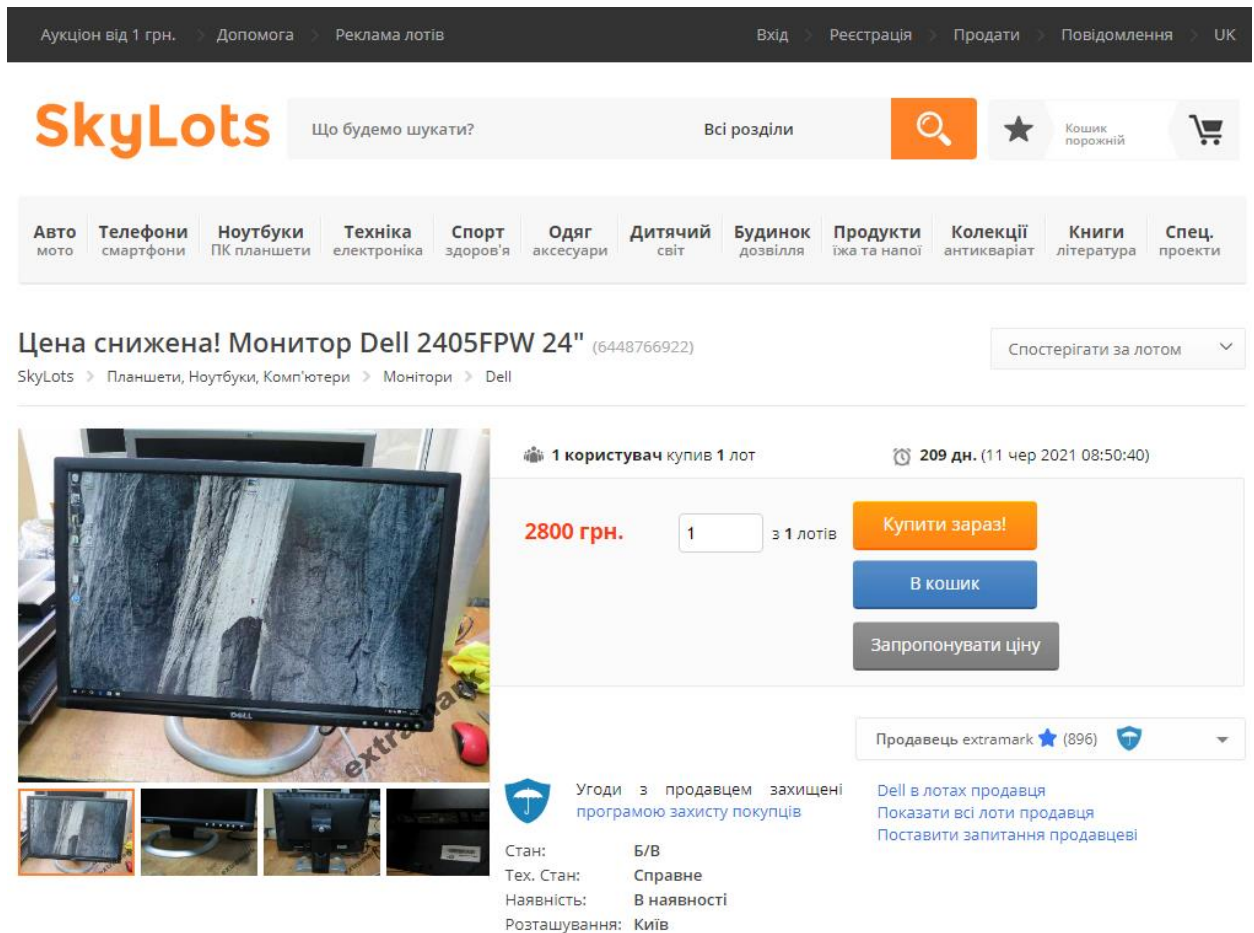


Рисунок 1.2 – Сторінка аукціону SkyLots

Правила, які визначають торги на майданчику Viology (рис. 1.3):

– якщо користувач розмістив лот на аукціон і встановив на нього мінімальну ціну, то він не має права передумати і достроково зняти предмет з продажу – ціну можна змінити, лише якщо по лоту ще немає жодної ставки, якщо мінімальна ставка є і її ніхто не перебіє, користувач повинен буде продати покупцеві цей предмет за встановленою мінімальною ціною;

- продавець зобов'язаний дати відповідь покупцеві протягом 72 годин (3 діб) після завершення торгів і обумовити деталі оплати і доставки, відправку продавець повинен зробити протягом 3 днів після оплати;
- всі комунікації здійснюються тільки всередині платформи через спеціальний чат, який відкривається тільки після завершення торгів між продавцем і покупцем-переможцем;
- обов'язки є і у покупців: вони не можуть скасувати вже зроблену ставку і в разі перемоги зобов'язані купити товар, оплату необхідно перевести не пізніше, ніж через три дні з моменту отримання реквізитів;
- розірвати угоду без участі адміністрації майданчика сторони не можуть [7].

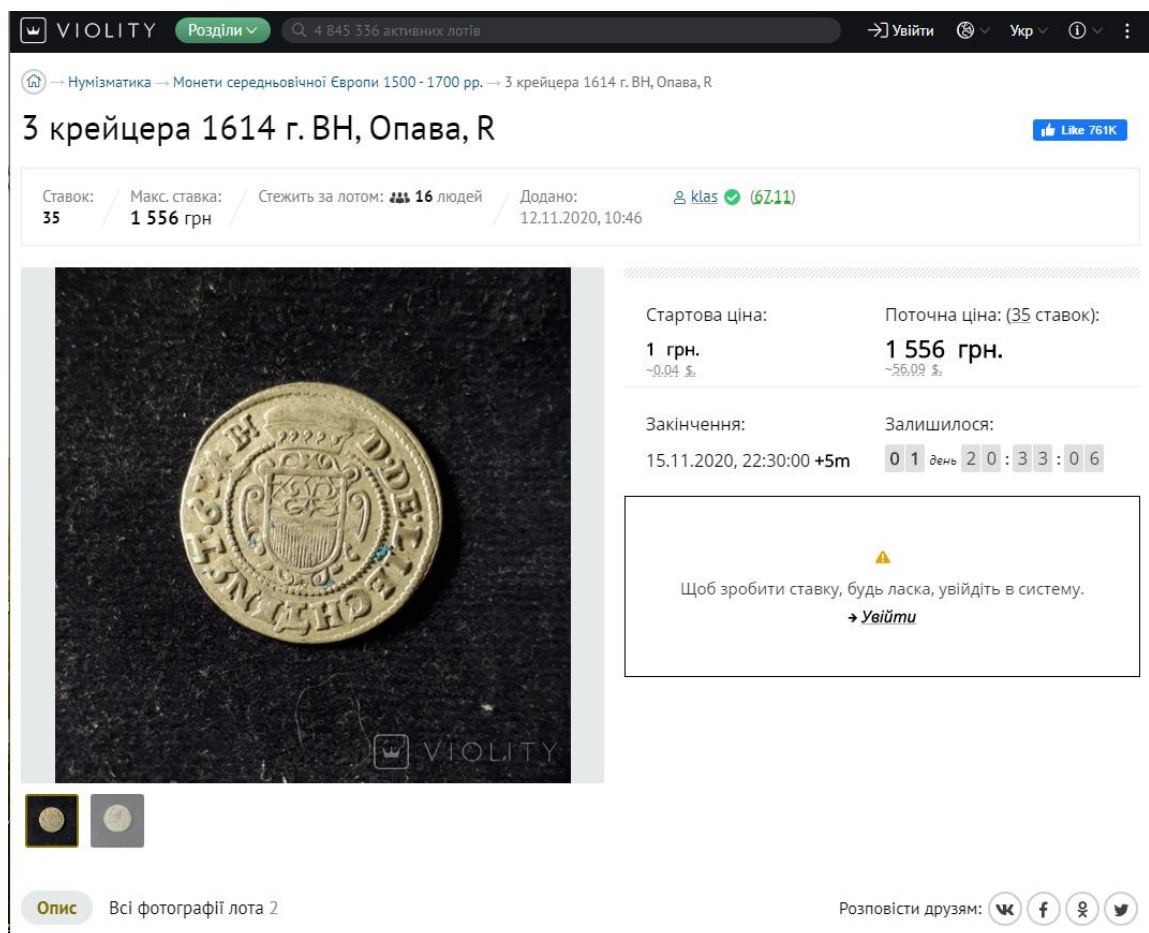


Рисунок 1.3 – Сторінка аукціону Violity

1.3 Постановка завдань дослідження

Зважаючи на встановлену мету роботи і отримані результати аналізу проблеми, було поставлено наступні завдання дослідження:

- розробка методу підтримки проведення аукціону;
- розробка архітектури програми;
- розробка програми;
- розробка організаційно-економічної частини;
- розробка заходів з охорони праці та цивільної безпеки.

1.4 Висновки за розділом 1

За результатами виконання розділу 1 було отримано наступні результати:

- розглянуто проблему проведення електронних аукціонів;
- проаналізовано існуючі типи аукціонів;
- виконано огляд програмних засобів для проведення аукціонів;
- визначено завдання дослідження.

2 ОБГРУНТУВАННЯ МЕТОДУ ПІДТРИМКИ ПРОВЕДЕННЯ АУКЦІОНУ

2.1 Методи підтримки аукціонної діяльності

Методи підтримки аукціонної діяльності стосуються різних етапів проведення аукціонів. Частина з них може визначати процес проведення аукціону (модель), але загалом вони достатньо відомі і були розглянуті в розділі 1. Інша частина методів направлена на забезпечення визначення певних показників, необхідних під час проведення аукціону.

Однією з основних економічних складових ефективної аукціонної діяльності є ціноутворення: при всій його різноманітності центральне місце займає науково обґрунтована оцінка об'єктів аукціонів і визначення стартової ціни, це досить складне завдання і в теоретичному відношенні, і у втіленні на практиці, підтвердженням служить значний розрив між розрахунковими оцінками вартості об'єктів, стартовими цінами і цінами, за якими вони дійсно продаються, звідси доцільно зупинитися на стадіях і методах визначення оціночної вартості і стартових цін, дані розрахунки специфічні для різних об'єктів аукціону, у теорії виділяють і на практиці використовують ряд стадій даного процесу, з яких можна виділити зокрема наступні:

- перший етап полягає у формуванні мети і функції оцінки – саме він зумовлює подальший вибір методики оцінки, неправильно сформульована мета тягне за собою використання неадекватного методу і внаслідок цього помилковий результат, оцінка може вестися для оподаткування, страхування, кредитування, оренди, ліквідації або для визначення стартової ціни;

- попередній огляд об'єкта починається з вибору експерта-оцінювача: оцінка невеликого об'єкта нерухомості може здійснюватися їм одним, для оцінки великого об'єкта експерт-оцінювач пропонує групу експертів в складі економістів, маркетологів, технічних експертів, екологів, фахівців інших профілів, для цієї мети також необхідно залучати представників замовника,

які мають необхідні знання та досвід, експерти визначають обсяг і склад інформаційного матеріалу, якість і зміст якого істотно впливає на результат;

– вибір методів оцінки об'єкта: у ринкових умовах господарювання для оцінки вартості об'єктів нерухомості використовуються переважно три загальновизнаних методи (дохідний, метод прямого порівняльного аналізу продажів і витратний), на основі яких і визначається необхідна вартість об'єкта, для оцінки об'єктів необхідно їх згрупувати за сферами діяльності (виробнича, невиробнича, торгівля), з цієї точки зору різні за своїм призначенням об'єкти повинні оцінюватися за різними методами і складеним на їх основі методикам, зокрема відповідно до прибуткового методу вартість об'єкта оцінки встановлюється по його прибутковості, загальна вартість майна, активів розраховується як сума оцінок вартості їх від відповідних видів, сума доходу може обчислюватися як загальна їх сукупність або шляхом підсумовування доходів за окремими видами активів, для використання цього методу необхідні наступні дані: балансова вартість активів і очікувана прибутковість від їх використання, необхідна величина ставки відсотків за вкладами, так, остання зазвичай варіюється за різними кредитним організаціям, територіями, найбільш стабільним орієнтиром служить облікова ставка або ставка рефінансування, за якою, як відомо, Центральний банк країни надає кредити комерційним банкам [8].

Загалом потрібно визначити, що основні існуючі методи визначення ціноутворення вартості лотів аукціону ґрунтуються на роботі експертів, вимагають значного набору даних, а відповідно на звичайному електронному аукціону є або непридатними, або придатними з великими обмеженнями, вимагається простіший механізм, який дозволить продавати не тільки продукти мистецтва, коштовні речі, а дозволить працювати з продажем елементарних товарів, які мають невелику вартість.

2.2 Методи пошуку подібних зображень

Існує достатньо багато різних методів, які можна використати для пошуку подібних зображень. За результатами проведеного аналізу було виділено дві групи методів, які в подальшому будуть використовуватися в роботі.

Перцептивні геші використовуються для знаходження подібних зображень, щодо них можна виділити наступні властивості:

- можна порівнювати між собою і робити висновок про ступінь відмінності двох наборів даних – саме ця властивість і використовується для визначення ступеня схожості зображень: оскільки перцептивні геш-алгоритми описують клас функцій для генерації порівнюваних гешів, то характеристики зображення використовуються для генерації індивідуального (але не унікального) відбитка, і ці відбитки можна порівнювати один з одним, таким чином, завдання порівняння зображень зводиться до обчислення геш-значень цих зображень, і обчислення відстані Геммінга між ними, відповідно чим менша відстань Геммінга, тим більше схожі дані зображення;

- алгоритми обчислення перцептивного геша мають однакові базові властивості, тому зображення можна змінювати в розмірі, змінювати співвідношення сторін і навіть трохи змінювати колірні характеристики (яскравість, контраст), але вони все одно збігаються за гешем;

- для отримання геша є декілька розповсюджених алгоритмів: один з найпростіших перцептивних геш-алгоритмів відображає середнє значення низьких частот, у зображеннях високі частоти забезпечують деталізацію, а низькі частоти показують структуру, велике деталізоване зображення містить багато високих частот, у дуже маленькому зображенні немає деталей, так що воно цілком складається з низьких частот, найпростіший спосіб позбутися від високих частот – зменшити зображення, для порівняння різних гешів підраховується кількість різних бітів в них (відстань Геммінга) [9].

Ще одним інструментом, який може використовуватися для розв'язання проблеми, є нейронні мережі. На даний момент вони дозволяють отримувати достатньо високі результати для даної проблеми. Нейронною мережею часто називають повнозв'язну нейронну мережу: у ній кожен вузол (крім вхідного і вихідного) виступає як входом, так і виходом, утворюючи прихований шар нейронів, і кожен нейрон наступного шару з'єднаний з усіма нейронами попереднього, входи подаються з вагами, які в процесі навчання налаштовуються і не змінюються надалі, при цьому у кожного нейрона є поріг активації, після проходження якого він приймає одне з двох можливих значень (-1 або 1, або 0 або 1) [10].

2.3 Згорткові нейронні мережі

Згорткові нейронні мережі мають спеціальну архітектуру, що дозволяють їм максимально ефективно розпізнавати образи: сама ідея згорткової нейронної мережі ґрунтується на чергуванні згорткових і субдискретизуючих шарів (pooling), а структура є однонаправленою, згорткова нейронна мережа отримала свою назву від операції згортки, яка передбачає, що кожен фрагмент зображення буде помножений на ядро згортки поелементно, при цьому отриманий результат повинен додаватися і записуватися в схожу позицію вихідного зображення, така архітектура забезпечує інваріантність розпізнавання щодо зсуву об'єкта, поступово збільшуючи вікно, на яке приходиться згортка, виявляючи все більш і більш великі структури і патерни в зображенні [10].

Виділимо основні характеристики шарів згорткової нейронної мережі та принципів їх роботи в наступний набір:

- роль згорткової нейронної мережі полягає у зменшенні зображень у форму, яку легше обробити, не втрачаючи особливостей, які є критично важливими для отримання хорошого прогнозу;

– метою операції згортання є вилучення високоякісних характеристик, таких як межі, із вхідного зображення, мережу не потрібно обмежувати лише одним згортковим шаром, так зазвичай, перший шар відповідає за виділення характеристик низького рівня, таких як колір, градієнт тощо, за допомогою доданих шарів архітектура також адаптується до характеристик високого рівня, надаючи мережу, яка має цілісне розуміння зображень у наборі даних;

– ядро представляє собою фільтр або вікно, яке ковзає по всій області попередньої карти і знаходить певні ознаки об'єктів, розмір ядра зазвичай беруть в межах від 3×3 до 7×7 : якщо розмір ядра маленький, то воно не зможе виділити ознаку, якщо занадто великий, то збільшується кількість зв'язків між нейронами, також розмір ядра вибирається таким, щоб розмір карт згорткового шару був парним, це дозволяє не втрачати інформацію при зменшенні розмірності в шарі пулінгу;

– шар пулінгу відповідає за зменшення просторового розміру об'єкта, це має зменшити обчислювальну потужність, необхідну для обробки даних, завдяки зменшенню розмірності, крім того, це корисно для вилучення домінантних ознак, які є інваріантними щодо обертання та позиції, таким чином підтримуючи процес ефективного навчання моделі, існує два типи даного шару: максимальне об'єднання (maxpool) та середнє об'єднання (average pool), при цьому максимальне об'єднання повертає максимальне значення частини зображення, охопленої ядром, відповідно середнє об'єднання повертає середнє значення серед всіх отриманих значень ядра;

– додавання повністю зв'язаного шару – це (як правило) дешевий спосіб вивчення нелінійних комбінацій ознак високого рівня, представлених результатами згорткового шару, повнопов'язаний шар вивчає можливу нелінійну функцію в цьому просторі, тепер, після перетворення вхідного зображення у відповідну форму для багат шарового персептрона, зображення перетворюється у вектор стовпця, згладжений вихід подається в нейронну мережу прямого зв'язку і зворотного розповсюдження помилки, де застосовується до кожної ітерації навчання, протягом ряду епох модель

здатна розрізняти домінуючі та певні низькорівневі ознаки на зображеннях та класифікувати їх, використовуючи техніку класифікації Softmax [11]-[12].

2.4 Нейронна мережа VGG-19

Нейронна мережа VGG-19 (рис. 2.1) складається з наступних шарів:

- згортковий шар Conv3x3 (глибина 64);
- згортковий шар Conv3x3 (глибина 64);
- шар MaxPool;
- згортковий шар Conv3x3 (глибина 128);
- згортковий шар Conv3x3 (глибина 128);
- шар MaxPool;
- згортковий шар Conv3x3 (глибина 256);
- згортковий шар Conv3x3 (глибина 256);
- згортковий шар Conv3x3 (глибина 256);
- згортковий шар Conv3x3 (глибина 256);
- шар MaxPool;
- згортковий шар Conv3x3 (глибина 512);
- згортковий шар Conv3x3 (глибина 512);
- згортковий шар Conv3x3 (глибина 512);
- згортковий шар Conv3x3 (глибина 512);
- шар MaxPool;
- згортковий шар Conv3x3 (глибина 512);
- згортковий шар Conv3x3 (глибина 512);
- згортковий шар Conv3x3 (глибина 512);
- згортковий шар Conv3x3 (глибина 512);
- шар MaxPool;
- повнозв'язний шар Fully Connected (4096);
- повнозв'язний шар Fully Connected (4096);
- повнозв'язний шар Fully Connected (1000);

– шар SoftMax [13]-[14].

