

System C2C a zmniejszenie konsumpcjonizmu w społeczeństwie

PROJEKT DYPLOMOWY

DANE PARTNERÓW

A1. Dane Promotora

Imię i nazwisko	Mariusz Nogala
Stopień / Tytuł naukowy	Doktor
Data i podpis	

A2. Dane członków Zespołu projektu

Imię i nazwisko	Iwona Ogrodowska
Kierunek studiów	Informatyka
Tryb studiów	Hybrydowy
Nr albumu	89281

Imię i nazwisko	Łukasz Bielski
Kierunek studiów	Informatyka
Tryb studiów	Hybrydowy
Nr albumu	130485

Link do kodu źródłowego projektu

<https://github.com/anowiogr/C2C.git>

ZAŁOŻENIA PROJEKTU

B1. Opis projektu

1. Uzasadnienie wyboru tematu

Wzrost usług i dóbr sprawia, że konsumenci kupują więcej niż w rzeczywistości potrzebują. Przyczyn takiego zjawiska może być wiele. Można wyróżnić, np. postrzeganie posiadania wartościowych przedmiotów w kontekście podniesienia swojej pozycji w społeczeństwie; presję wywieraną przez społeczeństwo, w efekcie której konsumenci kupują więcej dóbr, które nie są tak naprawdę niezbędne; marketing, który przyczynia się do chęci nabywania nowszych produktów i wreszcie same firmy, które produkują tak, aby po jakimś czasie zaszła konieczność zakupu np. nowego sprzętu, którego starszej wersji system nie będzie już aktualizowany. Nadmiar konsumpcji ma niestety swoje negatywne skutki. Oprócz nadmiernego wydawania pieniędzy, przyczynia się także do degradacji środowiska, ponieważ produkcja skorelowana jest z transportem towarów i emisją gazów cieplarnianych; zużyciem surowców naturalnych; wycinką lasów, aż po generowanie odpadów. Problem generowania odpadów jest na tym zauważalny, że obecnie nacisk stawia się na gospodarkę w obiegu zamkniętym. Obieg zamknięty to nie tylko kwestia producentów dóbr i usług, którym zadaniem np. projektowanie produktów tak, aby łatwo było je naprawić lub ponownie wykorzystać. Samym celem gospodarki w obiegu zamkniętym jest koncentracja na wykorzystaniu zasobów i minimalizowaniu odpadów. Gospodarka w obiegu zamkniętym stawia więc na ponowne wykorzystanie, recykling i odnawianie zasobów. Gospodarka w obiegu zamkniętym to także trend wśród samych konsumentów, którzy czują potrzebę dać „drugie życie” przedmiotom. W odpowiedzi na ten trend wyróżnia się slow fashion, minimalizm i odpowiedzialną konsumpcję. Jak podaje Jarzębska (2019)¹, już od 1970 roku została przekroczona zdolność regeneracyjna środowiska i społeczeństwo zużywa więcej zasobów. Autorka przytacza także, że globalny problem plastiku w 1950 roku wynosił 1,5 mln ton rocznie, a w 2016 roku wzrósł do 335 mln ton rocznie, z czego 19% dotyczy samej Europy.

Promowanie trendu tzw. slow fashion, który obejmuje świadome, zrównoważone zakupy, przyczynia się również do wprowadzania na rynek platform sprzedażowych

¹ Jarzębska, E. (2019). Konsument w gospodarce o obiegu zamkniętym. *Studia i Prace Kolegium Zarządzania i Finansów*, (172), 53-69.

opartych na modelu sprzedaży C2C (*Consumer to Consumer*)². Jest to rodzaj sprzedaży, w której jedna osoba sprzedaje produkt lub usługę drugiej osobie, najczęściej za pośrednictwem platform internetowych, które umożliwiają wystawianie ogłoszeń i komunikację pomiędzy użytkownikami. Przykładem C2C popularnego wśród Polaków jest sprzedaż używanych przedmiotów na platformach takich jak OLX, czy Vinted. Tworzenie nowych, konkurencyjnych platform opartych na modelu sprzedaży C2C uznaje się za zasadne biorąc pod uwagę takie aspekty jak: niedostateczna liczba ofert na istniejących platformach (choć OLX i Vinted są bardzo popularne, to nie zaspakają wszystkich potrzeb sprzedażowych skupiających się na konkretnych kategoriach produktów, np. sprzedaż specjalistycznego sprzętu); promowanie świadomości ekologicznej i zrównoważonego rozwoju (rosnąca świadomość konsumentów na temat marnotrawienia zasobów i promowanie potrzeb dbałości o środowisko); dostosowanie do nowych trendów (nowe platformy mogą zaoferować innowacyjne funkcje, które nie są obecne na istniejących serwisach); lokalizacja i personalizacja (ukierunkowanie nowych platform na określone regiony, szczególnie wśród mniejszych społeczności); zwiększone bezpieczeństwo transakcji (zwłaszcza transakcji zawieranych z seniorami, którzy mogą być nieświadomi, że są ofiarą oszustwa); zwiększający się rynek wynajmu i współdzielenia (wynajem narzędzi, sprzętu sportowego, aż po samochody i mieszkania), nowe platformy mogą ułatwiać te procesy, a nawet specjalizować się w konkretnych usługach współdzielenia.

Mając na uwadze powyższe spostrzeżenia nasze własne jak i na podstawie literatury, założenia naszego projektu obejmują ocenę wpływu systemu C2C na zmniejszenie konsumpcjonizmu w społeczeństwie w kontekście zaproponowania funkcjonalnego rozwiązania dla platformy opartej na modelu sprzedaży C2C.

2. Problem badawczy

Główny Problem badawczy

W jaki sposób platformy sprzedażowe oparte na modelu C2C mogą przyczynić się do zmniejszenia konsumpcjonizmu w społeczeństwie, promując bardziej zrównoważoną i odpowiedzialną konsumpcję?

² Kos-Labędowicz, J. (2015). Aukcje internetowe jako źródło informacji o produktach. *Studia Ekonomiczne*, 232, 103-112.

Pytania pomocnicze

- Jakie mechanizmy kierują użytkownikami platform sprzedażowych opartych na modelu sprzedaży C2C?
- Jakie funkcjonalności powinna oferować platforma oparta na modelu sprzedaży C2C?

3. Cel główny i cele szczegółowe projektu

Cel główny

Opracowanie koncepcji funkcjonalności dla platformy sprzedażowej opartej na modelu C2C, która przyczyni się do zmniejszenia konsumpcjonizmu w społeczeństwie poprzez promowanie zrównoważonego, odpowiedzialnego podejścia do konsumpcji i ponownego wykorzystywania zasobów.

Cele szczegółowe

- **Analiza roli systemu C2C w przeciwdziałaniu nadmiernemu konsumpcjonizmowi.**

Wpływ platform C2C na postawy konsumentów oraz ich podejście do kupowania nowych produktów w porównaniu do wykorzystywania przedmiotów używanych lub wymiany.

- **Określenie cech, które powinna posiadać platforma oparta na modelu sprzedaży C2C, aby skutecznie promować zrównoważoną konsumpcję.**

Zidentyfikowanie funkcji, które mogą wspierać ideę odpowiedzialnej konsumpcji.

- **Projektowanie i zaprezentowanie koncepcji funkcjonalności platformy opartej na modelu sprzedaży C2C, odpowiadającej na potrzeby zrównoważonego rozwoju.**

Opracowanie funkcjonalności dla platformy sprzedażowej, uwzględniającej aspekty technologiczne, społeczno-ekologiczne oraz użytkowe.

- **Promocja i wdrożenie innowacyjnych rozwiązań.** Opracowanie strategii wdrożenia i popularyzacji.

4. Zakres podmiotowy, przedmiotowy, czasowy i przestrzenny

- **Zakres podmiotowy**

Zakres podmiotowy projektu obejmuje uczestników rynku sprzedaży C2C, czyli zarówno konsumentów, którzy kupują i sprzedają używane przedmioty, jak i platformy internetowe

umożliwiające takie transakcje. Główną grupą docelową w projekcie są studenci, ale projekt może ewoluować, ponadto projekt może włączać organizacje pozarządowe i instytucje zajmujące się promowaniem zrównoważonego rozwoju oraz przeciwdziałaniem nadmiernemu konsumpcjonizmowi, takie jak fundacje działające na rzecz ochrony planety.

- **Zakres przedmiotowy**

Zakres przedmiotowy dotyczy funkcjonalności platformy C2C, która ma na celu promowanie zrównoważonego i odpowiedzialnego podejścia do konsumpcji. Kluczowe zagadnienia to:

- Analiza wpływu systemu C2C na postawy konsumentów oraz zmniejszenie konsumpcjonizmu.
- Określenie funkcji platformy, które platforma powinna posiadać, aby wspierać zrównoważoną konsumpcję.

- **Zakres czasowy**

Realizację projektu podzielono na następujące fazy:

Faza badawcza (1-2 miesiące)

Analiza rynku C2C, analiza postaw konsumentów wobec zrównoważonego rozwoju, identyfikacja potrzeb funkcjonalnych platformy.

Faza projektowa (4-5 miesięcy)

Zaprojektowanie funkcjonalności platformy, uwzględniając ocenę rynku.

Faza wdrożeniowa (5-6 miesięcy)

Implementacja zaprojektowanych funkcji oraz testowanie technologii.

- **Zakres przestrzenny**

Projekt dotyczy platformy sprzedażowej, która ma funkcjonować na poziomie krajowym. Działalność platformy obejmuje przestrzeń cyfrową (platforma internetowa), a projekt koncentruje się na integracji z istniejącymi infrastrukturami logistycznymi, ekologicznymi i społecznymi, które wspierają zrównoważone podejście do konsumpcji wśród studentów.

5. Metody i techniki badawcze

- **Analiza literatury**

Analiza literatury fachowej, raportów, badań i artykułów naukowych dotyczących funkcjonalności systemu C2C w kontekście zmniejszenia konsumpcjonizmu w społeczeństwie.

- **Metoda projektowania funkcjonalności**

Stworzenie funkcjonalności jako elementu popularyzowania i promowania korzystania z platform opartych na modelu sprzedażowym C2C

- **Symulacje i testy użytkowe**

Przeprowadzenie próbnych operacji z wykorzystaniem prototypu.

Dobór powyższych metod i technik pozwoli na zbadanie problemu oraz weryfikację przydatności zaproponowanego rozwiązania.

B2. Zadania w projekcie

Cele szczegółowe projektu	Zadania w projekcie oraz termin rozpoczęcia i zakończenia realizacji zadania	Osoby zaangażowane w realizację zadania
Cel 1: Analiza roli systemu C2C w przeciwdziałaniu nadmiernemu konsumpcjonizmowi.	Zadanie 1: Przegląd literatury i wybór najnowszych doniesień literaturowych w Polsce i na świecie. (01.01.2024-01.02.2024)	1. Łukasz Bielski 2. Iwona Ogrodowska/Łukasz Bielski
	Zadanie 2: Określenie na podstawie literatury wpływu platform C2C na postawy konsumentów oraz ich podejście do kupowania nowych produktów w porównaniu do wykorzystywania przedmiotów używanych lub wymiany. (01.02.2024-01.03.2024)	
Cel 2: Określenie cech, które powinna posiadać platforma oparta na modelu sprzedaży C2C, aby skutecznie promować	Zadanie 1: Weryfikacja dostępnych platform internetowych opartych na systemie sprzedażowym C2C. (01.03.2024-01.05.2024)	1. Iwona Ogrodowska 2. Iwona Ogrodowska 3. Łukasz Bielski

zrównoważoną konsumpcję.	Zadanie 2: Zidentyfikowanie funkcji, które mogą wspierać ideę odpowiedzialnej konsumpcji (01.03.2024-01.06.2024)	
	Zadanie 3: Przygotowanie wstępnego szkicu bazy danych. (01.03.2024-01.06.2024)	
Cel 3: Projektowanie i zaprezentowanie koncepcji funkcjonalności platformy opartej na modelu sprzedaży C2C, odpowiadającej na potrzeby zrównoważonego rozwoju	Zadanie 1: Opracowanie funkcjonalności dla platformy sprzedażowej, uwzględniającej aspekty technologiczne, społeczno-ekologiczne oraz użytkowe. (01.09.2024-01.10.2024)	1. Iwona Ogrodowska 2. Łukasz Bielski 3. Iwona Ogrodowska 4. Łukasz Bielski
	Zadanie 2: Przygotowanie bazy danych. (01.09.2024-01.11.2024)	
	Zadanie 3: Scalenie wykonanych zadań 1 i 2. (01.11.2024-31.12.2024)	
	Zadanie 4. Weryfikacja funkcjonalności (01.01.2025-30.01.2025)	
Cel 4: Promocja i wdrożenie innowacyjnych rozwiązań	Zadanie 1: Opracowanie strategii wdrożenia i popularyzacji założeń projektu. (01.01.2025-30.01.2025)	1. Iwona Ogrodowska 2. Łukasz Bielski
	Zadanie 2: Opracowanie przewidywań rozwoju projektu i sformułowanie wniosków. (01.02.2025-07.02.2025)	

C1. Opracowanie projektu

1. Założenia teoretyczne

1.1. Przegląd literatury

1.1.1. Model C2C – definicje, mechanizmy i potencjał rynkowy

W dobie dynamicznego rozwoju technologii cyfrowych oraz rosnącej popularności zakupów online, platformy sprzedażowe stały się nieodłącznym elementem współczesnego rynku. Coraz częściej obserwuje się, że nie tylko oferują one możliwość bezpośredniej wymiany dóbr, ale także kształtują postawy oraz zachowania konsumentów. W kontekście rosnących obaw związanych z konsumpcjonizmem, marnotrawstwem zasobów oraz degradacją środowiska, funkcjonalności platform sprzedażowych mogą pełnić rolę narzędzia wspierającego transformację nawyków zakupowych.

Model sprzedażowy C2C, to skrót *Consumer to Consumer* co w tłumaczeniu z języka angielskiego oznacza „konsument do konsumenta”. Są to swoistego rodzaju interakcje biznesowe zawierane między konsumentami z wykorzystaniem środków elektronicznej komunikacji. Model sprzedażowy C2C to portale z ogłoszeniami, wirtualne „domy aukcyjne”, np. Allegro, eBay. Aukcje online, określane także jako aukcje elektroniczne, odbywają się na portalach internetowych, gdzie sprzedawcy prezentują swoje oferty dotyczące towarów lub usług. Takie platformy, wyposażone w odpowiednie narzędzia, umożliwiają przeprowadzenie całego procesu sprzedażowego – od momentu uruchomienia aukcji i ustalenia ceny wyjściowej, przez zbieranie ofert od zainteresowanych nabywców, aż do zakończenia licytacji i wyłonienia kupującego lub kupujących, którzy zaoferowali najwyższą cenę. Aukcje internetowe nie podlegają ograniczeniom czasowym – zarówno oferty sprzedaży, jak i zakupu mogą być składane o dowolnej porze, kiedy system aukcyjny jest aktywny. Ta swoboda jest jednym z kluczowych atutów takich form transakcji. Sprzedający może zdecydować, czy dany przedmiot będzie dostępny przez określony czas, czy też do momentu osiągnięcia ustalonej ceny, natomiast kupujący mają możliwość łatwego wyszukiwania, porównywania i licytowania interesujących ich ofert (Kawecki 2012).³

³ Kawecki, T. (2012). Handel C2C-czy to tylko aukcje? E-mentor, 44(2), 61-64.

Jak podaje Rilfi (2021)⁴, model biznesowy C2C może eliminować potrzebę korzystania z pośredników między sprzedającymi a kupującymi. W mediach społecznościowych ludzie mogą udostępniać informacje związane z ich chęcią zakupu, sprzedaży lub korzystania z produktów i usług, co pełni rolę rynku dla handlu elektronicznego C2C. Na polskim rynku e-commerce funkcjonuje wiele platform aukcyjnych, które od siebie różnią się m.in. rodzajem organizowanych aukcji oraz wymogami stawianymi użytkownikom, jednak ich główne zasady funkcjonowania pozostają zbliżone. Największym i najstarszym graczem w tej branży jest Allegro, którego dominacja na rynku jest zauważalna od lat, mimo że na polskim rynku pojawiają się także zagraniczne serwisy aukcyjne.⁵

1.1.2. Konsumpcjonizm i zrównoważony rozwój – analiza zjawiska i jego konsekwencje

Konsumpcjonizm, rozumiany jako dominujący model społeczno-ekonomiczny oparty na intensywnym nabywaniu dóbr materialnych. Od dziesięcioleci kształtuje współczesne społeczeństwa. Jego dynamiczny rozwój napędzany jest przez postęp technologiczny, globalizację oraz wszechobecną reklamę, co przekłada się na wzrost gospodarczy i podniesienie poziomu życia. Jednakże rosnąca konsumpcja niesie ze sobą również poważne wyzwania, zwłaszcza w kontekście zrównoważonego rozwoju – idei zakładającej harmonijne połączenie aspektów ekonomicznych, społecznych i środowiskowych. Pangsy-Kania (2024)⁶, zwraca uwagę, że konsumpcjonizm, czyli nadmierna, niekontrolowana konsumpcja, prowadzi do kryzysu wartości osobistych oraz kryzysów gospodarczych. Problem konsumpcjonizmu jest jednym z kluczowych problemów współczesnego świata w wielu aspektach i, poprzez degradację systemu wartości, stanowi ogromne zagrożenie dla przyszłości ludzkości. Konsumpcja ma kluczowy wpływ na poziom i jakość życia społeczeństw, a jednocześnie może pełnić funkcję socjalizacyjną, umożliwiając przyswajanie norm i zasad związanych z odpowiedzialnym korzystaniem z dóbr i usług, przy jednoczesnym dbaniu o środowisko.