Застосування даної моделі для пошуку схожих зображень зокрема виконано в роботах [15]-[17].

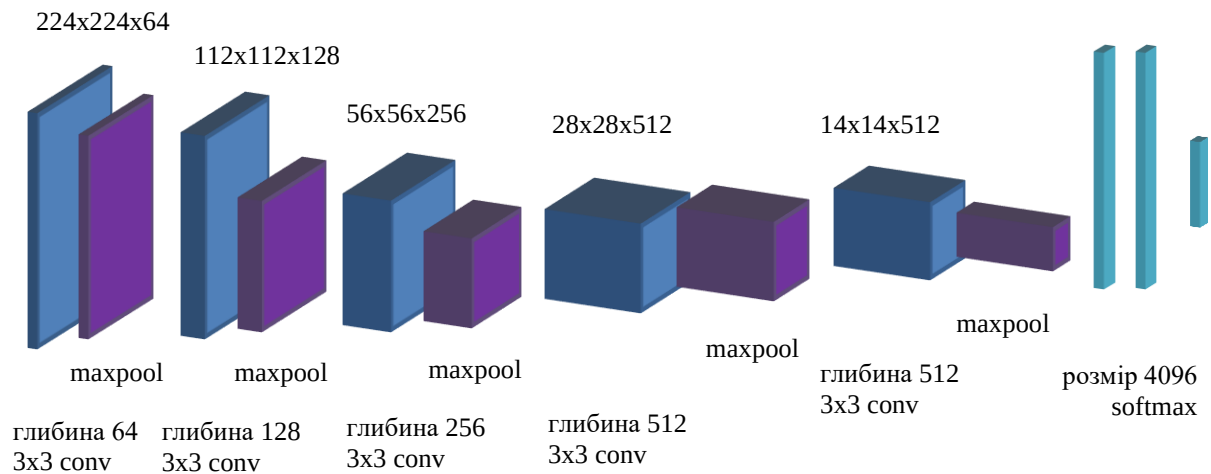


Рисунок 2.1 – Архітектура моделі VGG-19

2.5 Метод підтримки проведення аукціону

Користуючись описаними рішеннями, було сформульовано метод підтримки проведення аукціону. Цей метод використовує стандартну модель проведення аукціону, коли всі користувачі знають про максимальну ціну поточної пропозиції. Тобто це модель відкритого багаторандового аукціону.

На першому етапі роботи методу формується продавцем лот з визначенням параметрів проведення аукціону, що включають дату і час його початку, дату і час його завершення та набір зображень (або окреме зображення), що характеризує лот.

На другому етапі визначається подібність зображення. Для цього використовується модель нейронної мережі на основі архітектури VGG-19. Вона використовується для виділення ознак зображення і формування представлення зображення на основі глибинної нейронної мережі. Нейронна мережа має навчатися для виділення груп лотів, тобто категорій, для яких зокрема ціноутворення може виконуватися по-різному, тобто це можуть бути

вартісні об'єкти мистецтва, звичайні об'єкти мистецтва, спортивні товари тощо. Далі отримане представлення використовується для роботи перцептивного геша, яка за допомогою відстані Гемінга дозволяє отримати оцінку подібності двох зображень між собою.

На третьому етапі визначається початкова і кінцева вартості лоту. Для цього для кожного лота, аукціон якого вже завершено, визначається оцінка подібності на даний лот. Для цього обирається найбільша оцінка подібності серед всіх оцінок, отриманих попарно для всіх зображень цих двох лотів. Далі формується перелік лотів, за якими аукціон завершено, при цьому вони сортуються за отриманим максимальним значенням подібності. Вартість лоту визначається парою: середня початкова і середня фактична вартість продажу лотів, при цьому для усереднення беруться n перших значень.

На наступному етапі продавець узгоджує початкову вартість лоту. Він може встановити її такою, яку було визначено на попередньому етапі, може змінити. В деяких випадках продавець може бажати підвищити активність на аукціоні і встановити нижчу ціну. Тоді отримана вартість на попередньому етапі є маркерним показником, на який він може орієнтуватися при виставленні власної ціни.

На наступному етапі відбувається багатораундовий аукціон, коли протягом часу до завершення аукціону кожен покупець знаходить потрібний йому лот, за бажання встановлює пропозицію щодо купівлі лоту, після чого йому надається набір подібних лотів, які визначаються аналогічно способу, описаному на другому етапі. Тобто лоти, які мають подібні зображення, формують перелік рекомендованих для подальшої участі. Це дозволяє підтримувати процес участі користувачів у всіх аукціонах, розширюючи таким чином аудиторію.

На останньому етапі відбувається підведення підсумків аукціону. Визначається переможець або переможці аукціону.

Якщо лот продавався в кількості 1 одиниці, то він віддається тому, хто запропонував найбільшу суму.

Якщо лот продавався в кількості більше однієї одиниці, то покупцю, яку запропонував найбільшу вартість, віддається потрібна йому кількість товару, а далі вся інша наявна кількість віддається покупцю, який запропонував другу за значенням вартість, і так циклічно продовжується крок за кроком, поки не буде продано всю наявну кількість товару, або поки не залишиться пропозицій.

2.6 Висновки за розділом 2

За результатами виконання розділу 2 було отримано наступні результати:

- проаналізовано методи підтримки аукціонної діяльності, які на даний момент на практиці функціонують в основному за допомогою експертних оцінок;
- проаналізовано існуючі методи пошуку подібних зображень, серед яких розглянуто перцептивні геші та нейронні мережі;
- розглянуто принципи побудови і функціонування згорткових нейронних мереж з детальним визначенням структури нейронної мережі VGG-19;
- запропоновано метод підтримки проведення аукціону.

Наукова новизна роботи полягає в запропонованому методі підтримки проведення аукціону, проведення якого ґрунтується на моделі відкритого багатораундового аукціону, який використовує згорткову нейронну мережу на основі моделі архітектури VGG-19 для виділення подібних зображень лотів, що дозволяє узгоджувати початкову вартість лоту з подібними раніше проданими лотами, отримувати оцінку фактичної вартості лоту продавцям та знаходити подібні лоти для забезпечення синхронності аукціону для покупців.

3 ОПИС ПРОГРАМИ

3.1 Вибір мови програмування

Основними альтернативами для розробки програмного забезпечення підтримки проведення аукціону є мови програмування Java і C# [18], оскільки розроблюваний застосунок є настільним рішенням.

Отримані результати порівняння даних мов програмування для реалізації розроблюваного програмного проєкту в роботі були зведені до табл. 3.1.

Таблиця 3.1 – Вибір мови програмування

Показник	Java	C#
Парадигма	На основі класів, об'єктно-орієнтована мова, похідна від C++	Об'єктно-орієнтована, компонентно-орієнтована, функціональна
Збирач сміття	Є	Є
Область застосування	Вебзастосунки	Веброзробка, розробка ігор, розробка настільних застосунків
Розробка застосунків для Windows	Можлива	Основний напрямок

У підсумку саме мову C# було обрано для реалізації, оскільки вона зручніша для розробки настільних Windows-застосунків, окрім того має можливості компонентно-орієнтованого програмування, що додатково сприяє повторному використанню коду в проєкті.

3.2 Моделювання функціональних вимог до програмного забезпечення

Виконаний огляд програмних аналогів у першому розділі дозволив виділити в якості функціональних вимог до програмного забезпечення наступний набір:

- додавання лотів і внесення їх даних;
- керування лотом;
- пошук лотів;
- перегляд лотів;
- створення пропозиції щодо купівлі лоту;
- перегляд власних пропозицій;
- перегляд власних проданих лотів;
- перегляд власних активних лотів;
- перегляд власних куплених лотів;
- реєстрація користувачів;
- авторизація користувачів;
- видалення лотів;
- автоматичне визначення переможців аукціонів.

З програмою мають працювати покупці, продавці та адміністратори.

На рис. 3.1 змодельовано діаграму прецедентів для роботи покупця.

Покупці та продавці безпосередньо працюють з аукціоном.

Покупець працює з програмою наступним чином:

- пошук лотів;
- перегляд лотів;
- створення пропозиції щодо купівлі лоту;
- перегляд власних пропозицій;
- перегляд власних куплених лотів;
- реєстрація;
- авторизація.



Рисунок 3.1 – Діаграма прецедентів для роботи покупця з програмою

На рис. 3.2 змодельовано діаграму прецедентів для роботи продавця.

Продавець працює з програмою наступним чином:

- додавання лотів і внесення їх даних;
- керування лотом;
- перегляд власних проданих лотів;
- перегляд власних активних лотів;
- авторизація;
- видалення лотів.

Адміністратори працюють над результатами аукціонів.



Рисунок 3.2 – Діаграма прецедентів для роботи продавця з програмою

На рис. 3.3 змодельовано діаграму прецедентів для роботи адміністратора.

Адміністратор працює з програмою наступним чином:

- керування лотом;
- пошук лотів;
- перегляд лотів;
- авторизація;
- видалення лотів;
- перегляд результатів аукціону.

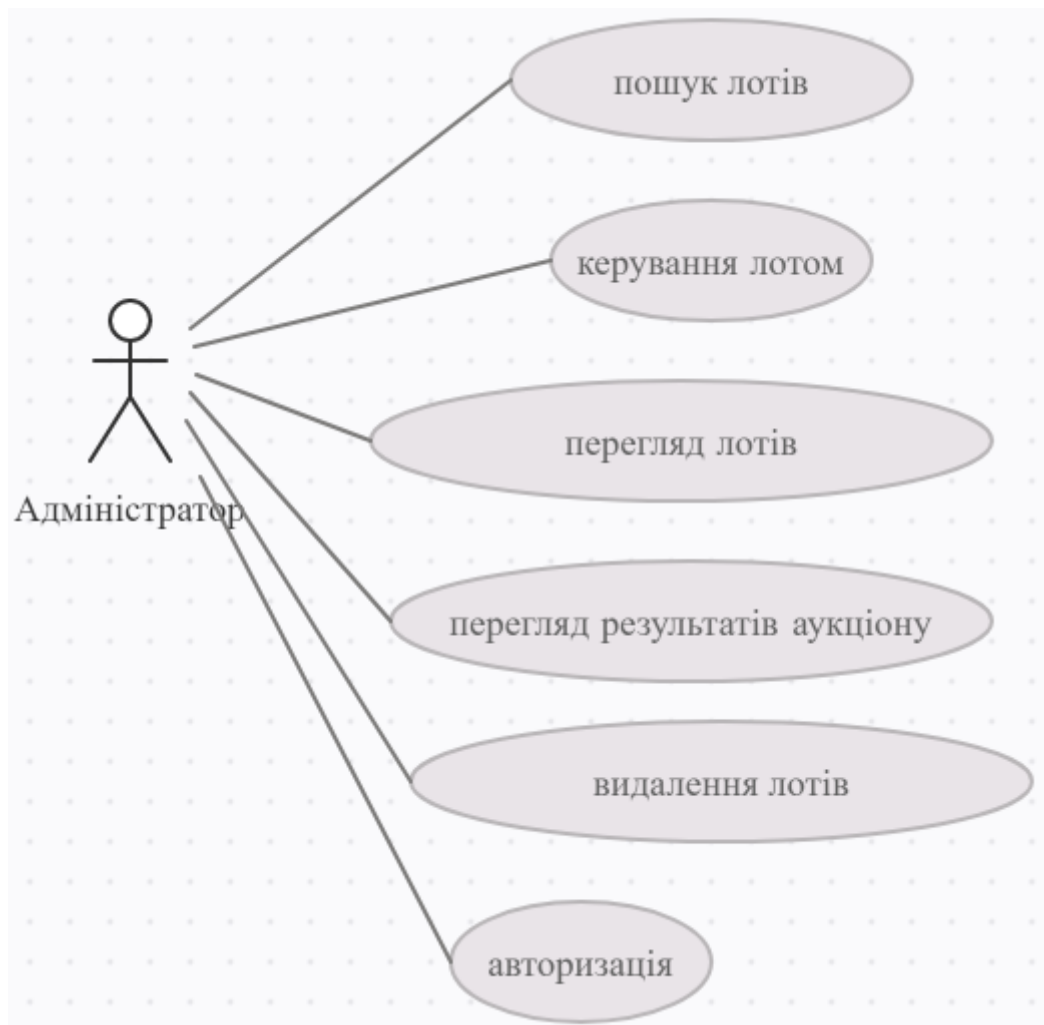


Рисунок 3.3 – Діаграма прецедентів для роботи адміністратора з програмою

3.3 Проєктування структури бази даних

Аналіз предметної області призвів до виділення основних сутностей, що є основою побудови бази даних:

- лот аукціону Lot;
- зображення лоту ImageF;
- рахунок Account;
- аукціонер Auctioneer;
- пропозиція щодо купівлі Proposition.

Лот аукціону визначає безпосередньо товар, який продається на аукціоні. Кожен лот може мати нуль, одну або більше пропозицій щодо його

купівлі, які виставляються покупцями. Тобто таблиці Lot і Proposition мають зв'язок один-до-багатьох.

Кожен лот може мати нуль, одне або більше зображень, при цьому кожне зображення відповідає тільки одному лоту. Тобто таблиці Lot і ImageF мають зв'язок один-до-багатьох.

Таблиця Account використовується для створення базової сутності, необхідної для збереження даних облікового запису користувача, до яких в програмі мають входити як адміністратори, менеджери аукціонів, так і учасники аукціонів, покупці та продавці. Для збереження безпосередньо самих даних і про покупців, і про продавців на аукціоні використовується таблиця Auctioneer. Відповідно кожен аукціонер повинен мати запис в таблиці Account, при чому один такий запис. Кожен користувач з таблиці Account може не мати жодного, а може мати один запис в таблиці Auctioneer. Тобто таблиці Account і Auctioneer мають між собою зв'язок один-до-одного.

Кожен продавець (аукціонер відповідного типу) може виставити на аукціон нуль, один або багато власних лотів. При цьому один лот може бути створена тільки одним аукціонером. Тобто таблиці Auctioneer і Lot мають зв'язок один-до-багатьох.

Кожен покупець (аукціонер відповідного типу) може створити нуль, одну або багато пропозицій на придбання різних лотів. При цьому одна пропозиція може бути створена тільки одним аукціонером. Тобто таблиці Auctioneer і Proposition мають зв'язок один-до-багатьох.

Таблиця Lot дозволяє зберігати в базі даних дані лотів. Поля даної таблиці:

- поле ID – унікальний номер лоту;
- поле Name – ім'я лоту;
- поле Info – опис лоту;
- поле Price – ціна лоту;
- поле Count – кількість одиниць;

- поле Author – автор лоту (продавець);
- поле Type – поточний стан лоту;
- поле BeginDate – дата створення лоту або початку аукціону, якщо вони відрізняються;

- поле EndDate – дата завершення аукціону.

Поле Author є зовнішнім ключем для зв'язку таблиць Lot і Auctioneer.

Таблиця Account дозволяє зберігати в базі даних дані акаунтів. Поля даної таблиці:

- поле ID – унікальний номер користувача;
- поле Login – ім'я користувача;
- поле Password – пароль користувача;
- поле Type – тип користувача.

Таблиця Auctioneer дозволяє зберігати в базі даних дані аукціонерів.

Поля даної таблиці:

- поле ID – унікальний номер аукціонера;
- поле Auctioneer – унікальний номер користувача;
- поле Surname – прізвище аукціонера;
- поле Name – ім'я аукціонера;
- поле FatherName – по батькові аукціонера;
- поле Type – тип аукціонера;
- поле Mail – пошта аукціонера;
- поле Phone – телефон аукціонера;
- поле Address – адреса аукціонера;
- поле Info – опис аукціонера.

Поле Auctioneer є зовнішнім ключем для зв'язку таблиць Auctioneer і Account.

На рис. 3.4 зображена структура бази даних. В якості системи керування базами даних було обрано Microsoft SQL Server.

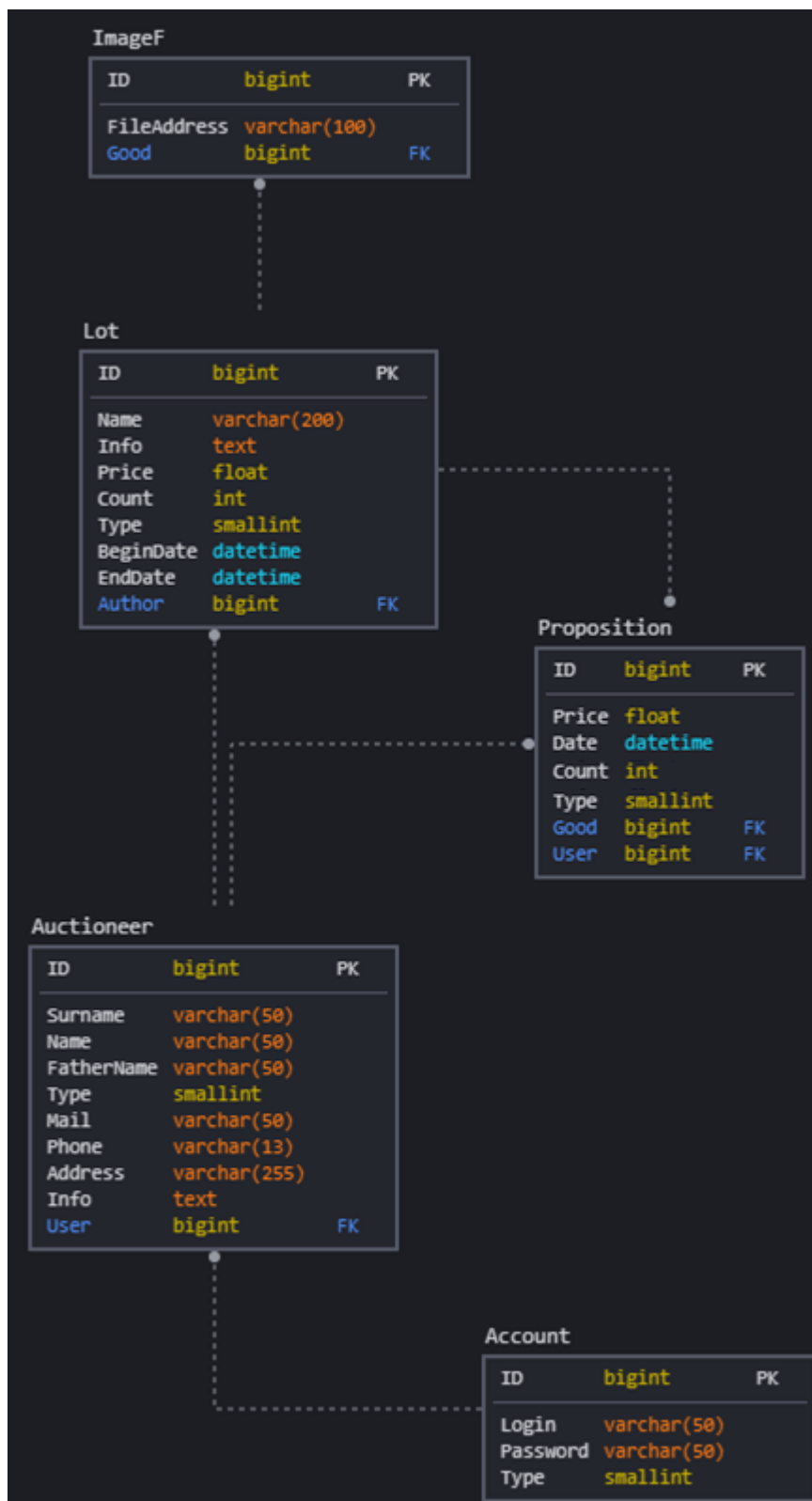


Рисунок 3.4 – Структура бази даних

Таблиця Proposition дозволяє зберігати в базі даних дані аукціонерів.