Współczesny konsument żyje w otoczeniu, w którym dostępnych jest mnóstwo różnorodnych produktów i usług. Decyzje zakupowe są często wynikiem oddziaływania ze strony producentów, sprzedawców oraz dostawców usług, którzy wykorzystują narzędzia

⁴ Rilfi, M. R. M. (2021). Product attribute extraction from C2C social media messages.

⁵ Sikora, M., Dzieża, G., & Lwowski, Z. (2011). Ograniczenia w realizacji transakcji internetowych. *Studia i Materiały Polskiego Stowarzyszenia Zarządzania Wiedzą*, 49, 157-171.

⁶ Pangsy-Kania, S. (2024). Czy za pieniądze można kupić szczęście? – Konsumpcjonizm jako przyczyna kryzysów. *Optimum. Economic Studies*, 116(2), 20-42.

marketingu oraz socjotechniki, a także oferują wygodne opcje finansowania, takie jak zakupy na raty. Przykładem negatywnych skutków konsumpcjonizmu jest marnotrawstwo żywności i produktów przemysłowych. Komisja Europejska prognozuje, że ilość wyrzucanej żywności będzie rosła wraz ze wzrostem populacji, rosnącym zapotrzebowaniem na żywność oraz wzrostem poziomu dobrobytu (Komisja Europejska 2010). Podobne czynniki przyczynią się także do zwiększenia ilości odpadów pochodzących z produktów przemysłowych, odzieży, obuwia oraz innych dóbr codziennego użytku.⁷

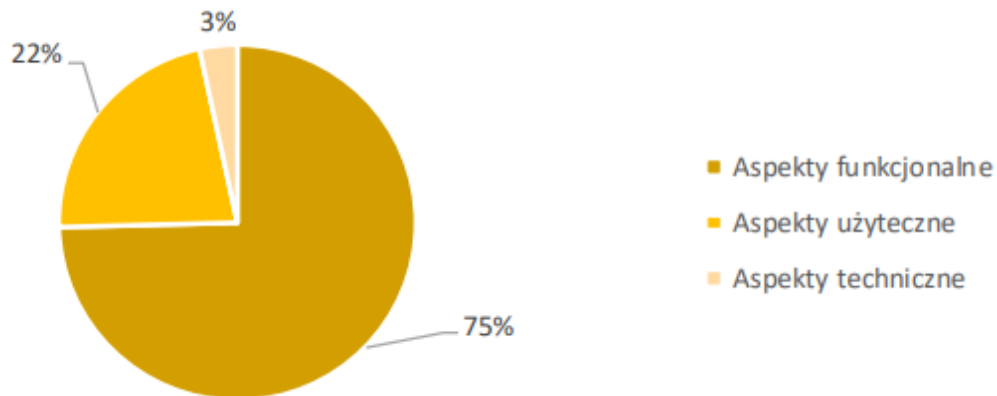
1.1.3. Funkcjonalności platform sprzedażowych jako narzędzie zmiany zachowań konsumentów

W dobie, gdy coraz więcej osób korzysta z Internetu, aby kupować i sprzedawać różnorodne przedmioty, platformy oparte na modelu sprzedażowym C2C, takie jak OLX i Vinted zyskały ogromną popularność. Oba serwisy umożliwiają handel używanymi rzeczami, jednak pojawia się pytanie: który z nich jest lepszy? Odpowiedź nie będzie jednoznaczna, ponieważ „to zależy” jaki efekt sprzedający i/lub kupujący chce osiągnąć. Należy więc skupić się na podstawowych funkcjonalnościach serwisów opartych na modelu sprzedaży C2C, np. kategoryzacja produktów, bezpieczeństwo transakcji, możliwość negocjacji ceny bez wysyłania wiadomości do sprzedającego. Obie platformy umożliwiają handel w wielu kategoriach, jednak ich asortymenty różnią się pod względem zakresu. OLX oferuje szeroką gamę kategorii – od elektroniki, poprzez nieruchomości, aż po samochody. To wszystko sprawia, że jest to uniwersalne narzędzie do sprzedaży i zakupu różnorodnych produktów. Vinted natomiast koncentruje się głównie na modzie i odzieży, choć w ofercie można znaleźć także kosmetyki oraz akcesoria, a nawet elektronikę. Zarówno OLX, jak i Vinted oferują mechanizmy umożliwiające ocenę oraz komentowanie użytkowników, a także opcje płatności realizowanych za pośrednictwem platformy. OLX często kojarzy się z możliwością negocjacji cenowych. Użytkownicy mogą bezpośrednio rozmawiać ze sprzedającymi, starając się uzyskać korzystniejsze warunki transakcji. Z kolei Vinted działa w oparciu o negocjację ceny z użyciem funkcji „zaproponuj cenę”, co nie wymusza bezpośredniej wymiany wiadomości ze sprzedającym.

⁷ Dąbrowska, A. (2015). Postawy polskich konsumentów – od konsumpcjonizmu do zrównoważonej konsumpcji. *Handel Wewnętrzny*, 355(2), 88-100.

Jak podają Cebulska i Pieczarka (2024)⁸, w miarę jak technologia cyfrowa rozwija się w zawrotnym tempie, a zakupy online stają się coraz bardziej powszechne, platformy sprzedażowe zyskują na znaczeniu w nowoczesnym handlu. Obecnie konsumenci mają do wyboru wiele stron internetowych – od światowych liderów rynku po mniejsze, specjalistyczne sklepy, co czyni analizę czynników wpływających na ich decyzje zakupowe szczególnie ważną. Celem autorek było zbadanie i ocena wpływu aspektów technicznych, funkcjonalnych oraz użytkowych platform sprzedażowych na wybór miejsca dokonywania zakupów online. W pracy Cebulska i Pieczarka (2024) podjęto próbę odpowiedzi na pytanie badawcze: W jaki sposób te trzy kategorie czynników determinują decyzje konsumentów przy wyborze konkretnej platformy do zakupów w Internecie? Ponadto zostały przytoczone badania ankietowe, które sugerują, że 95,3% ankietowanych kobiet i mężczyzn dokonuje zakupów na platformach sprzedażowych, a częstotliwość tych zakupów jest następująca: kilka razy w miesiącu (55,7%), kilka razy w roku (26,2%), kilka razy w tygodniu (16,4%), codziennie: (1,7%). Autorzy dokonali również oceny aspektów wpływających na decyzję wyboru platformy zakupowej (rys. 1). Do aspektów funkcjonalnych w tym artykule brano pod uwagę: dostępność do różnorodnych metod płatności, metody wysyłki i dostawy zamówienia, opcje zarządzania kontem. Do aspektów użytkowych natomiast wzięto pod uwagę: prostotę procesu zakupowego, intuicyjność interfejsu, możliwość dostosowania doświadczeń zakupowych do indywidualnych preferencji użytkownika. Z kolei spośród aspektów technicznych: bezpieczeństwo transakcji, szybkość z jaką strona się ładuje i reaguje na działania użytkownika, responsywność (dostosowanie strony do różnych urządzeń i rozdzielczości ekranu). Autorzy (Cebulska i Pieczarka, 2024) wskazują, że respondenci najbardziej cenią te elementy platformy, które bezpośrednio kształtują ich codzienne doświadczenia zakupowe i skutecznie odpowiadają na ich praktyczne potrzeby. W związku z tym można przypuszczać, że główne platformy już osiągnęły wysoki poziom techniczny, co sprawia, że to inne kryteria stają się kluczowymi wyróżnikami decydującymi o ostatecznym wyborze.

⁸ Cebulska, A., Pieczarka, K. (2024). Znaczenie aspektów technicznych, funkcjonalnych i użytkowych platform sprzedażowych. Wyższej Szkoły Zarządzania i Przedsiębiorczości z siedzibą w Wałbrzychu, 5.



Rys. 1. Ocena aspektów wpływający na decyzje wyboru platformy zakupowej wśród Polaków (Cebulska i Pieczarka 2024)⁶.

1.2. Wymagania funkcjonalne i нефункционалне projektu pracy dyplomowej

1.2.1. Wymagania funkcjonalne

- **Rejestracja i logowanie użytkowników**
 - Użytkownik ma możliwość rejestracji przy użyciu adresu e-mail, numeru telefonu lub kont mediów społecznościowych.
 - System przeprowadza weryfikację tożsamości (np. poprzez wysłanie linku aktywacyjnego) w celu potwierdzenia rejestracji.
 - Logowanie odbywa się w bezpieczny sposób, z mechanizmem umożliwiającym odzyskiwanie zapomnianego hasła.
- **Tworzenie i zarządzanie profilem użytkownika**
 - Użytkownik może uzupełniać i modyfikować dane swojego profilu, w tym dane osobowe, kontaktowe oraz adresowe.
 - System umożliwia przypisanie użytkownikowi określonego typu (np. administrator, zwykły użytkownik), co determinuje zakres uprawnień.
 - W ramach weryfikacji konta przewidziano mechanizm, w którym weryfikację (np. potwierdzenie autentyczności) wykonuje inny użytkownik.
- **Publikowanie ofert sprzedaży (aukcji)**
 - Użytkownik może tworzyć oferty sprzedaży, podając tytuł, opis, dodając zdjęcia oraz przypisując ofertę do odpowiedniej kategorii.
 - Oferta zawiera informacje o stanie produktu, cenie, walucie oraz ewentualnych dodatkowych cechach (np. oznaczenie, czy produkt jest nowy lub używany).

- Weryfikacja aukcji, czyli potwierdzenie jej wiarygodności odbywa się przez innego użytkownika.
- **Wyszukiwanie i filtrowanie ofert**
 - System umożliwia użytkownikom wyszukiwanie aukcji na podstawie słów kluczowych.
 - Użytkownicy mogą filtrować oferty według kryteriów takich jak kategoria, cena, stan produktu, lokalizacja oraz inne parametry wpływające na wybór oferty.
 - Interfejs wyszukiwania zapewnia intuicyjną nawigację i łatwy dostęp do interesujących użytkownika ofert.
- **System transakcji**
 - Platforma umożliwia użytkownikom inicjowanie i przeprowadzanie transakcji w oparciu o oferty sprzedaży.
 - System przewiduje mechanizmy negocjacji cen oraz rejestrację pełnej historii transakcji.
- **System komunikacji**
 - Wbudowany system wiadomości pozwala użytkownikom komunikować się między sobą przed, w trakcie oraz po zakończeniu transakcji.
 - Użytkownicy otrzymują powiadomienia o nowych wiadomościach oraz zmianach statusu prowadzonych transakcji.

1.2.2. Wymagania niefunkcjonalne

- **Niezawodność i dostępność**
 - System musi być dostępny przez całą dobę (24/7) z minimalnym czasem przestoju.
 - Wdrożenie mechanizmów backupu oraz odzyskiwania danych w razie awarii, aby zapewnić ciągłość działania platformy.
- **Użyteczność (usability)**
 - Interfejs użytkownika powinien być intuicyjny, estetyczny i łatwy w nawigacji.
 - System, wykorzystujący Bootstrap, ma zapewnić spójny wygląd i łatwą obsługę na różnych urządzeniach.
- **Responsywność**
 - Aplikacja musi być zoptymalizowana pod kątem działania na różnych urządzeniach: desktop, tablet, smartfon.

- Interfejs powinien dostosowywać się do różnych rozdzielczości ekranu, zapewniając komfort korzystania z platformy.
- **Zgodność z przepisami**
 - System musi spełniać obowiązujące przepisy dotyczące ochrony danych osobowych (np. RODO) oraz regulacje związane z handlem elektronicznym i ochroną konsumentów.
 - Regularne audyty i aktualizacje systemu są wymagane, aby utrzymać zgodność z normami prawnymi.
- **Efektywność energetyczna**
 - Kod aplikacji i infrastruktura serwerowa powinny być zoptymalizowane pod kątem zużycia energii, wspierając ideę zrównoważonego rozwoju.
 - Wdrożenie energooszczędnych technologii oraz rozwiązań chmurowych jest pożądane.
- **Dostępność i konserwacja**
 - System musi być łatwy w utrzymaniu, z przejrzystą dokumentacją techniczną i użytkownika.
 - Regularne aktualizacje, szybka reakcja na zgłoszenia błędów oraz jasny proces konserwacji zapewnią długoterminową stabilność platformy.
- **Lokalizacja i internacjonalizacja**
 - Platforma powinna umożliwiać wdrożenie wersji wielojęzycznych oraz łatwe dostosowanie treści do specyfiki lokalnych rynków.
 - System powinien wspierać konfigurację ustawień regionalnych, aby spełniać wymagania użytkowników z różnych krajów.

2. Opis sytuacji faktycznej

Jak podają Leloch i Kowalska (2023)⁹, 77% internautów dokonuje zakupów online, a wśród nich aż 75% wybiera polskie sklepy internetowe. Największy odsetek klientów online to osoby w przedziale wiekowym 35–49 lat (33%), natomiast użytkownicy powyżej 50. roku życia stanowią 30% kupujących. Grupy wiekowe 25–34 lata oraz 15–24 lata reprezentują kolejno 20% i 17% zakupujących w sieci. Można zatem stwierdzić, że e-commerce cieszy się popularnością nie tylko wśród młodszych pokoleń i przeważnie są

⁹ Leloch, N., & Kowalska, M. Zrównoważony rozwój e-commerce w dobie przemian społeczno-gospodarczych. Wybrane Problemy Logistyki I Usług Turystycznych Analiza Sektorowa, Zrównoważony Rozwój, Rachunkowość, 101.

to mieszkańcy dużych miast, osoby z wyższym wykształceniem, które oceniają swoją sytuację finansową jako przeciętną.

Wśród Polaków, według źródeł internetowych (np. brandly360.com) do najpopularniejszych serwisów opartych na modelu sprzedażowym C2C należą: Allegro, OLX, Vinted, Facebook Marketplace, Amazon, AliExpress. Niniejsza praca skupia się jednak tylko na tych, które oferują możliwość sprzedaży lub wymiany produktów m.in. używanych, w kontekście ograniczenia konsumpcjonizmu w społeczeństwie. W polskim sektorze e-commerce serwisy aukcyjne odgrywają znaczącą rolę, umożliwiając użytkownikom sprzedaż oraz zakup różnorodnych produktów z tzw. „drugiej ręki”. Wśród dostępnych platform szczególnie wyróżniają się OLX oraz Vinted, które choć działają w nieco innych niszach, cieszą się dużą popularnością wśród konsumentów.

OLX to jedna z najpopularniejszych platform ogłoszeniowych w Polsce, działająca również na arenie międzynarodowej. Serwis umożliwia użytkownikom zamieszczanie darmowych ogłoszeń dotyczących sprzedaży, kupna, wynajmu, a także wymiany różnorodnych produktów i usług. OLX stanowi miejsce, w którym zarówno osoby prywatne, jak i firmy mogą szybko i efektywnie nawiązać kontakt z potencjalnymi klientami lub kontrahentami. Kluczową zaletą OLX jest jego wszechstronność – serwis obejmuje szeroki zakres kategorii, takich jak nieruchomości, motoryzacja, elektronika, meble, moda, czy praca. Użytkownicy mają możliwość precyzyjnego wyszukiwania ofert dzięki zaawansowanym filtrom, co pozwala im odnaleźć interesujące produkty lub usługi w swojej okolicy. Intuicyjny interfejs serwisu sprawia, że korzystanie z platformy jest proste i przyjemne, nawet dla osób nieposiadających zaawansowanych umiejętności technologicznych.

Vinted to internetowa platforma dedykowana kupnie, sprzedaży oraz wymianie używanej odzieży, obuwia i akcesoriów, która zdobyła dużą popularność szczególnie wśród młodszych użytkowników. Powstała z myślą o stworzeniu miejsca, gdzie każdy może łatwo i szybko pozbyć się niepotrzebnych ubrań, a jednocześnie znaleźć unikalne, modowe okazje w atrakcyjnych cenach. Platforma ta kładzie nacisk na aspekty zrównoważonego rozwoju, promując ideę ponownego wykorzystania odzieży, co wpisuje się w szerszy trend świadomej konsumpcji i dbania o środowisko. Jednym z głównych atutów Vinted jest przyjazny i intuicyjny interfejs, który umożliwia łatwe przeglądanie i publikowanie ofert. Użytkownicy mogą dodawać zdjęcia, opisy oraz ustalać ceny swoich produktów, a także korzystać z zaawansowanych filtrów wyszukiwania, które pomagają

znaleźć dokładnie to, czego szukają. Platforma umożliwia również bezpośrednią komunikację między sprzedającymi a kupującymi, co ułatwia negocjacje cenowe i budowanie zaufania wśród społeczności. Vinted wyróżnia się także silnym elementem społecznościowym. Użytkownicy często dzielą się swoimi doświadczeniami, opiniami i poradami dotyczącymi mody. Ta interakcja sprzyja tworzeniu autentycznej i zaangażowanej społeczności, w której każdy może poczuć się częścią większego ruchu promującego zrównoważony styl życia. Dzięki temu, platforma nie tylko umożliwia handel używaną odzieżą, ale również wspiera ideę świadomej mody i odpowiedzialnej konsumpcji, co przyczynia się do zmniejszenia negatywnego wpływu przemysłu odzieżowego na środowisko.

W niniejszej pracy dyplomowej skupiono się na zaproponowaniu funkcjonalnego rozwiązania dla platformy opartej na modelu sprzedaży C2C w kontekście wpływu systemu C2C na zmniejszenie konsumpcjonizmu w społeczeństwie.