Поля даної таблиці:

- поле ID – унікальний номер пропозиції;

- поле Auctioneer – номер покупця;
- поле Good – номер лоту;
- поле Price – вартість пропозиції;
- поле Date – дата пропозиції;
- поле Count – бажана кількість;
- поле Type – тип (стан) пропозиції.

Поле Auctioneer є зовнішнім ключем для зв'язку таблиць Proposition і Account.

Поле Good є зовнішнім ключем для зв'язку таблиць Proposition і Lot.

Таблиця ImageF дозволяє зберігати в базі даних дані зображень. Поля даної таблиці:

- поле ID – унікальний номер зображення;
- поле FileAddress – адреса файлу;
- поле Good – номер лоту.

Поле Good є зовнішнім ключем для зв'язку таблиць ImageF і Lot.

3.4 Проєктування структури програмного застосунку

На рис. 3.5 зображена структурна схема програмного застосунку.

Програмний застосунок AuctionSupporter складається з застосунку з графічним інтерфейсом користувача ASApp та програмної бібліотеки ASLibrary.

Застосунок ASApp включає до своєї структури основну вхідну точку всіх застосунків з графічним інтерфейсом мовою C# – файлу Program.cs. Він у свою чергу викликає під час роботи форму авторизації.

За необхідності форма авторизації звертається до форми реєстрації. Окрім того вона безпосередньо взаємодіє з формами покупця, продавця та адміністратора.



Рисунок 3.5 – Структурна схема програмного застосунку

Програмна бібліотека ASLibrary представлена динамічно приєднуваною бібліотекою у вигляді файлу ASLibrary.dll. Всередині цієї бібліотеки є файл ASMethod.cs, який реалізує роботу запропонованого методу, файл Interface.cs, який визначає програмний інтерфейс доступу до файлу AS.cs, що реалізує безпосередньо роботу в межах предметної області,

що реалізується. Файл DatabaseProcessor.cs всередині бібліотеки призначений для роботи з базою даних (виконання запитів до неї).

3.5 Висновки за розділом 3

За результатами виконання розділу 3 було отримано наступні результати:

- виконано порівняння мов програмування C# і Java для реалізації програмного забезпечення;
- мову C# обрано для реалізації;
- виділено функціональні вимоги до розроблюваного програмного забезпечення;
- побудовано діаграми прецедентів для роботи покупця, продавця та адміністратора з програмою;
- забезпечено проектування бази даних з приведенням структурної схеми бази даних;
- систему керування базами даних Microsoft SQL Server обрано для реалізації;
- забезпечено проектування структури програмного застосунку з приведенням структурної схеми.

4 РОЗРОБКА ПРОГРАМИ

4.1 Алгоритм функціонування програми

Центральним елементом роботи програми є лот, оскільки саме він визначає те, які саме дії ведуться і на якому саме етапі знаходиться аукціон. Продавець створює лоти, керує ними, а покупці створюють власні пропозиції, впливаючи на результат аукціону. Саме тому для опису алгоритму функціонування програми вирішено було використати саме діаграму станів лоту аукціону.

Робота програми над лотом розпочинається з того, що продавець його створює. Одразу після цього може розпочатися аукціон, і лот буде в стані активному. Аукціон може розпочатися також пізніше, тоді через деякий час лот стане активним.

Далі під час проведення аукціону, тобто поки час завершення роботи з лотом не перевищує поточний, продавець може зняти його з аукціону, після чого лот перейде в стан вилучений.

Якщо продавець не зняв лот з аукціону, то коли поточний момент часу буде дорівнювати часу завершення аукціону лота, лот буде переведено в стану придбаний.

Далі покупець може або сплатити за лот, а продавець отримати таку суму, після чого лот перейде в завершений стан, або покупець може не сплатити обіцяну суму, тоді лот перейде в стан вилученого, а далі продавець може його перевести знову в стан активного, або завершити по ньому роботу, перевівши його до завершеного стану. В завершеному стану лот продовжує зберігатися в базі даних, але ніякі роботи над ним не ведуться.

Результуючу діаграму станів, створену для лоту аукціону, зображено на рис. 4.1.

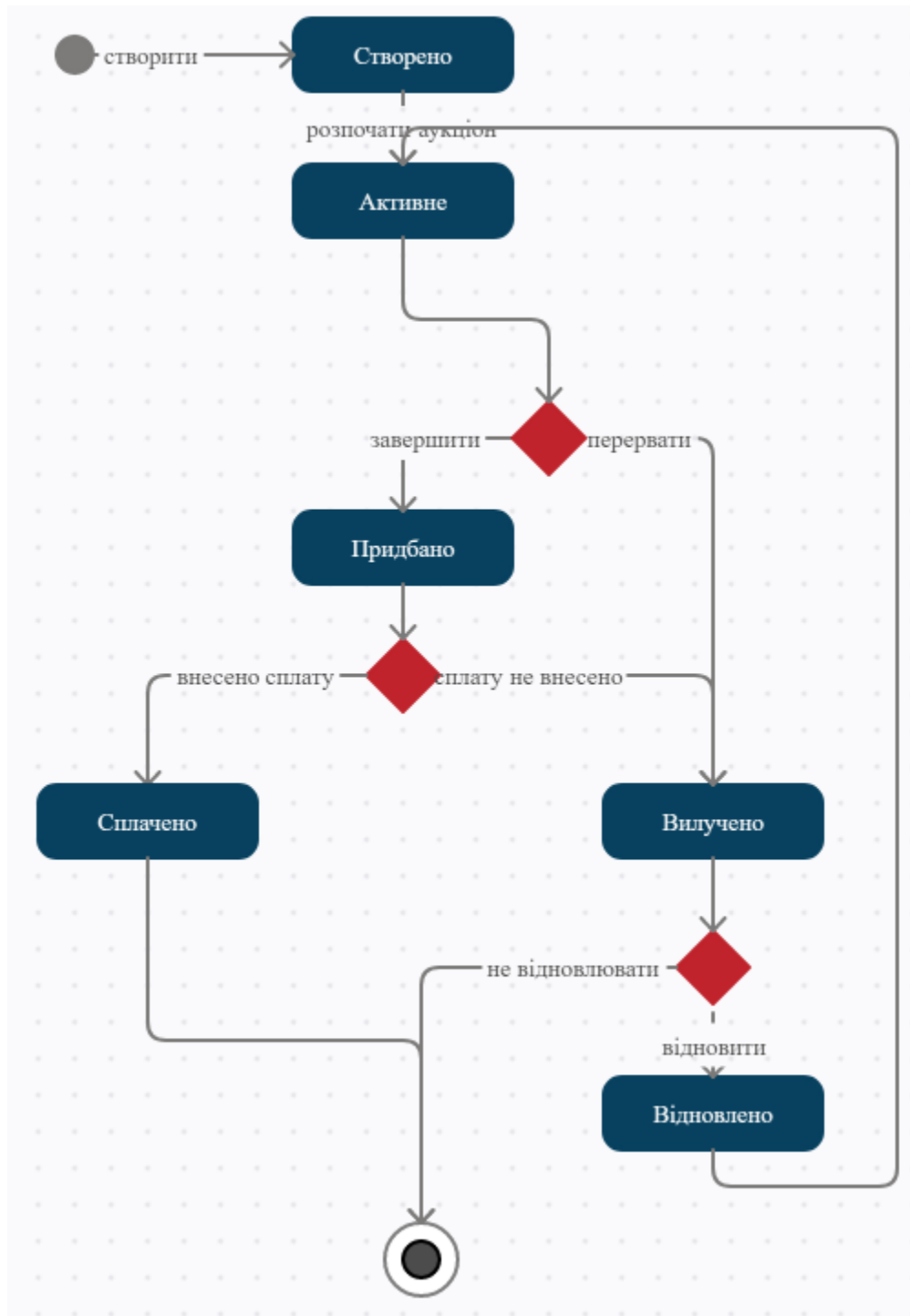


Рисунок 4.1 – Діаграма станів лоту аукціону

4.2 Опис розроблених модулів

Клас Lot дозволяє зберігати та виконувати необхідні дії над лотами аукціону.

Клас Lot містить такі поля:

- int id – номер лоту;
- string name – назва лоту;
- List<string> info – опис лоту;
- double price – вартість лоту;
- int count – кількість одиниць товару в лоті;
- Account author – автор лоту;
- int type – тип лоту;
- DateTime begindate – дата початку аукціону лоту;
- DateTime enddate – дата завершення аукціону лоту;
- List<String> images – перелік шляхів до файлів з зображеннями цього лоту;
- DatabaseProcessor dproc – з'єднання з базою даних.

Клас містить конструктор Lot(), який призначений для створення нового об'єкту і визначає номер лоту шляхом збільшення найбільшого номеру лоту в базі даних, конструктор Lot(int q) дозволяє встановити в якості номеру лоту значення -1. Для кожного поля класу призначено get-тери та set-тери.

Параметр type може приймати одне з наступних значень:

- значення «0» відповідає неактивному лоту (тільки створеному лоту);
- значення «1» відповідає активному лоту (який знаходиться на аукціоні зараз);
- значення «2» відповідає лоту, за яким вже підведено результати аукціону;
- значення «3» відповідає лоту, який вже фактично продано, тобто внесено відповідну сплату і проконтрольовано адміністратором;
- значення «4» відповідає лоту, який вилучено.

Клас Account дозволяє зберігати та виконувати необхідні дії над акаунтами.

Клас Account містить такі поля:

- id – унікальний номер користувача;

- string login – ім'я користувача;
- string password – пароль користувача;
- int type – тип користувача.

Для кожного поля класу призначено get-тери та set-тери.

Клас Auctioneer дозволяє зберігати та виконувати необхідні дії над записами учасників аукціону (покупцями та продавцями). Даний клас створюється шляхом наслідування класу Account, тобто будь-який учасник аукціону повинен мати акаунт в системі.

Клас Auctioneer містить такі поля:

- int ida – унікальний номер аукціонера;
- string surname – прізвище аукціонера;
- string name – ім'я аукціонера;
- string fathurname – по батькові аукціонера;
- int typea – тип аукціонера;
- string mail – пошта аукціонера;
- string phone – телефон аукціонера;
- string address – адреса аукціонера;
- List<string> info – опис аукціонера.

Для кожного поля класу призначено get-тери та set-тери.

Клас Salesman дозволяє зберігати та виконувати необхідні дії зі сторони продавця на аукціоні. Даний клас створюється шляхом наслідування класу Auctioneer, який містить необхідні основні характеристики даної сутності.

Клас Salesman містить додатково такі поля:

- DatabaseProcessor dproc – з'єднання з базою даних;
- List<Lot> activeLots – перелік активних лотів продавця;
- List<Lot> doneLots – перелік проданих лотів продавця.

Окрім конструкторів, організованих подібно описаним вище, клас також містить наступні методи.

Метод `void removeLot(int q)` виконує вилучення з переліку активних лотів лоту з номером `q`, тобто забезпечує зняття лоту з аукціону.

Метод `void ChangeLotType(int lot, int type)` реалізує зміну типу лоту з номером `lot` на тип `type`, вносячи відповідні зміни в базу даних, при цьому досліджується і перелік активних лотів, і перелік проданих лотів.

Метод `void buyLot(Lot good)` забезпечує фіксацію продажу лоту `good` зі внесенням відповідних даних в базу даних та паралельною фіксацією в базі даних зміни кількості товару.

Метод `void Save()` ініціює збереження в базу даних всіх даних, що містяться в даному об'єкті в поточний момент, включаючи всі дані акаунту, дані лотів продавця.

Метод `void getSalesman(int q)` виконує заповнення полів продавця даними з бази даних за його номером `q`, при цьому перелік його активних та проданих лотів заповнюються також.

Метод `void addLot(Lot good)` виконує додавання до переліку активних лотів продавця `activeLots` лоту `good`.

Клас `Buyer` дозволяє зберігати та виконувати необхідні дії зі сторони покупця на аукціоні. Даний клас створюється шляхом наслідування класу `Auctioneer`, який містить необхідні основні характеристики даної сутності.

Клас `Buyer` містить додатково такі поля:

- `DatabaseProcessor dproc` – з'єднання з базою даних;
- `List<Lot> activeLots` – перелік активних лотів;
- `List<Lot> doneLots` – перелік придбаних лотів;
- `List<Lot> activePropositions` – перелік активних пропозицій;
- `List<Lot> donePropositions` – перелік завершених пропозицій;
- `List<string> notifications` – перелік повідомлень.

Окрім конструкторів, організованих подібно описаним вище, клас також містить наступні методи.

Метод `void buy(Lot good)` фіксує виконання купівлі користувачем лоту аукціону `good`, вносячи зміни зокрема і в стану відповідної пропозиції.

Метод `void Save()` виконує збереження даних про покупця разом з переліком його пропозицій в базу даних.

Метод `void getBuyer(int q)` виконує заповнення полів покупця даними з бази даних за його номером `q`, при цьому переліки його активних і проданих лотів і пропозицій заповнюються також.

Метод `List<Lot> SearchLots(String name)` забезпечує виконання пошуку лотів, в назві яких міститься `name`, в результаті повертається перелік відповідних лотів.

Метод `void addProposition(Lot good, double price)` реалізує додавання пропозиції про купівлю товару `good` за ціною `price` активним покупцем, в якості дати операції використовується поточна.

Клас `Proposition` дозволяє зберігати та виконувати необхідні дії над пропозиціями щодо купівлі лотів аукціону.

Клас `Proposition` містить такі поля:

- `int id` – номер лоту;
- `float price` – вартість купівлі лоту в пропозиції;
- `datetime date` – дата і час реєстрації пропозиції;
- `int type` – стан пропозиції;
- `Lot good` – лот, щодо якого створено пропозицію;
- `Buyer user` – користувач, який бажає виконати купівлю;
- `int count` – бажана кількість;
- `DatabaseProcessor dproc` – з'єднання з базою даних.

Клас містить конструктор `Proposition ()`, який призначений для створення нового об'єкту і визначає номер лоту шляхом збільшення найбільшого номеру лоту в базі даних, конструктор `Proposition (int q)` дозволяє встановити в якості номеру лоту значення `-1`. Для кожного поля класу призначено `get`-тери та `set`-тери.

Клас `Auction` призначений для виконання інших дій над об'єктами класів і взаємодії з базою даних. Містить наступні методи.

Метод `int getAuctioneerType(string login, string password)` визначає тип облікового запису користувача з логіном `login` та паролем `password`, повертаючи тип запису в результаті своєї роботи. Необхідний для визначення, який саме інтерфейс потрібно відкривати після авторизації.

Метод `bool isAuctioneer(string login, string password)` виконує перевірку того, чи аукціонер з логіном `login` та паролем `password` акаунту існує в базі даних, тобто чи можна виконати аутентифікацію акаунту.

Метод `bool Register(string login, string password)` забезпечує реєстрацію акаунту з логіном `login` та паролем `password`.

Метод `List<Lot> getAllLots()` повертає перелік всіх лотів.

Метод `Lot getLotById(int q)` повертає повні дані лоту за його номером `q`.

Метод `List<int> getAuctionResults()` забезпечує підведення підсумків аукціону для всіх лотів, повертаючи в результаті перелік тих, за якими було визначено перелік переможців.

Клас `DatabaseProcessor` виконує запити до бази даних і зокрема надає наступні методи:

- `detectMaxLot()` – метод, який повертає максимальний номер лоту, що існує в базі даних;
- `detectMaxAccount()` – метод, який повертає максимальний номер акаунту, що існує в базі даних;
- `detectMaxAuctionerr()` – метод, який повертає максимальний номер аукціонера, що існує в базі даних;
- `changeLotTypeToDel(int q)` – метод, який змінює стан лоту для лоту номер `q` на вилучений;
- `changeLotType(int lot, int type)` – метод, який змінює стан лоту з номером `lot` на тип `type`;
- `changeLotTypeToBought(int q)` – метод, який змінює стан лоту `q` на придбаний;

– AddSalesman(int ida, string surname, string name, string fathename, int typea, string mail, string phone, string address, List<string> info) – метод, який виконує додавання запису про продавця з номером ida, прізвищем surname, іменем name, по батькові fathename, типом typea, адресою пошти mail, телефоном phone, адресою address та описом info;

– getSalesmanSurname(int q) – метод, який повертає прізвище продавця з номером q;

– getSalesmanName(int q) – метод, який повертає ім'я продавця з номером q;

– getSalesmanFatherName(int q) – метод, який повертає по батькові продавця з номером q;

– getSalesmanMail(int q) – метод, який повертає пошту продавця з номером q;

– getSalesmanPhone(int q) – метод, який повертає телефон продавця з номером q;

– getSalesmanAddress(int q) – метод, який повертає адресу продавця з номером q;

– getSalesmanInfo(int q) – метод, який повертає опис продавця з номером q;

– getSalesmanLotIds(int q) – метод, який повертає лоти продавця з номером q;

– getLotName(int l) – метод, який повертає назву лоту з номером l;

– getLotInfo(int l) – метод, який повертає назву лоту з номером l;

– getLotPrice(int l) – метод, який повертає назву лоту з номером l;

– getLotCount(int l) – метод, який повертає назву лоту з номером l;

– getLotAuthor(int l) – метод, який повертає назву лоту з номером l;

– getLotType(int l) – метод, який повертає назву лоту з номером l;

– getLotBeginDate(int l) – метод, який повертає назву лоту з номером l;

– getLotEndDate(int l) – метод, який повертає назву лоту з номером l;

– getLotImages(int l) – метод, який повертає назву лоту з номером l;

- `getBuyerSurname(int q)` – метод, який повертає прізвище покупця з номером `q`;
- `getBuyerName(int q)` – метод, який повертає ім'я покупця з номером `q`;
- `getBuyerFatherName(int q)` – метод, який повертає по батькові покупця з номером `q`;
- `getBuyerMail(int q)` – метод, який повертає пошту покупця з номером `q`;
- `getBuyerPhone(int q)` – метод, який повертає телефон покупця з номером `q`;
- `getBuyerAddress(int q)` – метод, який повертає адресу покупця з номером `q`;
- `getBuyerInfo(int q)` – метод, який повертає опис покупця з номером `q`;
- `getBuyerPropositionIds(int q)` – метод, який повертає номери пропозицій покупця з номером `q`;
- `getPropositionPrice(int l)` – метод, який повертає вартість пропозиції з номером `l`;
- `getPropositionDate(int l)` – метод, який повертає дату і час пропозиції з номером `l`;
- `getPropositionType(int l)` – метод, який повертає стан пропозиції з номером `l`;
- `getPropositionAuthor(int l)` – метод, який повертає номер акаунта автора пропозиції з номером `l`;
- `getLotIds()` – метод, який повертає перелік унікальних номерів лотів;
- `getMaxPropositionId()` – метод, який повертає максимальний номер, який використано для пропозиції в базі даних;
- `getAccountType(string login, string password)` – метод, який повертає тип облікового запису з логіном `login` та паролем `password`;
- `getAccountId(string login, string password)` – метод, який повертає номер облікового запису з логіном `login` та паролем `password`.

4.3 Висновки за розділом 4

За результатами виконання розділу 4 було отримано наступні результати:

- описано алгоритм функціонування програмного забезпечення з приведенням діаграми станів для лоту аукціону;
- реалізовано модулі та класи програмного забезпечення;
- описано основні розроблені класи з приведенням відповідних методів та описом їх роботи.

5 ЕКСПЛУАТАЦІЯ ТА ТЕСТУВАННЯ ПРОГРАМИ

5.1 Призначення програми

Програма призначена для підтримки проведення аукціону на різних етапах, що зокрема включають внесення даних лотів, забезпечення конкурсу під час реалізації аукціону, визначення переможців аукціону з можливістю декількох переможців одночасно.

5.2 Умови виконання програми

Для виконання програми на сервері потрібно забезпечити наступні апаратні засоби:

- чотирьохядерний процесор з частотою не меншою за 3 ГГц;
- обсяг оперативного запам'ятовуючого пристрою повинен бути не менше за 4 Гб;
- обсяг вільного простору на жорсткому диску – не менше 100 Гб.

Програмний застосунок є Windows-сумісним, тобто для його експлуатації має виконуватися умова підтримки операційної системи Windows. Виходячи з того, що актуальна підтримка зараз здійснюється для версії 7, то саме вона є мінімальною вимогою експлуатації програми.

Для роботи програмного забезпечення має бути обов'язково встановлено в системі фреймворк .NET версії не нижче ніж 4.7.

5.3 Виконання програми

Після відкриття програми на екран буде виведено форму авторизації акаунту (рис. 5.1).

Форма містить два поля для введення логіну та паролю акаунту, відповідно підписані. Після введення потрібно натиснути на кнопку «Вхід».

Рисунок 5.1 – Форма авторизації акаунту

Якщо користувач не має акаунту, то потрібно натиснути на кнопку «Реєстрація», після чого відкриється форма реєстрації акаунту.

Якщо користувач має акаунт продавця, то відкривається після авторизації форма продавця (рис. 5.2). Вона складається з 5 базових областей:

- область «Додавання лоту», що дозволяє виставити на аукціоні новий лот і визначити його параметри;
- область «Керування лотом», що дозволяє змінити стан лоту за рахунок зняття його з аукціону або скасування його продажу у випадку, якщо не вдалося отримати сплату: керування відбувається за рахунок вибору лоту з таблиці, після чого автоматично заповнюється поле «Лот номер» або за рахунок ручного введення номеру лоту, після чого необхідно натиснути на кнопку «Вилучити»;
- область «Обраний лот» дозволяє переглядати детальні параметри обраного з таблиці лоту;
- таблиця «Активні лоти» перелічує номер, назву, кількість товарів та поточну вартість всіх активних лотів авторизованого продавця;
- таблиця «Продані лоти» перелічує номер, назву, кількість товарів та фактичну вартість всіх проданих лотів авторизованого продавця.

Аукціон: Ви увійшли як продавець

Додавання лоту

Назва лоту

Кількість 0

Початок

Завершення

Вартість

Очікувана

Пропонована

Встановити

Опис

Обрати

Вилучити

Активувати

Керування лотом

Лот номер

Вилучити

Обраний лот

Лот номер

До завершення

Вартість

Опис

Обрати

Вилучити

Зберегти

Активні лоти

ID	Назва	Кількість	Вартість
»*			

Продані лоти

ID	Назва	Кількість	Вартість
*			

Рисунок 5.2 – Форма продавця

Область «Додавання лоту» (рис. 5.3) дозволяє продавцю визначити параметри лоту і виставити його на аукціон.

Для цього потрібно вказати назву лоту, ввести кількість екземплярів, дату і час початку і завершення аукціону (за замовчуванням початком є поточна дата і час, що заповнюються, як тільки буде введено назву лоту, обидва поля є маскованими, дозволяють ввести дані тільки в визначеному форматі), опис лоту. Окремо визначаються зображення лоту. Для цього потрібно натиснути на кнопку «Обрати» в цій області під зображенням. Після цього відкриється вікно додавання файлу з зображенням. Як тільки файл буде збережено, він відобразиться на формі. За допомогою кнопки «Вилучити» можна вилучити його. За допомогою кнопок вліво та вправо можна переміщуватися між зображеннями, якщо їх декілька.

Аукціон: Ви увійшли як продавець

Додавання лоту

Назва лоту
Чашка ручної роботи

Кількість

Початок
20.11.2020 10:00

Завершення
30.11.2020 10:00



Опис
Керамічна чашка повністю ручної роботи з зображенням папути на гілці. Створено з душою для теплих напоїв і довгих розмов.

Вартість

Очікувана

Пропонована

Встановити

Обрати

Вилучити

Активувати

Керування лотом

Лот номер

Вилучити

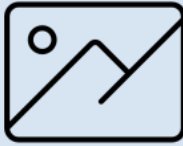
Обраний лот

Лот номер

До завершення

Вартість

Опис



Обрати

Вилучити

Зберегти

Активні лоти

ID	Назва	Кількість	Вартість
20201119141501	Чашка ручної роботи	1	300
20201119203047	Чашка ручної роботи	1	200
20201118201032	Кухоль ручної роботи	1	700
20201118200702	Набір тарілок ручної роботи	1	2800

Продані лоти

ID	Назва	Кількість	Вартість
20201118...	Глечик на добрі справи	1	2700
*			

Рисунок 5.3 – Додавання нового лоту

Після завантаження зображення відбувається його аналіз за допомогою запропонованого методу і якщо було встановлено подібні зображення, то визначається очікувана вартість лоту на основі фактичних вартостей проданих подібних лотів. Розраховане значення виводиться в області «Вартість» в полі «Очікувана». В полі «Пропонована» виводиться вартість отримана на основі початкових вартостей подібних лотів. У поле «Встановити» користувач має ввести значення вартості, яке буде фактично використовуватися як початкова вартість лоту.

Після того, як всі дані лоту задані, потрібно натиснути на кнопку «Активувати» для збереження лоту.

Якщо з таблиці активних або проданих лотів обрати один з лотів, то в правій частині вікна в області «Керування лотом» відобразиться номер

обраного лоту та активною стане кнопка «Вилучити» (рис. 5.4). Номер лоту можна змінити або ввести вручну.

Додавання лоту

Назва лоту

Кількість 0

Початок

Завершення

Вартість

Очікувана

Пропонована

Встановити

Опис

Обрати

Вилучити

Активувати

Керування лотом

Лот номер 20201118200702

Вилучити

Обраний лот

Лот номер 20201118200702

До завершення 11.00.0000 02:16

Вартість 2000

Опис

Обрати

Вилучити

Зберегти

Активні лоти

ID	Назва	Кількість	Вартість
20201119141501	Чашка ручної роботи	1	300
20201119203047	Чашка ручної роботи	1	200
20201118201032	Кухоль ручної роботи	1	700
20201118200702	Набір тарілок ручної роботи	1	2800

Продані лоти

ID	Назва	Кількість	Вартість
20201118...	Глечик на добрі справи	1	2700
*			

Рисунок 5.4 – Керування лотом

Детальна інформація про обраний лот відобразиться в області «Обраний лот». Для нього буде відображено повний текст опису, зображення, час до завершення (в даному випадку 11 діб 2 години 16 хвилин), початкову вартість лоту.

Можна внести зміни в опис лоту або в набір зображень. Після цього для фіксації змін потрібно натиснути на кнопку «Зберегти». Також безпосередньо в таблиці можна змінити назву лоту, всі інші зміни з таблиці не зберігаються.

Якщо виконано авторизацію акаунту покупця, то відкриється форма покупця (рис. 5.5).

Активні лоти

Назва	Кількість	Вартість
Ключниця-пальто з натуральної шкіри	1	270
Авторська картина "Регата"	1	3000
Авторська картина "Хвиля"	1	2500
Ключниця-будиночок ручної роботи	1	190
Сяюча парасоля	5	270
Чашка ручної роботи подвійна	1	350
Гудзикова статуетка	1	650
Набір скетчів з міст світу	1	1300
Пісовий кухоль у вигляді лицаря	1	900
Дерев'яний конструктор "Бентлі 1972"	1	4900
Авторська картина "Піс біля моря"	1	2100

Назва

Шукати

Лот номер

Пропозиція

Вартість

Кількість

Запропонувати

Подібні

Обраний лот

Лот номер

До завершення

Початкова вартість

Опис

Власні пропозиції

Назва	Кількість	Вартість
Сяюча парасоля	2	180
Пивний кухоль з крихтами піска з Мадейри	1	900
Пивний кухоль з крихтами піска з Мадейри	1	670
Глечик на добрі справи	1	2700

Куплені лоти

Назва	Кількість	Вартість
Пивний кухоль з крихтами піска з Мадейри	1	900
Глечик на добрі справи	1	2700

Рисунок 5.5 – Форма покупця

Форма покупця складається з 6 базових областей:

- таблиця «Активні лоти» містить набір активних лотів (усі, результати пошуку або подібні лоти);
- поле «Назва» дозволяє ввести назву або частину назви лоту, після чого потрібно натиснути на кнопку «Шукати» і в таблиці «Активні лоти» відобразяться лоти, назви яких містять вказаний текст;
- область «Пропозиція» дозволяє користувачу ввести його вартість та потрібну кількість екземплярів, після чого потрібно натиснути на кнопку «Запропонувати» для збереження пропозиції;

- область «Обраний лот» дозволяє переглядати детальні параметри обраного з таблиці лоту (рис. 5.6);
- таблиця «Власні пропозиції» перелічує назву, кількість товарів та зазначену вартість покупцем у пропозиції для купівлі лоту авторизованим користувачем;
- таблиця «Куплені лоти» перелічує назву, кількість товарів та фактичну вартість всіх куплених лотів авторизованого продавця.

Аукціон: Ви увійшли як покупець

Активні лоти

Назва	Кількість	Вартість
Ключниця-пальто з натуральної шкіри	1	270
Авторська картина "Регата"	1	3000
Авторська картина "Хвиля"	1	2500
Ключниця-будиночок ручної роботи	1	190
Сяюча парасоля	5	270
Чашка ручної роботи подвійна	1	350
Гудзикова статуетка	1	650
Набір скетчів з міст світу	1	1300
Пісовий кухоль у вигляді лицаря	1	900
Дерев'яний конструктор "Бентлі 1972"	1	4900
Авторська картина "Піс біля моря"	1	2100

Обраний лот

Лот номер: 20201101221138

До завершення: 02.00.0000 10:32

Початкова вартість: 300

Опис

Мініатюрна статуетка в стилі нецке, яку можна використовувати в якості гудзика.



Власні пропозиції

Назва	Кількість	Вартість
Сяюча парасоля	2	180
Півний кухоль з крихтами піска з Мадейри	1	900
Півний кухоль з крихтами піска з Мадейри	1	670
Глечик на добрі справи	1	2700

Куплені лоти

Назва	Кількість	Вартість
Півний кухоль з крихтами піска з Мадейри	1	900
Глечик на добрі справи	1	2700

Назва:

Лот номер: 20201101221138

Пропозиція

Вартість:

Кількість:

Шукати

Запропонувати

Подібні

Рисунок 5.6 – Перегляд даних лоту

Кнопка «Подібні» дозволяє відобразити в таблиці «Активні лоти» ті лоти, зображення яких подібні на зображення обраного в даний момент лоту з таблиці «Активні лоти» (рис. 5.7). Таким чином покупець може переглядати подібні лоти і вибирати, які він хоче придбати.

Аукціон: Ви увійшли як покупець

Активні лоти

	Назва	Кількість	Вартість
	Гудзикова статуетка	1	650
▶	Статуетка ескімоски	1	700
	Мініатюрний гудзик для прикраси одягу	1	640
*			

Назва

Шукати

Лот номер

20201025211740

Пропозиція

Вартість

720

Кількість

1

Запропонувати

Подібні

Обраний лот

Лот номер

20201025211740

До завершення

01.00.0000 19:00

Початкова вартість

350

Опис

Статуетка ескімоски, яка може використовуватися в якості елемента декору.



<

>

Власні пропозиції

	Назва	Кількість	Вартість
	Сяюча парасоля	2	180
	Півний кухоль з крихтами піска з Мадейри	1	900
	Півний кухоль з крихтами піска з Мадейри	1	670
	Глечик на добрі справи	1	2700
*			

Куплені лоти

	Назва	Кількість	Вартість
	Півний кухоль з крихтами піска з Мадейри	1	900
	Глечик на добрі справи	1	2700
*			

Рисунок 5.7 – Перегляд подібних лотів

6 ЕКОНОМІКО-ОРГАНІЗАЦІЙНА ЧАСТИНА

Програмне забезпечення призначено для виконання підтримки проведення аукціону. У процесі підтримки визначається початкова вартість лоту на аукціоні. На даний момент процес було організовано таким чином, що підтримкою діяльності займався аналітик. У результаті впровадження програмного продукту менеджери аукціону зможуть виконувати власну діяльність без звернень до аналітика, тобто аналітика можна буде скоротити.

6.1 Планування розробки програмного продукту

На рис. 6.1 представлений укрупнений алгоритм розробки нової програми [19].

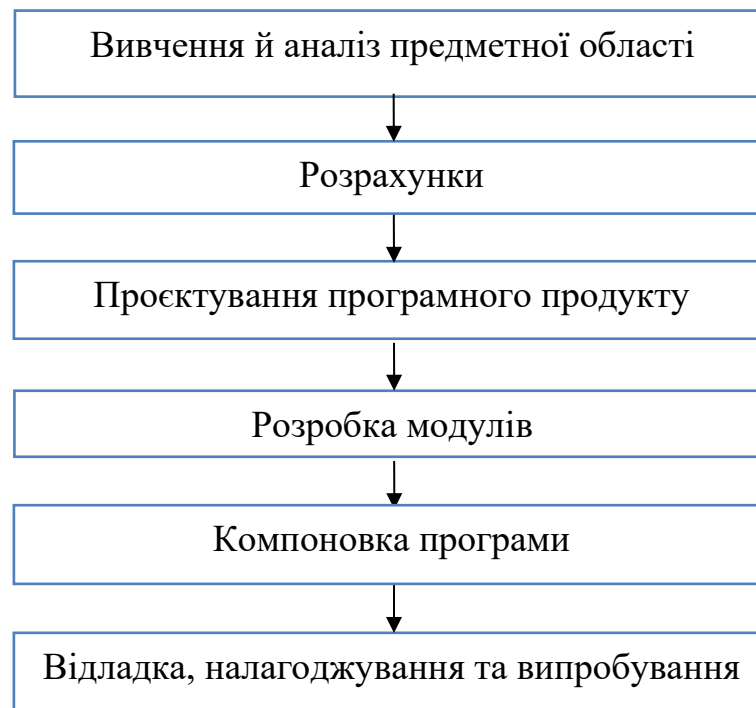


Рисунок 6.1 – Укрупнений алгоритм програми

При плануванні розробки програмного продукту необхідно врахувати витрати робочого часу на виконання основних етапів роботи, що включають

планування й аналіз вимог (предметної області), проектування й іспит програмного продукту [19].

Результати планування заносяться до табл. 6.1.

Таблиця 6.1 – Характеристика виконуваних робіт з розробки програмного продукту

Найменування робіт	Трудомісткість		Виконавці		Тривалість розробки, днів
	люд.-днів	% до підсумку	спеціальність	кількість осіб	
1 Вивчення й аналіз предметної області	10	16,7	Програміст, консультант	2	5
2 Розрахунки	4	6,6	Програміст, консультант	2	2
3 Розробка програми					
3.1 Проектування програмного продукту	20	33,3	Програміст	1	20
3.2 Розробка модулів	10	16,7	Програміст	1	10
3.3 Компоновка програми	10	16,7	Програміст	1	10
4 Тестування і налагодження програмного продукту	6	10	Програміст, консультант	2	3
Разом	60	100			50 10

Найменування робіт	Календарний період, дні									
	7.09 – 11.09	14.09 – 18.09	21.09 – 25.09	28.09 – 2.10	5.10 – 9.10	12.10 – 16.10	19.10 – 23.10	26.10 – 30.10	2.11 – 6.11	9.11 – 13.11
1 Вивчення й аналіз предметної області										
2 Розрахунки										
3 Проєктування програмного продукту										
4 Розробка модулів										
5 Компоновка програми										
6 Тестування і налагодження програмного продукту										

Графік побудований згідно табл. 6.1, де темносірим позначений програміст, а світлосірим – консультант.

6.2 Визначення витрат на розробку програмного продукту

Для визначення витрат на розробку програмного продукту складається калькуляція кошторисної вартості робіт, яка включає наступні пункти:

- основна заробітна плата;
- додаткова заробітна плата;
- відрахування на єдиний соціальний внесок;
- витрати на спеціальне обладнання;
- матеріали і комплектуючі вироби;
- накладні витрати;
- податки [19].

6.3 Розрахунок витрат на оплату праці

Фонд оплати праці складається з основної заробітної плати і додаткової заробітної плати програміста та консультанта, відповідно основна заробітна плата працівника визначається тарифними ставками, посадовими окладами, відрядними розцінками, її розмір залежить від результатів роботи самого працівника [19].

Фонд оплати праці фахівців, грн., розраховується на основі формули (6.1):

$$\text{ФОП} = 3\text{П}_{\text{дод.}} + 3\text{П}_{\text{осн.}}, \quad (6.1)$$

де $3\text{П}_{\text{осн.}}$, $3\text{П}_{\text{дод.}}$ – основна та додаткова заробітна плата фахівців-розроблювачів програмного продукту, грн. [19].

Розраховується основна заробітна плата, грн., на основі формули (6.2):

$$ЗП_{\text{осн.}} = ЗП_{\text{ден.}} + Д_{\text{розр.}} \quad (6.2)$$

де $ЗП_{\text{ден.}}$ – денна заробітна плата фахівця, грн./день;

$Д_{\text{розр.}}$ – тривалість розробки програмного продукту, днів [19].

Денна заробітна плата фахівців, грн./день, розраховується на основі формули (6.3):

$$ЗП_{\text{ден.}} = \frac{О}{Д_{\text{роб.дн.}}}, \quad (6.3)$$

де $О$ – посадовий оклад фахівця, грн. (приймається посадовий оклад консультанта – 8130, програміста – 6000);

$Д_{\text{роб.дн.}}$ – кількість робочих днів в місяці [19].