3. Badania własne / opis metod, technik i narzędzi badawczych / aparatura / oprogramowanie

3.1. Technologie i standardy wykorzystane w projekcie

Programowanie obiektowe w PHP (OOP), skupia się na tworzeniu programów poprzez definiowanie i wykorzystywanie obiektów. Obiekt to instancja klasy, czyli struktury, która łączy w sobie dane (właściwości) oraz metody (funkcje) operujące na tych danych. Dzięki temu podejściu kod staje się bardziej modularny, łatwiejszy do zarządzania, utrzymania i rozwoju.¹⁰

Cechy charakterystyczne OOP w PHP:

- **Modularność:** Dzięki podziałowi kodu na klasy i obiekty, łatwiej jest zarządzać poszczególnymi fragmentami aplikacji.
- **Łatwość utrzymania:** Kod napisany w stylu obiektowym jest bardziej przejrzysty, co ułatwia jego modyfikacje i rozbudowę.
- **Ponowne użycie kodu:** Dziedziczenie i polimorfizm umożliwiają wykorzystywanie już istniejących rozwiązań w nowych kontekstach.

¹⁰ Wójcik, A., Wolski, M., & Smółka, J. B. (2019). Analiza wydajnościowa frameworka Symfony do tworzenia nowoczesnych aplikacji webowych na podstawie wybranych wersji. *Journal of Computer Sciences Institute*, 13, 293-297.

- Skalowalność: Aplikacje oparte na OOP są lepiej przystosowane do rozwoju i mogą być łatwiej rozbudowywane wraz ze wzrostem potrzeb projektu.
- Bezpieczeństwo: Enkapsulacja pozwala na kontrolowanie dostępu do danych, co zwiększa bezpieczeństwo aplikacji.

Bootstrap to otwartoźródłowy framework front-endowy oparty na HTML, CSS i JavaScript, który umożliwia szybkie tworzenie responsywnych oraz mobilnych stron internetowych i aplikacji. Został stworzony przez zespół programistów z Twittera.

Cechy charakterystyczne Bootstrap:

- Responsywność i podejście mobile-first: system siatki (grid system) umożliwia tworzenie elastycznych układów, które automatycznie dostosowują się do różnych rozdzielczości ekranów, od smartfonów po duże monitory.
- Gotowe komponenty i szablony: framework oferuje wybór gotowych komponentów, takich jak przyciski, formularze, nawigacje, modale, karuzele, alerty czy tooltipy. Komponenty te są łatwe do użycia i dostosowania, co pozwala na szybkie prototypowanie i wdrażanie interfejsów użytkownika bez konieczności pisania dużej ilości kodu od podstaw.
- Spójny i estetyczny design: Bootstrap zapewnia jednolity wygląd i stylizację elementów interfejsu, co pomaga w utrzymaniu spójności wizualnej całego projektu.
- Rozszerzalność i personalizacja: framework jest konfigurowalny. Bootstrap można rozszerzać o komponenty lub integrować z innymi bibliotekami JavaScript.

MVC, czyli Model-View-Controller, to popularny wzorzec architektoniczny stosowany w projektowaniu aplikacji, którego głównym celem jest oddzielenie logiki biznesowej od warstwy prezentacji. Dzięki temu aplikacja staje się bardziej modularna, łatwiejsza do utrzymania, rozwijania oraz testowania.

Cechy charakterystyczne MVC:

- Model: reprezentuje dane aplikacji oraz logikę biznesową. To on zarządza przechowywaniem, pobieraniem i modyfikacją danych, a także implementuje zasady i reguły działania systemu. Funkcje: przechowywanie danych (np. za pomocą baz danych); implementacja logiki biznesowej i reguł przetwarzania

danych; aktualizacja stanu aplikacji w odpowiedzi na działania użytkownika lub zmiany w otoczeniu.

- Widok (View): odpowiada za prezentację danych użytkownikowi. To warstwa interfejsu, w której dane z modelu są przekształcane w zrozumiałą i atrakcyjną formę wizualną. Funkcje: generowanie interfejsu użytkownika (np. strony internetowe, aplikacje desktopowe, mobilne); wyświetlanie informacji pobranych z modelu; odbieranie pewnych interakcji użytkownika, które są następnie przekazywane do kontrolera.
- Kontroler (Controller): pełni rolę pośrednika między modelem a widokiem. Odbiera żądania od użytkownika, przetwarza je i decyduje, które dane z modelu powinny zostać wyświetlone w widoku. Funkcje: interpretacja i obsługa akcji użytkownika (np. kliknięcia, wpisywanie danych); aktualizacja modelu na podstawie działań użytkownika; wybór odpowiedniego widoku do prezentacji wyników operacji.

MySQL to system zarządzania relacyjnymi bazami danych (RDBMS), który opiera się na języku SQL (Structured Query Language). Jest to rozwiązanie otwartoźródłowe, co oznacza, że jego kod jest dostępny publicznie, a użytkownicy mają możliwość swobodnej modyfikacji oraz rozpowszechniania programu zgodnie z warunkami licencji GPL. MySQL jest szeroko stosowany w aplikacjach webowych, często jako część stosu technologicznego LAMP (Linux, Apache, MySQL, PHP/Perl/Python), co przyczyniło się do jego ogromnej popularności wśród deweloperów na całym świecie.¹¹

Cechy charakterystyczne MySQL:

- Wydajność i skalowalność: MySQL został zaprojektowany z myślą o wysokiej wydajności i możliwości obsługi dużej liczby jednoczesnych zapytań. Dzięki optymalizacjom oraz wsparciu dla różnych silników baz danych (np. InnoDB), system potrafi efektywnie zarządzać zarówno małymi, jak i bardzo dużymi zbiorami danych. Skalowalność MySQL umożliwia jego zastosowanie w projektach o rosnących wymaganiach, zarówno pod względem wielkości przechowywanych danych, jak i liczby obsługiwanych użytkowników.
- Bezpieczeństwo: system oferuje zaawansowane mechanizmy zabezpieczeń, w tym kontrolę dostępu, autoryzację użytkowników oraz szyfrowanie połączeń.

¹¹ Welling, L., & Thomson, L. (2003). PHP and MySQL Web development. Sams publishing.

- Wsparcie dla języka SQL: MySQL wykorzystuje SQL, standardowy język do komunikacji z bazami danych, co umożliwia wykonywanie operacji takich jak tworzenie, modyfikacja, pobieranie i usuwanie danych.
- Elastyczność i integracja: MySQL łatwo integruje się z różnorodnymi językami programowania i frameworkami, co czyni go wszechstronnym narzędziem w różnych środowiskach aplikacyjnych.
- Wsparcie społeczności i dokumentacja: jako projekt open-source, MySQL posiada rozbudowaną społeczność użytkowników i deweloperów, która aktywnie współtworzy dokumentację, udziela wsparcia i rozwija nowe funkcjonalności.

3.2. Narzędzia wykorzystywane podczas pracy nad projektem dyplomowym

Visual Studio Code (VS Code) to wieloplatformowy edytor kodu źródłowego stworzony przez firmę Microsoft. VS Code działa na systemach Windows, macOS oraz Linux. Edytor Obsługuje wiele języków programowania takich jak JavaScript, Python, C++, Java. Posiada zintegrowane narzędzia do debugowania, które umożliwiają analizę i naprawę błędów w kodzie bez konieczności opuszczania edytora. Wbudowana obsługa Git i innych systemów kontroli wersji umożliwia łatwe zarządzanie repozytoriami, śledzenie zmian w kodzie. Visual Studio Code oferuje także bibliotekę rozszerzeń dostępnych w marketplace.

XAMPP to otwartoźródłowy pakiet oprogramowania, który umożliwia łatwe tworzenie lokalnego środowiska serwerowego do testowania i rozwijania aplikacji webowych. Nazwa XAMPP pochodzi od składników, które zawiera: X – oznacza wieloplatformowość (działa na systemach Windows, Linux i macOS); A – Apache (serwer HTTP); M – MySQL (obecnie często zastępowany przez MariaDB); P – PHP (język skryptowy); P – Perl (język programowania).