Обчислимо:

$$ЗП_{\text{ден.прогр.}} = \frac{6000}{22} = 272,73,$$

$$ЗП_{\text{ден.конс.}} = \frac{8130}{22} = 369,54.$$

Розраховується основна заробітна плата програміста та консультанта, грн.:

$$ЗП_{\text{осн.прогр.}} = 272,73 \cdot 50 = 13636,50,$$

$$ЗП_{\text{осн.конс.}} = 369,54 \cdot 10 = 3695,40.$$

Результати розрахунків зведені в табл. 6.3.

Величина додаткової заробітної плати визначається кінцевими результатами діяльності підприємства і виступає у формі премій, винагород, заохочувальних виплат, доплат за особливі умови праці, оплати за

невідпрацьований час (відпустки, лікарняні) [19].

Таблиця 6.3 – Розрахунок основної заробітної плати фахівців

Посада виконавця	Чисельність фахівців, осіб	Місячний оклад фахівців, грн.	Кількість робочих днів розробки програми	Сума основної заробітної плати фахівців, грн.
Програміст	1	6000	50	13636,50
Консультант	1	8130	10	3695,40
Разом	2	-	-	17331,90

Розраховується додаткова заробітна плата фахівців, грн., на основі формули (6.4):

$$ЗП_{\text{дод.}} = ЗП_{\text{осн.}} \cdot K_{\text{дод.з.}}, \quad (6.4)$$

де $K_{\text{дод.з.}}$ – коефіцієнт додаткової заробітної плати, приймається 0,17 [19].

Обчислимо:

$$\begin{aligned} ЗП_{\text{дод. прогр.}} &= 13636,50 \cdot 0,10 = 1363,65, \\ ЗП_{\text{дод. конс.}} &= 3695,40 \cdot 0,10 = 369,54. \end{aligned}$$

Розраховується загальна заробітна плата фахівців, грн.

$$\begin{aligned} ФОП_{\text{конс.}} &= 3695,40 + 369,54 = 4064,94, \\ ФОП_{\text{фах.}} &= 13636,50 + 1363,65 = 15000,15. \end{aligned}$$

Результати розрахунків фонду заробітної плати наводяться у табл. 6.4.

Таблиця 6.4 – Фонд заробітної плати фахівців

Посада виконавця	Основна зарплата, грн.	Додаткова зарплата, грн.	Всього фонд оплати праці, грн.
Програміст	13636,50	1363,65	15000,15
Консультант	3695,40	369,54	4064,94
Разом	17331,90	1733,19	19065,09

Розраховуються відрахування на соціальні заходи – єдиний соціальний внесок (відрахування до Пенсійного фонду, фонду страхування на випадок тимчасової непрацездатності, фонду страхування по безробіттю та фонду страхування від нещасних випадків на виробництві), грн., на основі формули (6.5):

$$B_{\text{с.з.}} = (3П_{\text{дод.}} + 3П_{\text{осн.}}) \cdot K_{\text{с.з.}}, \quad (6.5)$$

де $K_{\text{с.з.}}$ – коефіцієнт відрахувань на соціальні заходи (єдиний соціальний внесок), згідно діючого законодавства 0,22 [19].

Обчислимо:

$$B_{\text{с.з.}} = (17331,90 + 1733,19) \cdot 0,22 = 4194,32.$$

6.4 Визначення витрат на матеріали

До складу матеріальних витрат включають вартість сировини, основних матеріалів, покупних комплектуючих виробів, напівфабрикатів, послуги виробничого характеру сторонніх підприємств, паливо та енергію та технологічні потреби [19].

Прямі матеріальні витрати, грн., розраховується на основі формули (6.6):

$$B_{\text{мат.}} = \sum q_i \cdot C_i, \quad (6.6)$$

де C_i – ціна i -го матеріалу, грн./од.;

q_i – норма витрат матеріалу, нат.од. [19].

Обчислимо:

$$B_{\text{мат1}} = 1 \cdot 120,80 = 120,80,$$

$$B_{\text{мат2}} = 1 \cdot 650,00 = 650,00,$$

$$B_{\text{мат3}} = 1 \cdot 15,50 = 15,50,$$

$$B_{\text{мат4}} = 1 \cdot 138,00 = 138,00,$$

$$B_{\text{мат5}} = 5 \cdot 79,00 = 395,00,$$

$$B_{\text{мат6}} = 1 \cdot 161,00 = 161,00.$$

Розрахунок прямих матеріальних витрат приведений у табл. 6.5.

Таблиця 6.5 – Прямі матеріальні витрати

Назва матеріалу	Одиниця вимірювання	Норма витрат, нат.од.	Ціна, грн./нат.од.	Сума, грн.
1 Папір	пачка	4	207,50	830,00
2 Картридж	од.	1	650,30	650,30
Разом	—	—	—	1480,30

6.5 Визначення витрат на спеціальне устаткування

Ця стаття витрат є комплексною і охоплює зокрема витрати, що включають амортизаційні відрахування, витрати на електроенергію, оплату праці з відповідними відрахуваннями на соціальні заходи робітників, які обслуговують обладнання, при цьому спочатку обов'язково складається річний кошторис витрат на утримання обладнання та відповідно на його

експлуатацію [19].

Річна сума амортизаційних відрахувань, грн., розраховується на основі формули (6.7):

$$A_{\text{річ.}} = B_{\text{перв.}} \cdot \frac{H_a}{100\%}, \quad (6.7)$$

де $B_{\text{перв.}}$ – первісна вартість комп'ютера, грн.;

H_a – річна норма амортизації, % [19].

Первісна вартість комп'ютера, грн., розраховується на основі формули (6.8):

$$B_{\text{перв.}} = B_{\text{придб.}} \cdot (1 + K_{\text{тр.в.}}), \quad (6.8)$$

де $B_{\text{придб.}}$ – вартість придбання (сума, сплачена постачальнику), грн., приймається 6099,00;

$K_{\text{тр.в.}}$ – коефіцієнт транспортних витрат, який враховує витрати на доставку, монтаж та налагодження, приймається 0,05 [19].

Обчислимо:

$$B_{\text{перв.}} = 6099,00 \cdot (1 + 0,05) = 6403,95 \text{ грн.},$$

$$A_{\text{річ.}} = 6403,95 \cdot 0,25 = 1600,99 \text{ грн.}$$

Річні витрати на електроенергію, грн., розраховується на основі формули (6.9):

$$B_{\text{ел.річн.}} = P_n \cdot F_{\text{еф}} \cdot K_{\text{зав}} \cdot K_{\text{зп}} \cdot \Pi_{\text{ел}}, \quad (6.9)$$

де P_n – номінальна потужність персонального комп'ютера (ПК), кВт/год., приймається за паспортом комп'ютера 0,4;

$F_{\text{еф}}$ – річний ефективний фонд роботи ПК, год.;

$K_{\text{зав.}}$ – середній коефіцієнт завантаження за часом, приймається 0,9;

$K_{\text{зп}}$ – коефіцієнт завантаження за потужністю, приймається 0,4;

$\Pi_{\text{ел}}$ – ціна електроенергії, грн./кВт-год., приймається 2,11 [19].

Річний ефективний фонд роботи обчислюється на основі формули (6.10):

$$F_{\text{еф}} = (D_{\text{кал}} - D_{\text{вих}} - D_{\text{св}}) \cdot t_{\text{зм}} - t_{\text{п.св.}} - t_{\text{обсл.}}, \quad (6.10)$$

де $D_{\text{кал}}$, $D_{\text{вих}}$, $D_{\text{св}}$ – відповідно календарні, вихідні та святкові дні (на 2019 рік – 365, 104, 11);

$t_{\text{зм}}$ – тривалість зміни, год., за даними підприємства приймається вісім;

$t_{\text{п.св.}}$ – скорочення робочих днів перед святковими днями на 1 год., на 2019 рік становить сім год.;

$t_{\text{обсл}}$ – час обслуговування ПК, приймається 104 год. [19].

Обчислимо:

$$F_{\text{еф}} = (365 - 104 - 11) \cdot 8 - 7 - 104 = 1889,$$

$$B_{\text{ел.річн.}} = 0,4 \cdot 1889 \cdot 0,9 \cdot 0,4 \cdot 2,11 = 573,95.$$

Розраховується річний фонд заробітної плати обслуговуючого персоналу, за умови, що один місяць – чергова відпустка, використовується формула (6.11):

$$\Phi ОП_{\text{обсл.річн.}} = (ЗП_{\text{осн.обсл.}} + ЗП_{\text{дод.обсл.}}) \cdot \frac{t_{\text{обсл.}}}{F_{\text{еф.}}}, \quad (6.11)$$

де $ЗП_{\text{осн.обсл.}}$ – основна заробітна плата обслуговуючого персоналу, приймається посадовий оклад 7446,00 грн.;

$ЗП_{\text{дод.обсл.}}$ – додаткова заробітна плата обслуговуючого персоналу, грн.,

розраховується за формулою (6.4) [19].

Обчислимо:

$$\begin{aligned} ЗП_{\text{дод.обсл.}} &= 7446,00 \cdot 0,10 = 744,60, \\ \text{ФОП}_{\text{обсл.річн.}} &= (7446,00 + 744,60) \cdot 12/1889 = 52,03 \text{ грн.} \end{aligned}$$

Розраховуються відрахування на соціальні заходи по заробітній платі обслуговуючого персоналу, грн., на основі формули (6.12):

$$В_{\text{с.з.обсл.}} = \text{ФОП}_{\text{обсл.річн.}} \cdot K_{\text{с.з.}} \text{ [19]}. \quad (6.12)$$

Обчислимо:

$$В_{\text{с.з.обсл.}} = 52,03 \cdot 0,22 = 11,45.$$

Розраховуються витрати на поточний ремонт устаткування, грн., на основі формули (6.13):

$$В_{\text{пот.рем.}} = В_{\text{перв.}} \cdot K_{\text{пот.рем.}}, \quad (6.13)$$

де $K_{\text{пот.рем.}}$ – коефіцієнт витрат на поточний ремонт, приймається 0,03 [19].

Обчислимо:

$$В_{\text{пот.рем.}} = 6403.95,00 \cdot 0,03 = 192,12.$$

Розраховуються інші витрати на утримання та експлуатацію обладнання, грн., на основі формули (6.14):

$$В_{\text{ін.}} = (А_{\text{річн.}} + В_{\text{ел.річн.}} + \text{ФОП}_{\text{обсл.}} + В_{\text{с.з.обсл.}} + В_{\text{пот.рем.}}) \cdot K_{\text{інш.}}, \quad (6.14)$$

де $K_{\text{інш.}}$ – коефіцієнт інших витрат, приймається 0,03 [19].

Обчислимо:

$$V_{\text{ін.}} = (1600,99 + 573,95 + 52,03 + 11,45 + 192,12) \cdot 0,03 = 72,89.$$

Розраховуються річні витрати на утримання та експлуатацію обладнання на основі формули (6.15):

$$V_{\text{у.е.о.}} = \Phi_{\text{річн.}} + V_{\text{ел.річн.}} + \Phi\text{ОП}_{\text{обсл.}} + V_{\text{с.з.обсл.}} + V_{\text{пот.рем.}} + V_{\text{ін.}} \quad (6.15)$$

Обчислимо:

$$V_{\text{у.е.о.}} = 1600,99 + 573,95 + 52,03 + 11,45 + 192,12 + 72,89 = 2503,43.$$

Річні витрати на утримання та експлуатацію обладнання (спеціального устаткування) наводяться в табл. 6.6.

Таблиця 6.6 – Кошторис витрат на обладнання

Найменування статей витрат	Сума, грн.
1. Амортизація	1600,99
2. Витрати на електроенергію	573,95
3. Основна та додаткова зарплата обслуговуючого персоналу	52,03
4. Відрахування на соціальні заходи	11,45
5. Витрати на поточний ремонт	192,12
6. Інші витрати	72,89
Разом	2503,43

Розраховуються експлуатаційні витрати на 1 годину роботи ПК, грн./год., на основі формули (6.16):

$$V_{\text{експл.1год.}} = V_{\text{у.е.о.}}/F_{\text{еф.}} [19]. \quad (6.16)$$

Обчислимо:

$$V_{\text{експл.1год.}} = 2503,43/1889 = 1.33.$$

Розраховуються експлуатаційні витрати на весь час роботи ПК при написанні програмного продукту, грн., на основі формули (6.17):

$$V_{\text{експл.}} = V_{\text{експ.1год.}} \cdot t_{\text{м.ч.}}, \quad (6.17)$$

де $t_{\text{м.ч.}}$ – час роботи ПК при написанні програмного продукту, в середньому робота за комп'ютером приймається 8 год./добу [19].

Показник розраховується на основі формули (6.18):

$$t_{\text{м.ч.}} = D_{\text{роб.прогр.}} \cdot 8, \quad (6.18)$$

де $D_{\text{роб.прогр.}}$ – кількість днів роботи програміста при розробці програмного продукту [19].

Обчислимо:

$$\begin{aligned} t_{\text{м.ч.}} &= 50 \cdot 8 = 400, \\ V_{\text{експл.}} &= 1,33 \cdot 400 = 530,11 \text{ грн.} \end{aligned}$$

6.6 Визначення накладних витрат

Накладні (непрямі) витрати – це витрати, які пов'язані не з виготовленням конкретних виробів напряму, а з процесом виробництва в цілому, включаючи зокрема такі витрати, як зарплата управлінського персоналу, утримання та експлуатація будівель, комунальні послуги та ін.,

відповідно розраховуються накладні (непрямі) витрати на основі формули (6.19):

$$V_{\text{накл.}} = (3П_{\text{осн.прогр.}} + 3П_{\text{осн.конс.}}) \cdot K_{\text{накл.в.}}, \quad (6.19)$$

де $K_{\text{накл.в.}}$ – коефіцієнт накладних витрат, приймається 0,50 [19].

Обчислимо:

$$V_{\text{накл.}} = (1363,50 + 3695,40) \cdot 0,50 = 8665,95.$$

Дані розрахунків витрат на розробку програмного продукту наводяться в табл. 6.7.

Таблиця 6.7 – Калькуляція собівартості робіт з розробки програми

Найменування статей витрат	Сума, грн.	Питома вага, %
1. Основна зарплата	17331,90	51,07
2. Додаткова зарплата	1733,19	5,11
3. Відрахування на соціальні заходи	4194,32	12,36
4. Матеріальні витрати	1480,30	4,36
5. Витрати на утримання та експлуатацію обладнання (експлуатаційні витрати)	530,11	1,56
6. Накладні витрати	8665,95	25,54
Разом	33935,77	100,00
Прибуток 30%	10392,50	
Разом	45034,16	
Податок на додану вартість 20%	9006,83	
Разом	54040,99	

Розраховуються загальні витрати на розробку програмного продукту, грн., на основі формули (6.20):

$$V_{\text{заг.}} = 3\Pi_{\text{осн.}} + 3\Pi_{\text{дод.}} + V_{\text{с.с.}} + V_{\text{мат.}} + V_{\text{експл.}} + V_{\text{накл.}} [19]. \quad (6.20)$$

Обчислимо:

$$\begin{aligned} V_{\text{заг.}} &= 17331,90 + 1733,19 + 4194,32 + 1480,30 + 530,11 + 8665,95 = \\ &= 33935,77 \text{ грн.} \end{aligned}$$

За результатами розрахунків, наведених у табл. 6.7, можна зробити висновок, що найбільшу питому вагу у витратах на розробку програмного продукту займають витрати на оплату праці фахівців.

6.7 Оцінка економічної ефективності

На даний момент (до реалізації програмного забезпечення) процес було організовано таким чином, що підтримкою даної діяльності займався аналітик. У результаті впровадження програмного продукту менеджери аукціону зможуть виконувати власну діяльність без звернень до аналітика, тобто аналітика можна буде скоротити.

Час роботи аналітика обходився в 60 гривень. Відповідно 500 годин – 30000 гривень на рік. Програма окупиться менш ніж за 2 роки, що менше навіть самого жорсткого нормативу для автоматизованих систем рівного 3 роки. Розробка програми економічно ефективна.

7 ОХОРОНА ПРАЦІ ТА БЕЗПЕКА В НАДЗВИЧАЙНИХ СИТУАЦІЯХ

7.1 Аналіз потенційних небезпек

У ході виконання магістерської роботи досліджуються та програмно реалізуються методи підтримки проведення аукціону.

При роботі постійно або періодично діють наступні небезпечні і шкідливі чинники:

- перевищення рівнів вібрації застарілого обладнання, що може призвести до сильної втоми, незначних змін функцій організму (порушення серцевої діяльності, нервової системи;
- недостатнє освітлення робочої зони внаслідок несправності освітлювальних приладів, що може призвести до погіршення зору;
- можливість загорянь внаслідок порушень правил пожежної безпеки, що може призвести до пожежі;
- неправильні дії персоналу під час надзвичайних ситуацій внаслідок відсутності управління та паніки, що може призвести до травм та летальних наслідків [20].

7.2 Заходи по забезпеченню безпеки

Для забезпечення ергономічності робочого місця згідно Нормативно-правового акту з охорони праці 0.00-7.15-18 «Вимоги щодо безпеки та захисту здоров'я працівників під час роботи з екранними пристроями» [20], де зазначена заборона захащення робочого місця, дотримання нормативів щодо площини робочого місця на одного працівника, також вимоги до організації робочого місця зазначені в ДСТУ 8604:2015 «Дизайн і ергономіка. Робоче місце для виконання робіт у положенні сидячи. Загальні ергономічні вимоги» [21].

За ступенем небезпеки ураження електричним струмом, відповідно до Правил улаштування електроустановок (ПУЕ), приміщення обчислювального

центру (ОЦ) відноситься до категорії приміщень без підвищеної небезпеки [22].

Було прийнято наступні рішення:

- у ОЦ передбачено використання тільки трьохполюсних розеток, один з полюсів яких підключено до шини захисного заземлення, з цим полюсом електрично з'єднані металеві частини обчислювальних машин, які можуть опинитися під напругою;

- електрична ізоляція забезпечується конструкціями пристроїв і способами їхнього підключення, відповідно до ПУЕ в електроустановках до 1000 В мінімальне значення опору ізоляції силових і освітлювальних електропроводок на ділянці між суміжними запобіжниками або за останніми запобіжниками між будь-яким проведенням і землею становить не менш 0,5 МОм [21].

Використовуються наступні засоби:

- для запобігання тунельного синдрому потрібно в першу чергу правильно підібрати робоче місце та встановити комп'ютер, потрібно мати спеціальний стіл та стілець, рекомендується мати можливість регулювати висоту робочої поверхні від 655 до 740 мм, важливим є простір для ніг користувача: висота – не менше 600мм, ширина – не менше 600 мм, глибина на рівні коліна – не менше 400 мм, на рівні підлоги – не менше 600 мм;

- також рекомендується мати на комп'ютерному робочому місці спеціальний стілець, який: зможе змінювати свою висоту, має можливість регулювати ухил опори для спини (від 0 до 200) , може повертатися, має опір для ліктів [22].

При роботі з мишкою і клавіатурою також слід дотримуватися певних правил: коли ви набираєте текст, рука повинна бути зігнута в лікті під прямим кутом (90°), а при роботі з мишкою стежте, щоб кисть була прямою і лежала на столі якнайдалі від краю, до речі, час роботи з комп'ютером слід обмежити до дійсно необхідного, а щоб попередити тунельний синдром потрібно робити спеціальні вправи для кистей, комплекс вправ потрібно

повторювати приблизно кожні 45 хвилин, тривалість однієї вправи – 1-2 хв. [23].

7.3 Заходи з виробничої санітарії і гігієни праці

У обчислювальному центрі, згідно Державних будівельних норм В.2.5-28:2018 «Природне та штучне освітлення» [24] передбачене природне та штучне освітлення, штучне освітлення в приміщенні, здійснено системою загального рівномірного освітлення, як джерела штучного освітлення в приміщенні застосовані люмінесцентні лампи типу ЛБ, при застосуванні яких дотримались наступних умов:

- температура навколишнього повітря не повинна бути нижче, ніж 5°C ;
- напруга на освітлювальних приладах повинна бути не менше, ніж 90 % номінальної [25].

Розрахуємо систему загального рівномірного штучного освітлення в офісному приміщенні.

Приймаємо розміри приміщення довжина $A = 15$ м, ширина $B = 9$ м, висота $H = 4$ м. Висота робочої поверхні складає $h = 0,7$ м. Коефіцієнти відбиття поверхонь приміщення: $\rho_{\text{с}}=70\%$; $\rho_{\text{ст}}=50\%$; $\rho_{\text{п}}=30\%$.

Згідно з Державних будівельних норм В.2.5-28:2018. Природне та штучне освітлення рівень нормованого загального освітлення в приміщенні нормуються як $E_{\text{н}} = 300$ лк.

Обираємо світильник типу ЛВП, призначений для приміщень висотою до 4,5 м. IP=54 – ступінь захисту світильника від пилу та бризок води, $L/h] = 1,3$ – числове значення коефіцієнта світильника.

Коефіцієнт запаса k_z приймаємо $k_z = 1,4$. Коефіцієнт нерівномірності (мінімального) освітлення z для люмінесцентних ламп низького тиску, як правило дорівнює $z = 1,1$.

Розрахуємо кількість рядів світильників у приміщенні:

$$N_p = \frac{B}{(H-h_p) \cdot [L/h]} = \frac{9}{(4-0,7) \cdot 1,3} = 2,1 \approx 3 \text{ ряди світильників.}$$

Визначимо максимально припустиму відстань між рядами світильників за формулою (7.1):

$$L_{\max} = \frac{B}{N_p}, \quad (7.1)$$

де B – ширина приміщення;

N_p – кількість рядів.

Розрахуємо:

$$L_{\max} = \frac{9}{3} = 3 \text{ м.}$$

Розраховуємо висоту підвісу світильника над робочою поверхнею за формулою (7.2):

$$h = \frac{L_{\max}}{[L/h]}. \quad (7.2)$$

Розрахуємо:

$$h = \frac{3}{1,3} = 2,3 \text{ м.}$$

Знайдемо висоту звисання світильника від стелі на основі формули (7.3):

$$h_3 = H - h_p - h, \quad (7.3)$$

де H – висота приміщення;

h_p – висота робочої поверхні;

h – висота підвісу світильника над робочою поверхнею.

Розрахуємо:

$$h_3 = 4 - 0,7 - 2,3 = 1 \text{ м.}$$

Чисельне значення індексу приміщення визначаємо за рівнянням (7.4):

$$i = \frac{AB}{h(A + B)}, \quad (7.4)$$

де A – довжина приміщення, м;

B – ширина приміщення, м;

h – висота розміщення світильників над робочою поверхнею, м.

Розрахуємо:

$$i = \frac{15 \cdot 9}{2,3 \cdot (15 + 9)} = 2,44.$$

Визначаємо сумарний світловий потік освітлювальної установки у даному виробничому приміщенні на основі формули (7.5):

$$\Phi_{\Sigma} = \frac{E_n \cdot S \cdot k_3 \cdot z}{\eta}, \quad (7.5)$$

де Φ_{Σ} – розрахункове значення сумарного світлового потоку у приміщенні, лм;

E_n – нормоване значення освітленості, лк;

S – площа освітлюваної поверхні, м²;

k_3 – коефіцієнт запасу;

z – коефіцієнт нерівномірності (мінімальної) освітленості;

η – коефіцієнт використання світлового потоку.

Розрахуємо:

$$\Phi_{\Sigma} = \frac{300 \cdot 15 \cdot 9 \cdot 1,4 \cdot 1,1}{0,58} = 107535 \text{ лм.}$$

Визначаємо умовну загальну кількість світильників у приміщенні на основі формули (7.6):

$$N^* = A \cdot \frac{B}{L_{\max}^2}. \quad (7.6)$$

Розрахуємо:

$$N^* = \frac{15 \cdot 9}{3 \cdot 3} = 15 \approx 15 \text{ шт.}$$

Кратне числу рядів $N_p = 3$.

Розрахуємо світловий потік умовного джерела світла на основі формули (7.7):

$$\Phi_{\text{л}}^* = \frac{\Phi_{\Sigma}}{N_{\text{л}}}, \text{ лм,} \quad (7.7)$$

де Φ_{Σ} – розрахункове значення сумарного світлового потоку у приміщенні, лм;

$N_{\text{л}}$ – загальна кількість ламп у приміщенні.

Розрахуємо на основі формули (7.8):

$$N_{\text{л}} = N^* \cdot n, \quad (7.8)$$

де n – кількість ламп у світильнику (в ЛВП світильнику знаходиться 2 лампи).

Розрахуємо:

$$N_{\text{л}} = 15 \cdot 2 = 30 \text{ шт.},$$

$$\Phi_{\text{л}} = \frac{107535}{30} = 3585 \text{ лм.}$$

Обираємо лампу ЛДЦ із $P_{\text{л}}=80$ Вт, $\Phi_{\text{л}} = 3800$ лм, використовуємо формулу (7.9):

$$m = \frac{\Phi_{\text{л}}^*}{\Phi_{\text{л}}}. \quad (7.9)$$

Розрахуємо:

$$m = \frac{3585}{3800} = 0,94.$$

Визначаємо оптимальну (фактичну) кількість світильників у приміщенні на основі формули (7.10):

$$N = N^* \cdot m. \quad (7.10)$$

Розрахуємо:

$$N = 15 \cdot 0,94 \approx 15 \text{ шт.}$$

Фактична кількість ламп у приміщенні обчислюється за формулою (7.11):

$$N_{\phi\lambda} = N_{\phi} \cdot n. \quad (7.11)$$

Розрахуємо:

$$N_{\phi\lambda} = 15 \cdot 2 = 30 \text{ шт.}$$

Визначаємо загальну розрахункову освітленість E_p у приміщенні, що створюється при застосуванні стандартних ламп за формулою (7.12):

$$E_p = \frac{\Phi_{\lambda} N_{\phi\lambda} \eta}{Sk_z z} \text{ лк.} \quad (7.12)$$

Розрахуємо:

$$E_p = \frac{3800 \cdot 30 \cdot 0,5}{15 \cdot 9 \cdot 1,4 \cdot 1,1} = 274 \text{ лк.}$$

Тобто виконується умова $E_p = (-10\% \dots +20\%) \cdot E_n$, лк.

Роботи в офісному приміщенні, належать до категорії Іб - легка робота, тому передбачені наступні оптимальні значення параметрів мікроклімату:

– у холодний період року: температура 21-23°C; відносна вологість: 40-60%; швидкість переміщення повітря: 0,1 м/с

– у теплий період року: температура 22-24°C; відносна вологість: 40-60%; швидкість переміщення повітря: 0,2 м/с [26].

Під впливом шуму знижується концентрація уваги, порушуються фізіологічні функції, рівні звукового тиску в октавних смугах частот, та еквівалентні рівні звуку на робочих місцях у приміщенні ОЦ нормуються згідно Державних санітарних норм 3.3.6.037-99 «Санітарні норми виробничого шуму, ультразвуку та інфразвуку» [27]. Оптимальний рівень

шуму в приміщення забезпечується за допомогою:

- використання блоків живлення ПК з вентиляторами на гумових підвісках;
- переведення жорсткого диска в режим сну (Standby), якщо комп'ютер не працює протягом визначеного часу [20].

7.4 Заходи безпеки у надзвичайних ситуаціях

7.4.1 Заходи з пожежної безпеки

Великою небезпекою для обчислювального центру є пожежа, відносно якої можна виділити наступне:

- обчислювальний центр наповнений різноманітними електричними приладами, а пожежа основний фактор для їх знищення, що може призвести до великих матеріальних втрат, так як в приміщенні обчислювального центру присутні всі три основні чинники, необхідні для виникнення пожежі: горючі речовини, окислення і джерела запалювання;

- для забезпечення пожежної безпеки проводилась пожежна профілактика, яка включає в себе комплекс організаційних і технічних заходів, спрямованих на забезпечення безпеки людей, на запобігання пожежі, обмеження її поширення, а також на створення умов для успішного гасіння пожежі [28].

Обчислювальний центр відноситься до категорії «П-Па» згідно ДСТУ Б В.1.1-36:2016 «Визначення категорій приміщень, будинків та зовнішніх установок за вибухопожежною та пожежною небезпекою», згідно Державних будівельних норм В.1.1-7:2016 «Пожежна безпека об'єктів будівництва. Загальні вимоги», а клас можливої пожежі визначається, як «А» [28]-[29].

Під час прийняття на роботу і в процесі праці усі працівники обчислювального центру повинні проходити протипожежний інструктаж, перевірку знань з питань пожежної безпеки [20].

7.4.2 Заходи з цивільного захисту

Вплив електромагнітного імпульсу на елементи виробництва:

– ядерні вибухи в атмосфері й більш високих шарах призводять до виникнення потужних електромагнітних полів з довжиною хвиль від 1 до 1000 м і більше, ці поля через короткочасне існування називають електромагнітним імпульсом, електромагнітний імпульс виникає при ядерному вибуху у воєнний час, у мирний час – при випробуванні ядерної зброї або ядерних аваріях і катастрофах в атмосфері й космосі;

– основною причиною виникнення електромагнітного імпульсу тривалістю менше 1 с вважають взаємодію гамма-променів і нейтронів ядерного вибуху з атомами газів повітря, внаслідок чого з них вибиваються електрони (ефект Комптона) і хаотично розлітаються в середовищі позитивно заряджених атомів газів, важливе значення має також виникнення асиметрії в розподілі просторових електричних зарядів, пов'язаних з особливостями поширення гама-променів і утворення електронів;

– електромагнітний імпульс небезпечний і за наявності міцних споруд, розрахованих на стійкість проти ударної хвилі наземного ядерного вибуху, проведеного на відстані кількох сотень метрів [30].

ВИСНОВКИ

Розглянуто проблему проведення електронних аукціонів, проаналізовано існуючі типи аукціонів, виконано огляд програмних засобів для проведення аукціонів. Проаналізовано методи підтримки аукціонної діяльності, які на даний момент на практиці функціонують в основному за допомогою експертних оцінок, проаналізовано існуючі методи пошуку подібних зображень, серед яких розглянуто перцептивні геші та нейронні мережі, розглянуто принципи побудови і функціонування згорткових нейронних мереж з детальним визначенням структури нейронної мережі VGG-19, запропоновано метод підтримки проведення аукціону.

Виконано порівняння мов програмування C# і Java для реалізації програмного забезпечення, мову C# обрано для реалізації. Виділено функціональні вимоги до розроблюваного програмного забезпечення, побудовано діаграми прецедентів для роботи покупця, продавця та адміністратора з програмою. Забезпечено проєктування бази даних з приведенням структурної схеми бази даних. Систему керування базами даних Microsoft SQL Server обрано для реалізації. Забезпечено проєктування структури програмного застосунку з приведенням структурної схеми.

Виконано реалізацію програмного забезпечення підтримки проведення аукціону. Описано алгоритм функціонування програмного забезпечення з приведенням діаграми станів для лоту аукціону, реалізовано модулі та класи програмного забезпечення, описано основні розроблені класи з приведенням відповідних методів та описом їх роботи. Описано процес роботи користувача з програмою.

Усі поставлені завдання на роботу виконано.

Наукова новизна роботи полягає в запропонованому методі підтримки проведення аукціону, проведення якого ґрунтується на моделі відкритого багатораундового аукціону, який використовує згорткову нейронну мережу на основі моделі архітектури VGG-19 для виділення подібних зображень

лотів, що дозволяє узгоджувати початкову вартість лоту з подібними раніше проданими лотами, отримувати оцінку фактичної вартості лоту продавцям та знаходити подібні лоти для забезпечення синхронності аукціону для покупців.

ПЕРЕЛІК ДЖЕРЕЛ ПОСИЛАННЯ

1. Щепкина, Н. Н. Электронные аукционы в условиях цифровой трансформации [Текст] : достижения и просчеты / Н. Н. Щепкина // Проблемы економіки. – 2018. – № 2. – С. 134-140.
2. Аукционы [Электрон. ресурс]. – Режим доступа : <https://topdealo.com/auksiony/>
3. Крупнейшие онлайн-аукционы мира [Электрон. ресурс]. – Режим доступа : <https://fcti.by/2017/01/28/8965/>
4. Володина, Е. Е. Классификация и характеристика способов проведения аукционов по продаже радиочастотного спектра [Текст] / Е. Е. Володина, Е. Е. Девяткин, Т. А. Суходольская // Труды НИИР. – 2013. – № 4. – С. 11-17.
5. Rizwan, M. eBay UK Review | Key Features Of eBay Kleinanzeigen [Electronic resource] / M. Rizwan. – Access mode : <https://thekrjaan.com/ebay-uk-review-key-features-of-ebay-kleinanzeigen/>
6. Как создать онлайн аукцион не хуже, чем eBay [пошаговое руководство] [Электрон. ресурс]. – Режим доступа : <https://dinarys.com/ru/blog/how-to-build-an-auction-website-like-ebay>
7. Яровая, М. Интернет-аукцион Violyty [Электрон. ресурс] : что это такое и как на нем продавать / М. Яровая. – Режим доступа : <https://ain.ua/2020/04/18/instrukciya-violity/>
8. Калугина, О.П. Аукционы, их сущность, ценообразование и совершенствование функционирования [Текст] : диссертация на соискание ученой степени кандидата экономических наук по специальности 08.00.01 – политическая экономия / О.П. Калугина. – Воронеж : ВГАСА, 2000. – 121 с.
9. Исследование схожести изображений с помощью перцептивных хэш-алгоритмов и расстояний Хэмминга [Текст] / Т. А. Паромова, И. Я. Зеленева, Н. В. Луценко, С. А. Кленачов // Наукові праці Донецького національного технічного університету. Серія : Інформатика, кібернетика та

обчислювальна техніка. – 2016. – Вип. 2. – С. 62-68.

10. Нейронные сети [Электрон. ресурс] : распознавание образов и изображений с помощью ИИ. – Режим доступа : <https://center2m.ru/ai-recognition>

11. Saha, S. A Comprehensive Guide to Convolutional Neural Networks – the ELI5 way [Electronic resource] / S. Saha. – Access mode : <https://towardsdatascience.com/a-comprehensive-guide-to-convolutional-neural-networks-the-eli5-way-3bd2b1164a53>

12. Сверточная нейронная сеть, часть 1 [Электрон. ресурс] : структура, топология, функции активации и обучающее множество. – Режим доступа : <https://habr.com/ru/post/348000/>

13. Zheng, Yu. Breast Cancer Screening Using Convolutional Neural Network and Follow-up Digital Mammography [Electronic resource] / Yufeng Zheng, Clifford Yang, Alex Merkulov // Computational Imaging : conference. – 2018. – Access mode : https://www.researchgate.net/publication/325137356_Breast_cancer_screening_using_convolutional_neural_network_and_follow-up_digital_mammography

14. Understanding the VGG19 Architecture [Electronic resource] . – Access mode : <https://iq.opengenus.org/vgg19-architecture/>

15. Vegetable Image Retrieval with Fine-tuning VGG Model and Image Hash [Electronic resource] / Zhaolu Yang, Jun Yue, Zhenbo Li, Ling Zhu. – IFAC-PapersOnLine. – 2018. – Volume 51, Issue 17. – Pp. 280-285. – Access mode : <https://www.sciencedirect.com/science/article/pii/S240589631831293X>

16. Wong, A. Building a Similar Images Finder without any training! [Electronic resource] / A. Wong. – Access mode : <https://towardsdatascience.com/building-a-similar-images-finder-without-any-training-f69c0db900b5>

17. Ramesh, A. Content Based Image Retrieval Using Autoencoders (vgg19 and convolutional neural network) [Electronic resource] / A. Ramesh. – Access mode : <https://medium.com/@arunadeviramesh1011/content-based-image->

retrieval-using-autoencoders-vgg19-and-convolutional-neural-network-354a1773de02

18. С# против Java [Електрон. ресурс] : какой язык программирования общего назначения выбрать? – Режим доступа : https://itvdn.com/ru/blog/article/csharp_vs_java

19. Остапенко, В.В. Методичні вказівки по економічному обґрунтуванню дипломних проєктів для студентів спеціальностей 7.080403 «Програмне забезпечення автоматизованих систем», 7.080402 «Інформаційні технології проєктування» і 7.091501 «Комп'ютерні та інтелектуальні системи і мережі» всіх форм навчання [Текст] / В.В. Остапенко, Е.М. Касьян, Є.М. Сердюк. – Запоріжжя : ЗНТУ, 2009. – 22 с.

20. Вимоги щодо безпеки та захисту здоров'я працівників під час роботи з екранними пристроями. НПАОП 0.00-7.15-18. Державні санітарні правила і норми [Текст]. – Харків : Форт, 2018. – 48 с.

21. ДСТУ 8604:2015 Дизайн і ергономіка. Робоче місце для виконання робіт у положенні сидючи. Загальні ергономічні вимоги [Текст]. – Київ : Держспоживстандарт України, 2015. – 15 с.

22. ДСТУ 7237:2011 Система стандартів безпеки праці. Електробезпека. Загальні вимоги та номенклатура видів захисту [Текст]. – Київ : Держспоживстандарт України, 2011. – 10 с.

23. Охорона праці в галузі. Навчальний посібник [Текст] / П. Атаманчук, В. Мендерецький, О. Панчук, Р. Билык. – К. : Центр навчальної літератури, 2017. – 322 с.

24. ДБН В.2.5-28:2018. Природне та штучне освітлення. Посібник [Текст] : проблеми природного і штучного освітлення. – К. : Етін, 2019. – 180 с.

25. Шмирко, В.І. Методичні вказівки до лабораторного заняття «Розрахунок загального рівномірного штучного освітлення виробничих приміщень» з дисципліни «Цивільний захист і охорона праці в галузі» [Текст] : для студентів усіх спеціальностей та форм навчання / В.І. Шмирко,

О.В. Коробко, Ю.І. Троян. – Запоріжжя : каф. ОПіНС. НУ «Запорізька політехніка», 2020. – 36 с.

26. ДСН 3.3.6.042-99 Санітарні норми мікроклімату виробничих приміщень [Текст]. – Харків : Форт, 2000. – 24 с.

27. ДСН 3.3.6.037-99 Санітарні норми виробничого шуму, ультразвуку та інфразвуку [Текст]. – К. : МОЗ України, 1999. – 24 с.