GitHub to internetowa platforma hostingowa oparta na systemie kontroli wersji Git, która umożliwia programistom zarządzanie kodem źródłowym oraz współpracę przy projektach informatycznych. Dzięki GitHub możliwe jest łatwe śledzenie zmian w kodzie, kontrola wersji oraz koordynacja pracy zespołów, co znacząco przyspiesza rozwój oprogramowania. GitHub opiera się na systemie Git, który pozwala na tworzenie repozytoriów, zarządzanie gałęziami, łączenie zmian oraz cofanie modyfikacji w kodzie. Platforma umożliwia współpracę nad projektami poprzez mechanizmy pull requestów,

które pozwalają na zgłaszanie zmian, ich przegląd oraz integrację z główną gałęzią kodu. Dodatkowo GitHub oferuje narzędzia do zarządzania zadaniami, zgłaszania problemów (issue tracking) oraz dokumentacji, co ułatwia organizację pracy nad projektami.

phpMyAdmin to otwartoźródłowe narzędzie webowe napisane w PHP, służące do zarządzania bazami danych MySQL oraz MariaDB. W środowisku XAMPP jest standardowym komponentem, który umożliwia łatwą administrację bazami danych za pomocą intuicyjnego interfejsu graficznego dostępnego przez przeglądarkę internetową. W pakiecie XAMPP phpMyAdmin jest zintegrowany jako narzędzie do zarządzania lokalnym środowiskiem bazodanowym. Ułatwia to deweloperom testowanie aplikacji webowych na ich własnych komputerach, zanim zostaną wdrożone na serwerze produkcyjnym. Narzędzie pozwala na tworzenie kopii zapasowych baz danych, przeglądanie struktury bazy, edycję danych bezpośrednio w tabelach oraz monitorowanie jej wydajności.

PlantUML.com to oficjalna strona narzędzia PlantUML, które umożliwia tworzenie diagramów (np. UML, diagramów sekwencji, klas, przypadków użycia, aktywności, stanów) przy użyciu prostego, tekstowego języka opisu. Narzędzie to pozwala na szybkie i efektywne generowanie diagramów na podstawie kodu, co ułatwia dokumentowanie architektury systemów oraz komunikację między członkami zespołu projektowego. Użytkownik definiuje diagramy za pomocą prostego języka opartego na tekstowych notacjach. Dzięki temu, zamiast korzystać z graficznych edytorów, można łatwo tworzyć, modyfikować i wersjonować diagramy jako pliki tekstowe. Generowanie diagramów odbywa się dynamicznie – wprowadzone zmiany w tekście są natychmiast odzwierciedlane w wizualizacji diagramu. To pozwala na szybkie eksperymentowanie i iteracyjne dopracowywanie diagramów.

DBDiagram.io to internetowa platforma dedykowana do tworzenia, wizualizacji oraz analizy diagramów baz danych. Narzędzie to jest szczególnie przydatne dla deweloperów, architektów systemów oraz administratorów baz danych, którzy potrzebują szybkiego i intuicyjnego sposobu na przedstawienie struktury swoich baz danych w formie graficznej. DBDiagram.io często oferuje funkcje umożliwiające import istniejących struktur baz danych z popularnych systemów zarządzania (np. MySQL, PostgreSQL, SQL Server).

Po wgraniu danych, narzędzie automatycznie generuje diagram, który można następnie modyfikować według potrzeb.

3.3. Schemat i model bazy danych

3.3.1. Założenia

Projektowana baza danych ma za zadanie wspierać funkcjonalności systemu aukcyjnego, w którym użytkownicy (o trzech poziomach uprawnień: administrator, moderator oraz zwykły użytkownik) mogą wystawiać aukcje, komunikować się w ramach transakcji sprzedaży oraz zarządzać informacjami o swoich kontach. Kluczowym aspektem projektu jest zapewnienie spójności danych oraz łatwej rozbudowy struktury w przyszłości. W tym celu zastosowano relacyjny model bazy danych z odpowiednimi ograniczeniami (klucze główne, obce) oraz powiązaniem między tabelami.

3.3.2. Struktura bazy danych

Baza (rys. 2) składa się z ośmiu głównych tabel, których struktury oraz wzajemne relacje opisano poniżej:

1. Tabela accounts

Tabela accounts przechowuje dane dotyczące użytkowników systemu.

Główne pola:

accountid – identyfikator użytkownika (klucz główny, autoinkrementacja), unikalnie identyfikuje każde konto.

login oraz password – dane niezbędne do uwierzytelnienia użytkownika.

registerdate – data rejestracji konta; wartość domyślna to bieżący znacznik czasowy.

account_type – identyfikator typu konta (np. administrator, moderator, user), przy czym wartość ta odnosi się do tabeli type.

verified – flaga wskazująca, czy konto zostało zweryfikowane.

whoever – pole określające, które konto (przez identyfikator) dokonało weryfikacji danego konta; relacja ta jest samoodwołująca się, gdyż wskazuje na accountid w tej samej tabeli.

Pozostałe pola (takie jak first_name, lastname, email, phone, address, codezip, city, country) zawierają dodatkowe informacje personalne użytkownika.

2. Tabela type

Tabela type definiuje rodzaje kont, dzięki czemu możliwe jest rozróżnienie uprawnień i ról użytkowników.

Główne pola:

type_id – unikalny identyfikator typu (klucz główny).

type_name – nazwa typu, np. „administrator”, „moderator”, „user”.

Tabela ta jest wykorzystywana przez tabelę accounts przy określaniu poziomu uprawnień użytkowników.

3. Tabela auctions

Tabela auctions przechowuje informacje o wystawionych aukcjach.

Główne pola:

auctionid – unikalny identyfikator aukcji (klucz główny, autoinkrementacja).

accountid – identyfikator sprzedawcy, czyli użytkownika wystawiającego aukcję; pole to stanowi klucz obcy odnoszący się do accounts.accountid.

categoryid – odniesienie do kategorii, w której sklasyfikowana jest aukcja (klucz obcy do tabeli category).

title oraz description – tytuł aukcji i szczegółowy opis przedmiotu aukcji.

used oraz private – pola określające stan przedmiotu (np. używany/nowy) oraz czy aukcja jest prywatna.

date_start i date_end – daty rozpoczęcia oraz zakończenia aukcji.

verified – pole określające, czy aukcja została zweryfikowana przez uprawniony podmiot (np. moderatora).

whoever – identyfikator konta, które dokonało weryfikacji aukcji; stanowi klucz obcy wskazujący na accounts.accountid.

selled – flaga określająca, czy przedmiot został sprzedany.

buyerid – identyfikator kupującego (klucz obcy do accounts.accountid), obecny w przypadku zakończonych transakcji.

price – cena wywoławcza lub ostateczna cena sprzedaży.

currencyid – odniesienie do waluty, w której ustalona jest cena (klucz obcy do tabeli currency).

4. Tabela category

Tabela category zawiera informacje o kategoriach przedmiotów aukcyjnych, co umożliwia uporządkowanie aukcji według określonych grup.

Główne pola:

categoryid – identyfikator kategorii (klucz główny, autoinkrementacja).

name – nazwa kategorii.

in_tree – pole pomocnicze, które może służyć do organizacji hierarchicznej kategorii.

5. Tabela currency

Tabela currency definiuje dostępne waluty wykorzystywane w aukcjach.

Główne pola:

currencyid – identyfikator waluty (klucz główny, autoinkrementacja).

currency_name – nazwa waluty (np. „PLN”, „EUR”).

Aukcje odnoszą się do tej tabeli poprzez pole currencyid, co pozwala na przechowywanie cen w różnych walutach.

6. Tabela file_to_auction

Tabela file_to_auction służy do przechowywania informacji o plikach (najczęściej zdjęciach) powiązanych z aukcjami.

Główne pola:

file_id – unikalny identyfikator pliku (klucz główny, autoinkrementacja).

auctionid – identyfikator aukcji, do której plik jest przypisany (klucz obcy do auctions.auctionid).

filename – nazwa pliku.

file_path – ścieżka do pliku w systemie plików lub zasobach serwera.

Dzięki tej tabeli możliwe jest powiązanie aukcji z obrazami przedmiotów, co znacząco zwiększa atrakcyjność wizualną prezentacji.

7. Tabela message_link

Tabela message_link pełni funkcję łączenia komunikatów w ramach pojedynczej konwersacji między sprzedającym a kupującym.

Główne pola:

mlid – unikalny identyfikator wątku wiadomości (klucz główny, autoinkrementacja).

auctionid – identyfikator aukcji, w której prowadzona jest konwersacja (klucz obcy do auctions.auctionid).

sellerid – identyfikator sprzedawcy (klucz obcy do accounts.accountid).

buyerid – identyfikator kupującego (klucz obcy do accounts.accountid).

Tabela umożliwia organizację wiadomości w logiczne wątki, co ułatwia późniejsze przeszukiwanie historii komunikacji.

8. Tabela message

Tabela message przechowuje poszczególne wiadomości wymieniane w ramach aukcji.

Główne pola:

id – unikalny identyfikator wiadomości (klucz główny, autoinkrementacja).

mlid – identyfikator wątku wiadomości, pozwalający na powiązanie z tabelą message_link.

auctionid – identyfikator aukcji, do której wiadomość się odnosi (klucz obcy).