28. ДСТУ Б В.1.1-36:2016 Визначення категорій приміщень, будинків та зовнішніх установок за вибухопожежною та пожежною небезпекою [Текст]. – Київ : Мінрегіонбуд України, 2016. – 66 с.

29. ДБН В.1.1-7:2016 Пожежна безпека об'єктів будівництва. Загальні вимоги [Текст]. – Київ : Міністерство регіонального розвитку, будівництва та житлово-комунального господарства України, 2017. – 39 с.

30. Стеблюк, М. І. Цивільна оборона та цивільний захист [Текст] : навч. посіб. для вузів / М. І. Стеблюк. – К. : Знання, 2013. – 487 с.

ДОДАТОК А
Технічне завдання

Вступ

Мета роботи – розроблення програмного забезпечення підтримки проведення аукціону для забезпечення обґрунтованого прийняття рішень щодо вибору початкових параметрів лотів.

A.1 Підстави для розробки

Підставою для розробки є завдання на дипломну кваліфікаційну роботу магістра за темою “Дослідження та програмна реалізація методів підтримки проведення аукціону”, затверджене наказом № 301 від 3 листопада 2020 р. за Національним університетом «Запорізька політехніка».

A.2 Призначення розробки

Програма призначена для підтримки проведення аукціону на різних етапах, що зокрема включають внесення даних лотів, забезпечення конкурсу під час реалізації аукціону, визначення переможців аукціону з можливістю декількох переможців одночасно.

A.3 Основні вимоги до програми

A.3.1 Вимоги до функціональних характеристик

Вимоги до функціональних характеристик програми включають наступний набір:

- додавання лотів і внесення їх даних;
- керування лотом;
- пошук лотів;
- перегляд лотів;

- створення пропозиції щодо купівлі лоту;
- перегляд власних пропозицій;
- перегляд власних проданих лотів;
- перегляд власних активних лотів;
- перегляд власних куплених лотів;
- реєстрація користувачів;
- авторизація користувачів;
- видалення лотів;
- автоматичне визначення переможців аукціонів.

A.3.2 Вимоги до надійності

Надійність функціонування має забезпечуватися коректною експлуатацією програмних та технічних засобів, а також за допомогою програмної перевірки даних, які вводиться користувачем для уникнення помилок під час роботи програми.

A.3.3 Умови експлуатації

Розроблюваний програмний застосунок є Windows-сумісним, тобто для його експлуатації має виконуватися умова підтримки операційної системи Windows. Виходячи з того, що актуальна підтримка зараз здійснюється для версії 7, то саме вона є мінімальною вимогою експлуатації програми. Для роботи програмного забезпечення має бути обов'язково встановлено в системі фреймворк .NET версії не нижче ніж 4.7.

A.3.4 Вимоги до складу та параметрів технічних засобів

Вимоги до складу та параметрів технічних засобів на сервері включають:

- чотирьохядерний процесор з частотою не меншою за 3 ГГц;
- обсяг оперативного запам'ятовуючого пристрою повинен бути не менше за 4 Гб;
- обсяг вільного простору на жорсткому диску – не менше 100 Гб.

А.3.5 Вимоги до транспортування та збереження

Транспортуватися та зберігатися програма має за правилами, які визначено для роботи з носіями та серверами, на яких програма розповсюджується або зберігається.

А.3.6 Вимоги до програмної документації

Вимоги до програмної документації включають необхідність розробки «Тексту програми», що оформлюється в пояснювальній записці додатком і до якого поміщається фрагмент коду програми, та дане «Технічне завдання».

А.3.7 Вимоги до маркування та упакування

Вимоги до маркування та упакування включають необхідність наявності на відповідному носії назви проєкту – «Дослідження та програмна реалізація методів підтримки проведення аукціону».

ДОДАТОК Б
Текст програми

```

//Текст файла AS.cs
using System;
using System.Collections.Generic;
using System.Text;
using System.IO;

namespace ASLibrary
{
    public class Lot : Interface
    {
        protected int id;
        protected string name;
        protected List<string> info;
        protected double price;
        protected int count;
        protected Account author;
        protected int type;
        protected DateTime begindate;
        protected DateTime enddate;
        protected List<String> images;
        private DatabaseProcessor dproc = new
DatabaseProcessor();

        public int ID
        {
            get
            {
                return this.id;
            }
            set
            {
                this.id = value;
            }
        }

        public string Name
        {
            get
            {
                return this.name;
            }
            set
            {
                this.name = value;
            }
        }

        public List<string> Info
        {
            get
            {
                return this.info;
            }
        }
    }
}

```

```

    }
    set
    {
        this.info = value;
    }
}

public double Price
{
    get
    {
        return this.price;
    }
    set
    {
        this.price = value;
    }
}

public int Count
{
    get
    {
        return this.count;
    }
    set
    {
        this.count = value;
    }
}

public Account Author
{
    get
    {
        return this.author;
    }
    set
    {
        this.author = value;
    }
}

public int Type
{
    get
    {
        return this.type;
    }
    set
    {
        this.type = value;
    }
}

```

```

    }
    public DateTime BeginDate
    {
        get
        {
            return this.begindate;
        }
        set
        {
            this.begindate = value;
        }
    }

    public DateTime EndDate
    {
        get
        {
            return this.enddate;
        }
        set
        {
            this.enddate = value;
        }
    }

    public List<String> Images
    {
        get
        {
            return this.images;
        }
        set
        {
            this.images = value;
        }
    }

    public Lot()
    {
        dproc.open();
        Id = dproc.detectMaxLot() + 1;
        dproc.close();
        Name = "";
        Info = new List<string>();
        Price = 0;
        Type = 0;
        Begindate = DateTime.UtcNow;
        Enddate = DateTime.UtcNow;
        Images = new List<String>();
        Author = new Account();
        Count = 0;
    }

```

```

public Lot(int q)
{
    Id = -1;
    Name = "";
    Info = new List<string>();
    Price = 0;
    Type = 0;
    Begindate = DateTime.UtcNow;
    Enddate = DateTime.UtcNow;
    Images = new List<String>();
    Author = new Account();
    Count = 0;
}

}

public class Account : Interface
{
    protected int id;
    protected string login;
    protected string password;
    protected int type;

    public int Id
    {
        get
        {
            return this.id;
        }
        set
        {
            this.id = value;
        }
    }

    public string Login
    {
        get
        {
            return this.login;
        }
        set
        {
            this.login = value;
        }
    }

    public string Password
    {
        get
        {

```

```

        return this.password;
    }
    set
    {
        this.password = value;
    }
}

public string Type
{
    get
    {
        return this.type;
    }
    set
    {
        this.type = value;
    }
}
}

public class Auctioneer : Account, Interface
{
    protected int ida;
    protected string surname;
    protected string name;
    protected string fathename;
    protected int typea;
    protected string mail;
    protected string phone;
    protected string address;
    protected List<string> info;

    public int Ida
    {
        get
        {
            return this.ida;
        }
        set
        {
            this.ida = value;
        }
    }

    public string Surname
    {
        get
        {
            return this.surname;
        }
        set
        {

```

```
        this.surname = value;
    }
}

public string Name
{
    get
    {
        return this.name;
    }
    set
    {
        this.name = value;
    }
}

public string Fathername
{
    get
    {
        return this.fathername;
    }
    set
    {
        this.fathername = value;
    }
}

public int Typea
{
    get
    {
        return this.typea;
    }
    set
    {
        this.typea = value;
    }
}

public string Mail
{
    get
    {
        return this.mail;
    }
    set
    {
        this.mail = value;
    }
}

public string Phone
```

```

    {
        get
        {
            return this.phone;
        }
        set
        {
            this.phone = value;
        }
    }

    public string Address
    {
        get
        {
            return this.address;
        }
        set
        {
            this.address = value;
        }
    }

    public List<string> Info
    {
        get
        {
            return this.info;
        }
        set
        {
            this.info = value;
        }
    }
}

public class Salesman : Auctioneer, Interface
{
    private DatabaseProcessor dproc = new
    DatabaseProcessor();
    private List<Lot> activeLots;
    private List<Lot> doneLots;

    public List<Lot> ActiveLots
    {
        get => activeLots;
        set => activeLots = value;
    }

    public List<Lot> DoneLots
    {
        get => doneLots;
    }
}

```

```

        set => doneLots = value;
    }
    public Salesman(int q)
    {
        Id = -1;
        Surname = "";
        Name = "";
        Fathername = "";
        Typea = 0;
        Mail = "";
        Phone = "";
        Address = "";
        Login = "";
        Password = "";
        Type = 0;
        Info = new List<String>();
        ActiveLots = new List<Lot>();
        DoneLots = new List<Lot>();
    }

    public Salesman()
    {
        dproc.open();
        Id = dproc.detectMaxAccount() + 1;
        Ida = dproc.detectMaxAuctionerr() + 1;
        dproc.close();
        Surname = "";
        Name = "";
        Fathername = "";
        Typea = 0;
        Mail = "";
        Phone = "";
        Address = "";
        Login = "";
        Password = "";
        Type = 0;
        Info = new List<String>();
        ActiveLots = new List<Lot>();
        DoneLots = new List<Lot>();
    }

    public void removeLot(int q)
    {
        foreach (Lot l in this.ActiveLots)
        {
            if (l.Id == q)
            {
                this.ActiveLots.Remove(l);
            }
        }
        dproc.open();
        dproc.changeLotTypeToDel(q);
        dproc.close();
    }

```

```

    }

    public void ChangeLotType(int lot, int type)
    {
        int z = 0;
        foreach (Lot l in this.ActiveLots)
        {
            if (l.Id == lot)
            {
                this.ActiveLots[z].Type = type;
            }
            z++;
        }
        z = 0;
        foreach (Lot l in this.DoneLots)
        {
            if (l.Id == lot)
            {
                this.DoneLots[z].Type = type;
            }
            z++;
        }
        dproc.open();
        dproc.changeLotType(lot, type);
        dproc.close();
    }

    public void buyLot(Lot good)
    {
        foreach (Lot l in this.ActiveLots)
        {
            if (l.Id == good.Id)
            {
                if (l.Count == good.Count)
                {
                    this.DoneLots.Add(l);
                    this.ActiveLots.Remove(l);
                    break;
                }
                if (l.Count > good.Count)
                {
                    l.Count -= good.Count;
                    break;
                }
            }
        }
        dproc.open();
        dproc.changeLotTypeToBought(good.ID);
        dproc.close();
    }

    public void Save()
    {

```

```

        dproc.open();
        dproc.AddSalesman(this.Ida, this.Surname, this.Name,
this.Fathername, this.Typea, this.Mail, this.Phone, this.Address,
this.Info);

        foreach (Lot l in this.ActiveLots)
        {
            Dproc.AddSalesmanLot(l.Id, l.Name, l.Info,
l.Price, l.Count, l.Type, l.BeginDate, l.EndDate, l.Images,
l.Author.Id);
        }
        dproc.close();
    }

    public void getSalesman(int q)
    {
        dproc.open();
        Surname = dproc.getSalesmanSurname(q);
        Name = dproc.getSalesmanName(q);
        Fathername = dproc.getSalesmanFatherName(q);
        Typea = dproc.getSalesmanType(q);
        Mail = dproc.getSalesmanMail(q);
        Phone = dproc.getSalesmanPhone(q);
        Address = dproc.getSalesmanAddress(q);
        Info = dproc.getSalesmanInfo(q);
        List<int> LotIds = dproc.getSalesmanLotIds(q);
        foreach (int l in LotIds)
        {
            Lot tl = new Lot();
            tl.Id = l;
            tl.Name = dproc.getLotName(l);
            tl.Info = dproc.getLotInfo(l);
            tl.Price = dproc.getLotPrice(l);
            tl.Count = dproc.getLotCount(l);
            int acc = dproc.getLotAuthor(l);
            tl.Author = new Account();
            tl.Author.Id = acc;
            tl.Type = dproc.getLotType(l);
            tl.Begindate = dproc.getLotBeginDate(l);
            tl.Enddate = dproc.getLotEndDate(l);
            tl.Images = dproc.getLotImages(l);
            if (tl.Type == 1)
                this.ActiveLots.Add(tl);
            else if (tl.Type == 2)
                this.DoneLots.Add(tl);
        }
        dproc.close();
    }

    public void addLot(Lot good)
    {
        ActiveLots.Add(good);
    }

```

```

    }

    public class Buyer : Auctioneer, Interface
    {
        private DatabaseProcessor dproc = new
DatabaseProcessor();
        private List<Lot> activeLots;
        private List<Lot> doneLots;
        private List<Lot> activePropositions;
        private List<Lot> donePropositions;
        private List<string> notifications;

        public List<Lot> ActivePropositions
        {
            get => activePropositions;
            set => activePropositions = value;
        }

        public List<Lot> DonePropositions
        {
            get => donePropositions;
            set => donePropositions = value;
        }

        public List<Lot> ActiveLots
        {
            get => activeLots;
            set => activeLots = value;
        }

        public List<Lot> DoneLots
        {
            get => doneLots;
            set => doneLots = value;
        }

        public List<string> Notifications
        {
            get => notifications;
            set => notifications = value;
        }

        public Buyer(int q)
        {
            Id = -1;
            Ida = -1;
            Surname = "";
            Name = "";
            Fathername = "";
            Typea = 0;
            Mail = "";
            Phone = "";
            Address = "";
        }
    }

```

```

        Login = "";
        Password = "";
        Type = 0;
        Info = new List<String>();
        ActivePropositions = new List<Proposition>();
        DonePropositions = new List<Proposition>();
        ActiveLots = new List<Lot>();
        DoneLots = new List<Lot>();
        Notifications = new List<string>();
    }

    public Buyer()
    {
        dproc.open();
        Id = dproc.detectMaxAccount() + 1;
        Ida = dproc.detectMaxAuctionerr() + 1;
        dproc.close();
        Surname = "";
        Name = "";
        Fathername = "";
        Typea = 0;
        Mail = "";
        Phone = "";
        Address = "";
        Login = "";
        Password = "";
        Type = 0;
        Info = new List<String>();
        ActivePropositions = new List<Proposition>();
        DonePropositions = new List<Proposition>();
        ActiveLots = new List<Lot>();
        DoneLots = new List<Lot>();
        Notifications = new List<string>();
    }

    public void buy(Lot good)
    {
        foreach (Lot l in this.ActiveLots)
        {
            if (l==good)
            {
                this.ActiveLots.Remove(l);
                l.Type = 2;
                this.DoneLots.Add(l);
                break;
            }
        }
        foreach (Proposition l in this.ActivePropositions)
        {
            if (l.Good==good)
            {
                this.ActivePropositions.Remove(l);
                l.Type = 2;
            }
        }
    }

```

```

        this.DonePropositions.Add(l);
        this.Notifications.Add("Вами придбано лот
"+good.Name);

        break;
    }
}

public void Save()
{
    dproc.open();
    dproc.AddBuyer(this.Ida, this.Surname, this.Name,
this.Fathername, this.Typea, this.Mail, this.Phone, this.Address,
this.Info);

    foreach (Proposition l in this.ActivePropositions)
    {
        Dproc.AddProposition(l.Good.Id, l.Name, l.Info,
l.Price, l.Count, l.Type, l.BeginDate, l.EndDate, l.Images,
l.Author.Id);
    }
    dproc.close();
}

public void getBuyer(int q)
{
    dproc.open();
    Surname = dproc.getBuyerSurname(q);
    Name = dproc.getBuyerName(q);
    Fathername = dproc.getBuyerFatherName(q);
    Typea = dproc.getBuyerType(q);
    Mail = dproc.getBuyerMail(q);
    Phone = dproc.getBuyerPhone(q);
    Address = dproc.getBuyerAddress(q);
    Info = dproc.getBuyerInfo(q);
    List<int> ProposIds =
dproc.getBuyerPropositionIds(q);
    foreach (int l in ProposIds)
    {
        Proposition p = new Proposition();
        p.Id = l;
        p.Price = dproc.getPropositionPrice(l);
        p.Date = dproc.getPropositionDate(l);
        p.Type = dproc.getPropositionType(l);
        Lot tl = new Lot();
        tl.Id = l;
        tl.Name = dproc.getLotName(l.Good.Id);
        tl.Info = dproc.getLotInfo(l.Good.Id);
        tl.Price = dproc.getLotPrice(l.Good.Id);
        tl.Count = dproc.getLotCount(l.Good.Id);
        int acc = dproc.getLotAuthor(l.Good.Id);
        tl.Author = new Account();
        tl.Author.Id = acc;
        tl.Type = dproc.getLotType(l.Good.Id);
    }
}

```

```

        tl.Begindate = dproc.getLotBeginDate(l.Good.Id);
        tl.Enddate = dproc.getLotEndDate(l.Good.Id);
        tl.Images = dproc.getLotImages(l.Good.Id);
        p.Good = tl;
        acc = dproc.getPropositionAuthor(l.Good.Id);
        p.User = new Account();
        p.User.Id = acc;
        if (p.Type == 1)
        {
            this.ActivePropositions.Add(p);
            this.ActiveLots.Add(p.Good);
        }
        else if (p.Type == 2)
        {
            this.DonePropositions.Add(p);
            this.DoneLots.Add(p.Good);
        }
    }
    dproc.close();
}

public List<Lot> SearchLots(String name)
{
    dproc.open();
    List<int> LotIds = dproc.getLotIds();
    List<Lot> ll = new List<Lot>();
    foreach (int l in LotIds)
    {
        if (l.Contains(name))
            ll.Add(l);
    }
    return ll;
}

public void addProposition(Lot good, double price)
{
    ActiveLots.Add(good);
    Proposition p;
    dproc.open();
    p.Price = price;
    p.Id = dproc.getMaxPropositionId()+1;
    p.Date = DateTime.UtcNow;
    p.Type = 1;
    p.Good = good;
    p.User = new Buyer();
    p.User = this;
    ActivePropositions.Add(p);
    dproc.addProposition(good.Id, price, Ida);
    dproc.close();
}
}

```

```

public class Proposition : Interface
{
    private int id;
    private float price;
    private datetime date;
    private int type;
    private Lot good;
    private int count;
    private Buyer user;

    public int Id
    {
        get
        {
            return this.id;
        }
        set
        {
            this.id = value;
        }
    }

    public float Price
    {
        get
        {
            return this.price;
        }
        set
        {
            this.price = value;
        }
    }

    public datetime Date
    {
        get
        {
            return this.date;
        }
        set
        {
            this.date = value;
        }
    }

    public int type
    {
        get
        {
            return this.type;
        }
        set
    }
}

```

```

        {
            this.type = value;
        }
    }

    public Lot Good
    {
        get
        {
            return this.good;
        }
        set
        {
            this.good = value;
        }
    }

    public int Count
    {
        get
        {
            return this.count;
        }
        set
        {
            this.count = value;
        }
    }

    public Buyer User
    {
        get
        {
            return this.user;
        }
        set
        {
            this.user = value;
        }
    }

    private DatabaseProcessor dproc = new
DatabaseProcessor();

    public Proposition()
    {
        dproc.open();
        Id = dproc.detectMaxAuctionerr() + 1;
        dproc.close();
        Price = 0;
        Date = DateTime.UtcNow;
        Type = 0;
        Buyer = new Buyer();
    }

```

```

        Good = new Lot();
    }

    public Proposition(int q)
    {
        Id = 0;
        Price = 0;
        Date = DateTime.UtcNow;
        Type = 0;
        Buyer = new Buyer();
        Good = new Lot();
    }
}

public class Auction : Interface
{
    private DatabaseProcessor dproc = new
DatabaseProcessor();

    public int getAuctioneerType(string login, string
password)
    {
        dproc.open();
        int t = dproc.getAccountType(login, password);
        dproc.close();
        return t;
    }

    public bool isAuctioneer(string login, string password)
    {
        dproc.open();
        int t = dproc.getId(login, password);
        dproc.close();
        if (t >= 0)
            return true;
        else return false;
    }

    public bool Register(string login, string password)
    {
        dproc.open();
        int t = dproc.getId(login, password);
        if (t >= 0)
        {
            return false;
        }
        else {
            Account acc = new Account();
            acc.Id = dproc.detectMaxAccount() + 1;
            acc.Login = login;
            acc.Password = password;
            acc.Type = 1;
        }
    }
}

```

```

        dproc.close();
    }

    public List<Lot> getAllLots()
    {
        dproc.open();
        List<Lot> ll = new List<Lot>();
        List<int> ln = dproc.getLotIds;
        foreach (int l in ln)
        {
            Lot tl = new Lot();
            tl.Id = l;
            tl.Name = dproc.getLotName(l);
            tl.Info = dproc.getLotInfo(l);
            tl.Price = dproc.getLotPrice(l);
            tl.Count = dproc.getLotCount(l);
            int acc = dproc.getLotAuthor(l);
            tl.Author = new Account();
            tl.Author.Id = acc;
            tl.Type = dproc.getLotType(l);
            tl.Begindate = dproc.getLotBeginDate(l);
            tl.Enddate = dproc.getLotEndDate(l);
            tl.Images = dproc.getLotImages(l);
            ll.Add(tl);
        }
        dproc.close();
        return ll;
    }

    public Lot getLotById(int q)
    {
        dproc.open();
        Lot tl = new Lot();
        tl.Id = q;
        tl.Name = dproc.getLotName(q);
        tl.Info = dproc.getLotInfo(q);
        tl.Price = dproc.getLotPrice(q);
        tl.Count = dproc.getLotCount(q);
        int acc = dproc.getLotAuthor(q);
        tl.Author = new Account();
        tl.Author.Id = acc;
        tl.Type = dproc.getLotType(q);
        tl.Begindate = dproc.getLotBeginDate(q);
        tl.Enddate = dproc.getLotEndDate(q);
        tl.Images = dproc.getLotImages(q);
        dproc.close();
        return tl;
    }

    public List<int> getAuctionResults()
    {
        dproc.open();
        List<int> ln = List<int>();
    }

```

```

ln = dproc.getLotIds();
List<Lot> lres = new List<Lot>();
foreach (int l in ln)
{
    if (dproc.getLotType(l) == 1 &
dproc.getLotEndDate(l) <= DateTime.UtcNow);
    List<int> lp = List<int>();
    lp = dproc.getPropositionIdsForLot(l);
    double maxp = 0;
    int maxid = 0;
    foreach (int p in lp)
    {
        if (dproc.getPropositionPrice(p) > maxp)
        {
            maxp = dproc.getPropositionPrice(p);
            maxid = p;
        }
    }
    Lres.Add(maxid);
}
dproc.close();
return lres;
}

}
}

```

ДОДАТОК В
Слайди презентації

Національний університет «Запорізька політехніка»
кафедра ПЗ

Дипломна кваліфікаційна робота магістра
«Дослідження та програмна реалізація методів
підтримки проведення аукціону»

Виконав
ст. гр. КНТ-119м

В. В. Кузнецов

Керівник
к.т.н., доцент

С. Ю. Скрупський

2020

Рисунок В.1 – Слайд 1

Об'єкт дослідження –
процес проведення аукціону.

Предмет дослідження –
програмне забезпечення підтримки проведення аукціону.

Мета роботи –
розроблення програмного забезпечення підтримки проведення аукціону для забезпечення обґрунтованого прийняття рішень щодо вибору початкових параметрів лотів.

2

Рисунок В.2 – Слайд 2

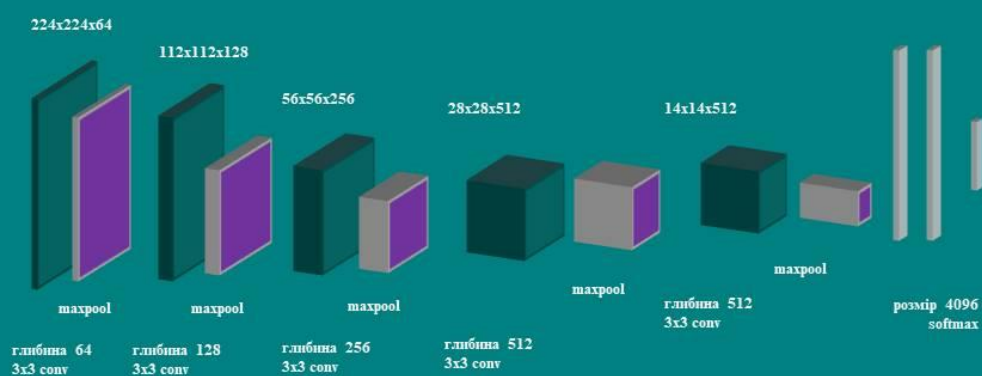
Завдання

- розробка методу підтримки проведення аукціону;
- розробка архітектури програми;
- розробка програми;
- розробка організаційно-економічної частини;
- розробка заходів з охорони праці та цивільної безпеки.

3

Рисунок В.3 – Слайд 3

Архітектура моделі VGG-19



4

Рисунок В.4 – Слайд 4

Метод підтримки проведення аукціону

На першому етапі формується продавцем лот з визначенням параметрів проведення аукціону, що включають дату і час його початку, дату і час його завершення та набір зображень (або окреме зображення), що характеризує лот.

На другому етапі визначається подібність зображення. Для цього використовується модель нейронної мережі на основі архітектури VGG-19 та перцептивний геш.

На третьому етапі визначається початкова і кінцева вартості лоту. Для цього для кожного лота, аукціон якого вже завершено, визначається оцінка подібності на даний лот і вартість визначається усередненням найбільш подібних.

На четвертому етапі продавець узгоджує початкову вартість лоту.

На п'ятому етапі відбувається багатораундовий аукціон, коли протягом часу до завершення аукціону кожен покупець знаходить потрібний йому лот, за бажання встановлює пропозицію щодо купівлі лоту, після чого йому надається набір подібних лотів

На останньому етапі відбувається підведення підсумків аукціону. Визначається переможець або переможці аукціону.

5

Рисунок В.5 – Слайд 5

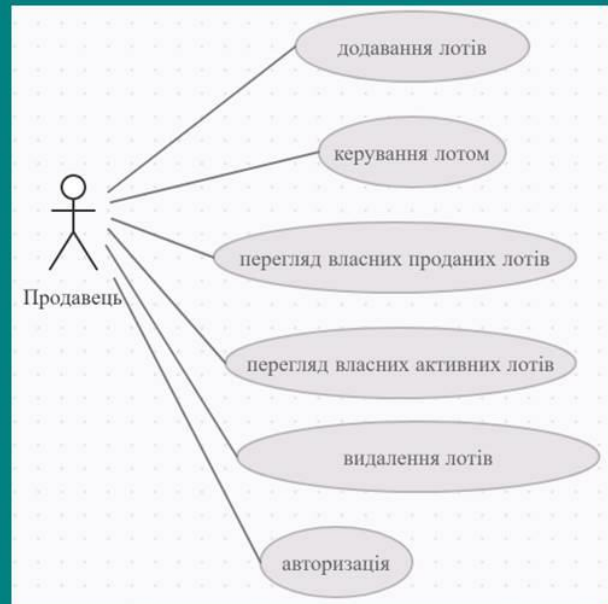
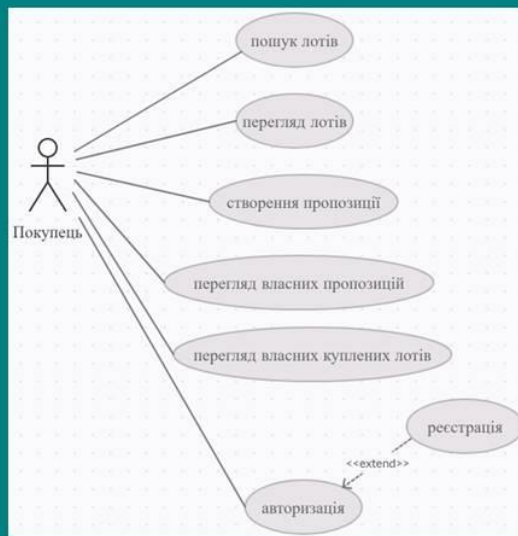
Вибір мови програмування

Показник	Java	C#
Парадигма	На основі класів, об'єктно-орієнтована мова, похідна від C++	Об'єктно-орієнтована, компонентно-орієнтована, функціональна
Збирач сміття	Є	Є
Область застосування	Вебзастосунки	Веброзробка, розробка ігор, розробка настільних застосунків
Розробка застосунків для Windows	Можлива	Основний напрямок

6

Рисунок В.6 – Слайд 6

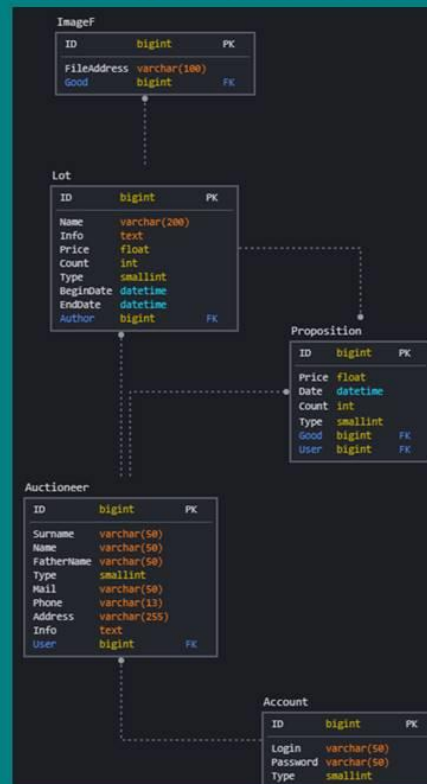
Моделювання функціональних вимог до програмного забезпечення



7

Рисунок В.7 – Слайд 7

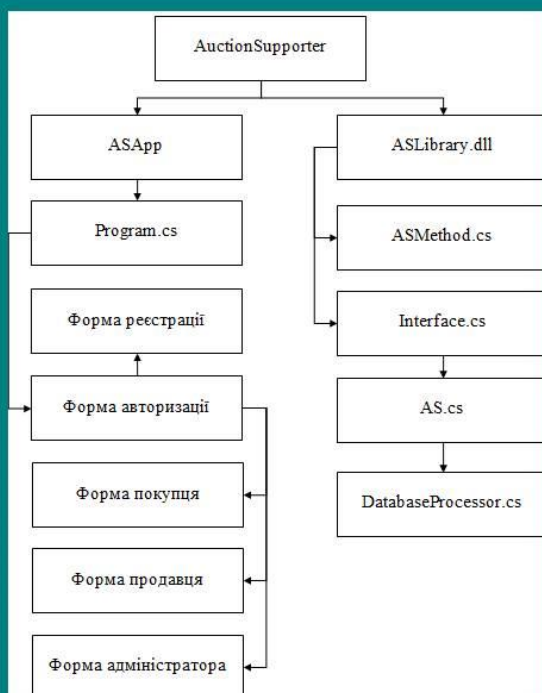
Структура бази даних



8

Рисунок В.8 – Слайд 8

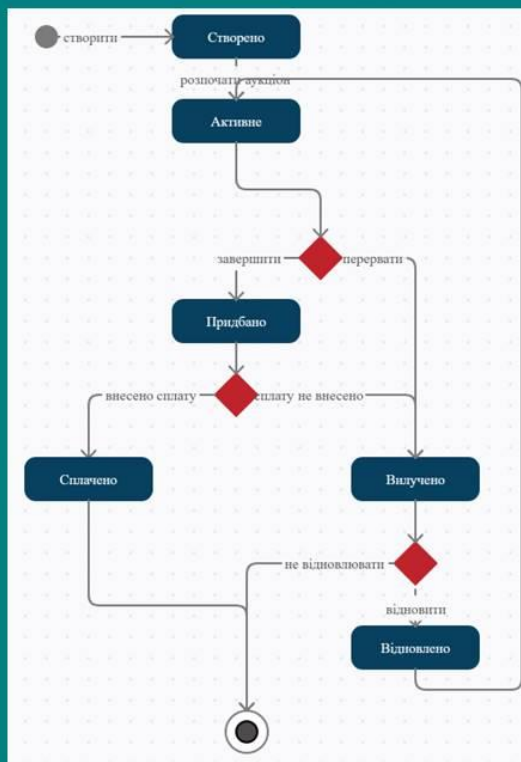
Структурна схема програмного застосунку



9

Рисунок В.9 – Слайд 9

Діаграма станів лоту аукціону



10

Рисунок В.10 – Слайд 10

Перегляд власних лотів продавцем

Аукціон: Ви увійшли як продавець

Додавання лоту

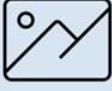
Назва лоту:

Кількість: 0 Вартість:

Початок: Очікувана:

Завершення: Пропонована:

Встановити:

Опис: 

Обрати Вилучити Активувати

Керування лотом

Лот номер: 20201118200702 Вилучити

Обраний лот

Лот номер: 20201118200702 До завершення: 11.00.0000 02:16 Вартість: 2000

Опис: 

Обрати Вилучити Зберегти

Активні лоти

ID	Назва	Кількість	Вартість
20201119141501	Чашка ручної роботи	1	300
20201119203047	Чашка ручної роботи	1	200
20201118201032	Кружка ручної роботи	1	700
20201118200702	Набір тарілок ручної роботи	1	2800

Продані лоти

ID	Назва	Кількість	Вартість
20201118...	Глечик на добрі справи	1	2700

11

Рисунок В.11 – Слайд 11

Додавання лоту та автоматичне визначення параметрів лоту

Аукціон: Ви увійшли як продавець

Додавання лоту

Назва лоту: Чашка ручної роботи

Кількість: 1 Вартість:

Початок: 20.11.2020 10:00 Очікувана: 500

Завершення: 30.11.2020 10:00 Пропонована: 300

Встановити: 300

Опис: 

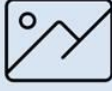
Обрати Вилучити Активувати

Керування лотом

Лот номер: Вилучити

Обраний лот

Лот номер: До завершення: Вартість:

Опис: 

Обрати Вилучити Зберегти

Активні лоти

ID	Назва	Кількість	Вартість
20201119141501	Чашка ручної роботи	1	300
20201119203047	Чашка ручної роботи	1	200
20201118201032	Кружка ручної роботи	1	700
20201118200702	Набір тарілок ручної роботи	1	2800

Продані лоти

ID	Назва	Кількість	Вартість
20201118...	Глечик на добрі справи	1	2700

12

Рисунок В.12 – Слайд 12

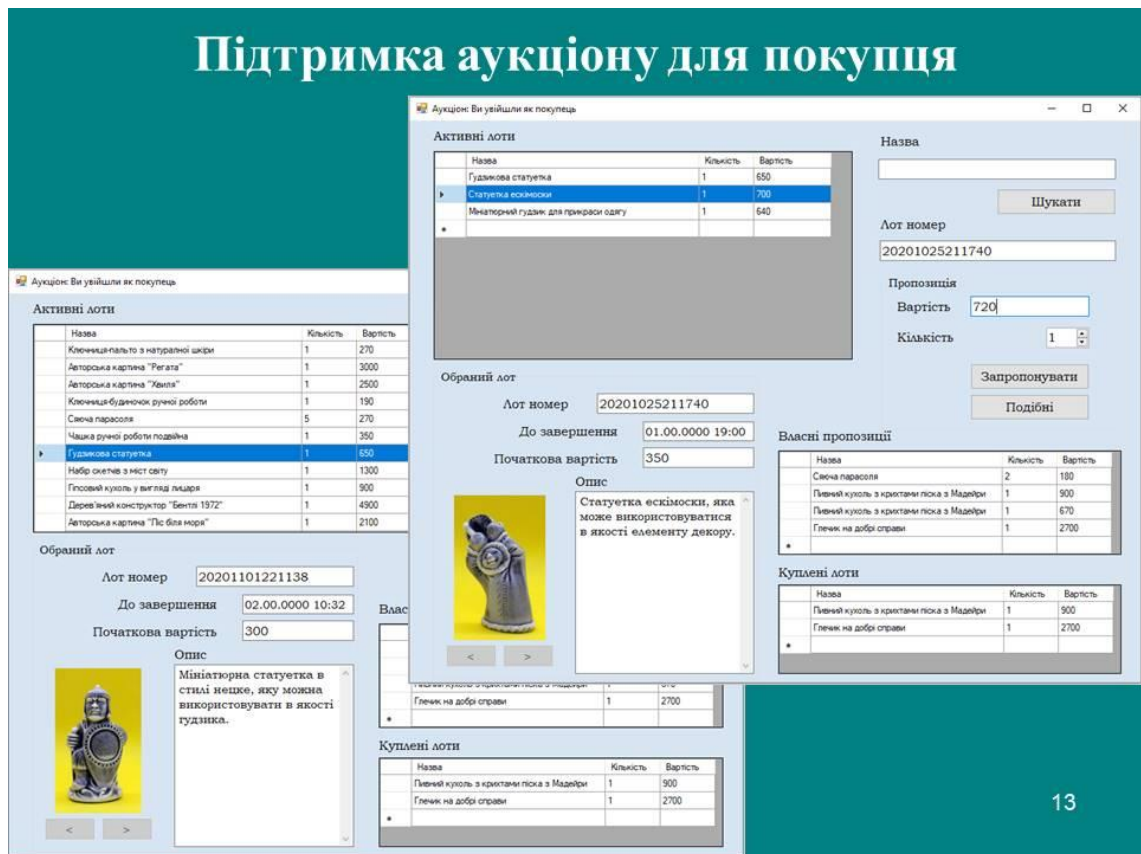


Рисунок В.13 – Слайд 13

Висновки

Виконано реалізацію програмного забезпечення підтримки проведення аукціону. Описано алгоритм функціонування програмного забезпечення з приведенням діаграми станів для лоту аукціону, реалізовано модулі та класи програмного забезпечення, описано основні розроблені класи з приведенням відповідних методів та описом їх роботи. Описано процес роботи користувача з програмою.

Усі поставлені завдання на роботу виконано.

Наукова новизна роботи полягає в запропонованому методі підтримки проведення аукціону, проведення якого ґрунтується на моделі відкритого багатораундового аукціону, який використовує згорткову нейронну мережу на основі моделі архітектури VGG-19 для виділення подібних зображень лотів, що дозволяє узгоджувати початкову вартість лоту з подібними раніше проданими лотами, отримувати оцінку фактичної вартості лоту продавцям та знаходити подібні лоти для забезпечення синхронності аукціону для покупців.

Рисунок В.14 – Слайд 14