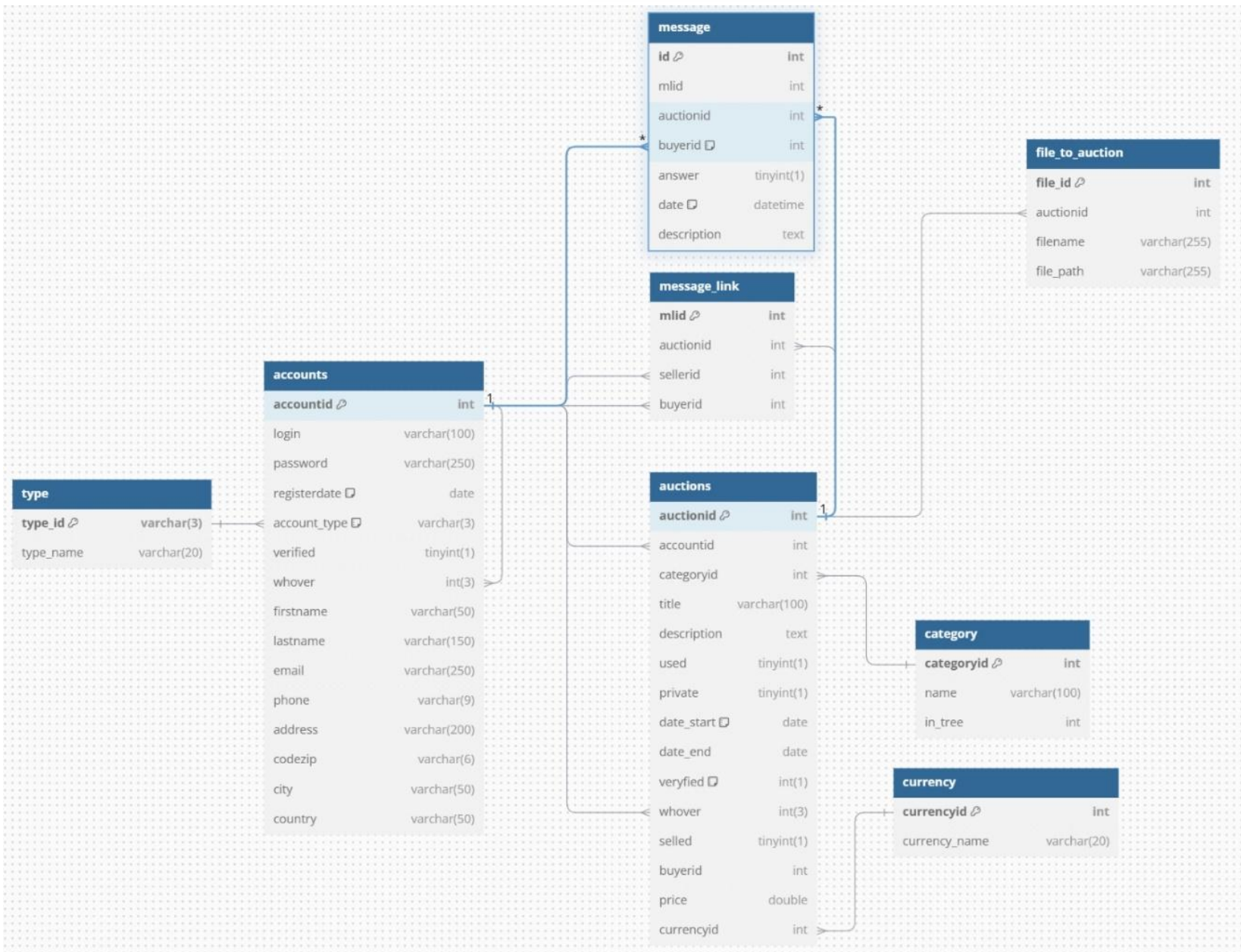
buyerid – identyfikator konta użytkownika (kupującego), który wysłał wiadomość; jest to pole o ograniczonej długości (3-cyfrowe) w kontekście przyjętej specyfikacji.

answer – flaga informująca, czy wiadomość jest odpowiedzią sprzedającego (wartość 1) czy zapytaniem kupującego (wartość 0).

date – data oraz godzina wysłania wiadomości, z wartością domyślną ustawioną na bieżący znacznik czasowy.

description – treść wiadomości.

Tabela ta umożliwia przechowywanie pełnej historii komunikacji pomiędzy użytkownikami, co jest kluczowe dla rozwiązywania ewentualnych sporów oraz dokumentacji transakcji.



Rys. 2 Baza danych.

3.3.3. Relacje między tabelami

Projekt bazy danych wykorzystuje relacje oparte na kluczach obcych, które gwarantują spójność danych. Kluczowe zależności to:

Relacja między accounts a type:

Pole `account_type` w tabeli `accounts` odnosi się do `type.type_id`, co umożliwia określenie roli danego użytkownika.

Samoodwołanie w tabeli accounts:

Pole `whoever` w tabeli `accounts` jest kluczem obcym odnoszącym się do `accounts.accountid` i wskazuje, które konto dokonało weryfikacji danego konta.

Relacje w tabeli auctions:

Pole `accountid` odnosi się do użytkownika wystawiającego aukcję (tabela `accounts`).

Pole `categoryid` wskazuje na kategorię aukcji (tabela `category`).

Pole `currencyid` definiuje walutę aukcji (tabela `currency`).

Pole `whoever` informuje o koncie, które zweryfikowało aukcję (również `accounts`).

Pole `buyerid` odnosi się do kupującego, w przypadku zakończonej transakcji.

Relacja między `file_to_auction` a `auctions`:

Pole `auctionid` w tabeli `file_to_auction` łączy dany plik (np. zdjęcie) z odpowiednią aukcją.

Relacje w kontekście komunikacji:

Tabela `message_link` łączy aukcje z konwersacjami, wskazując identyfikatory sprzedawcy i kupującego.

Tabela `message` zawiera poszczególne wiadomości, gdzie pole `auctionid` odnosi się do aukcji, a `buyerid` – do konta użytkownika wysyłającego wiadomość.

Klucz `mlid` w tabeli `message` pozwala zgrupować wiadomości w ramach jednego wątku komunikacyjnego, powiązanego z tabelą `message_link`.

3.3.4. Podsumowanie

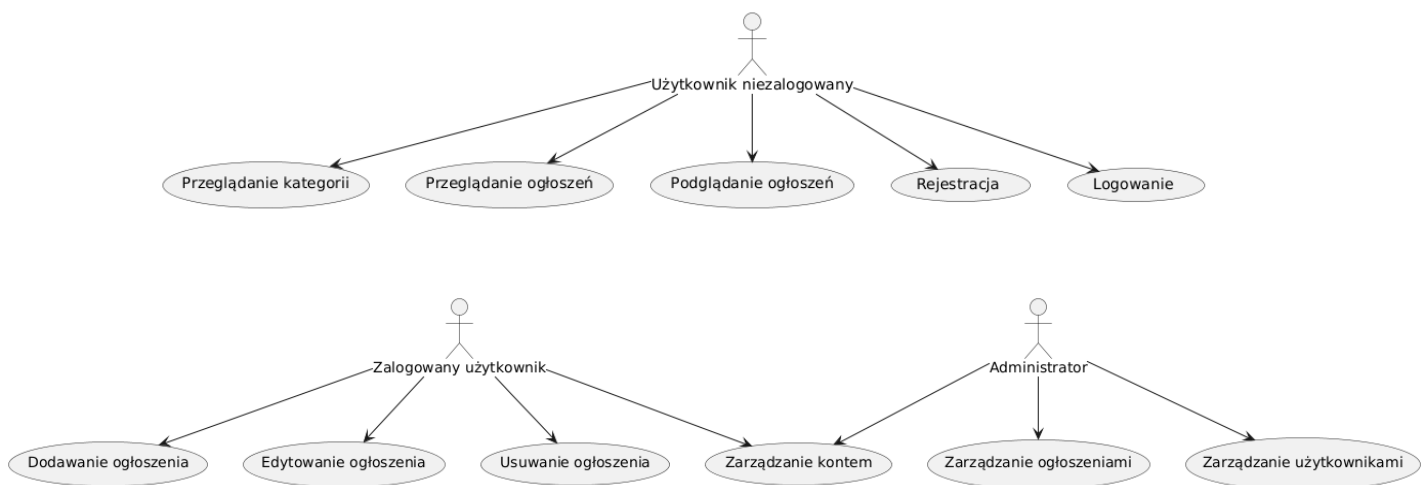
Projektowana baza danych została zaprojektowana z myślą o zapewnieniu wysokiej spójności i integralności danych w systemie aukcyjnym opartym na modelu sprzedażowym C2C. Dzięki zastosowaniu relacyjnego modelu danych, precyzyjnie zdefiniowanym tabelom oraz powiązaniom kluczowym możliwe jest efektywne przechowywanie i przetwarzanie informacji o użytkownikach, aukcjach, kategoriach, walutach, zdjęciach oraz komunikacji między stronami transakcji. Struktura bazy danych umożliwia również łatwą rozbudowę o dodatkowe funkcjonalności w przyszłych etapach rozwoju systemu.

Taki model bazy stanowi solidną podstawę dla implementacji systemu aukcyjnego, gwarantując zarówno wydajność operacyjną, jak i przejrzystość architektury danych.

3.4. Opracowanie modelu UML

3.4.1. Diagram przypadków użycia (Use Case Diagram)

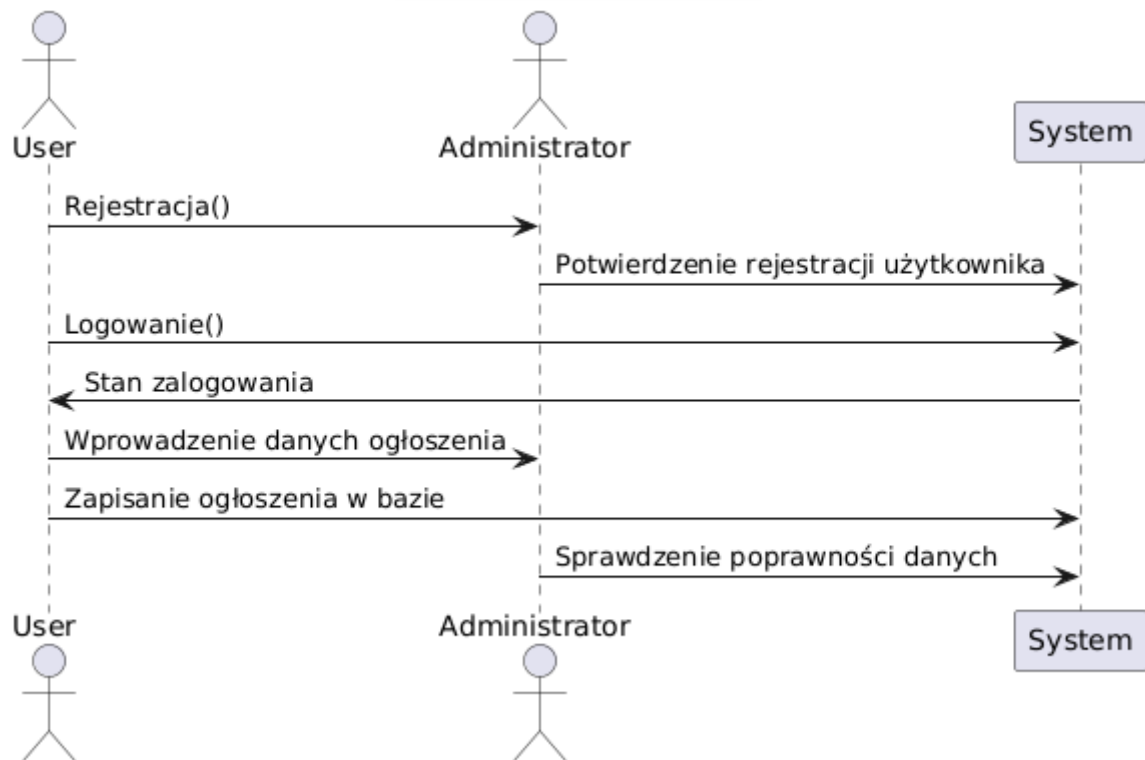
Diagram przypadków użycia (rys. 3) przedstawia główne funkcjonalności systemu z perspektywy użytkowników (aktorów). W przypadku strony C2C możemy wyróżnić następujących aktorów: użytkownika niezalogowanego który posiada dostęp do podstawowych informacji jak struktura kategorii czy listy / podglądu podstawowych informacji. 2. Diagram klas (Class Diagram) Diagram klas przedstawia strukturalny model systemu, gdzie klasy reprezentują obiekty w systemie, a ich relacje między sobą. W przypadku strony C2C, diagram może zawierać takie klasy jak: kategorie, aukcje, użytkownicy czy przypisanie relacji pomiędzy userem, a aukcją.



Rys. 3. Diagram przypadków użycia.

3.4.2. Diagram sekwencji (Sequence Diagram)

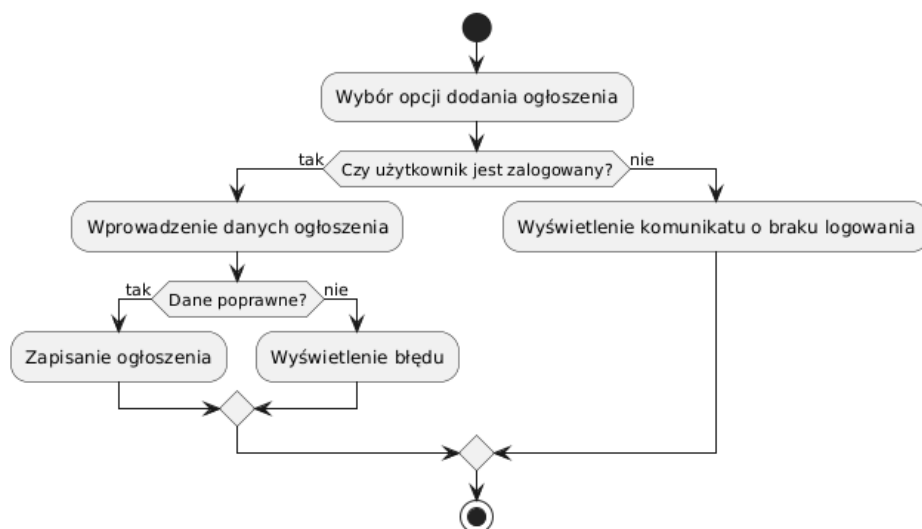
Diagram sekwencji (rys. 4) przedstawia interakcje między obiektami w czasie. Przykład takiego diagramu może pokazywać, jak użytkownik dodaje ogłoszenie na stronę. Przedstawiony poniżej diagram zawiera opis funkcjonowania aukcji.



Rys. 4. Diagram sekwencji.

3.4.3. Diagram aktywności (Activity Diagram)

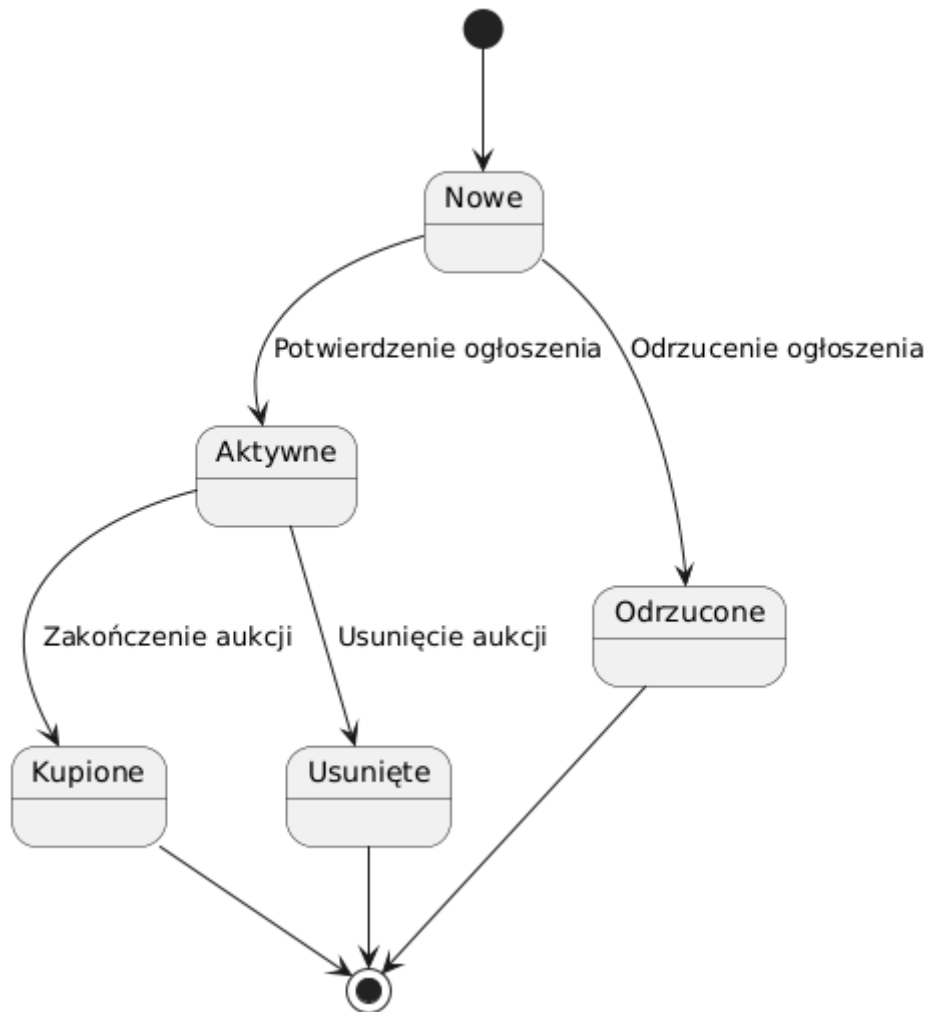
Diagram aktywności (rys. 5) przedstawia przepływ pracy w systemie. W niniejszym projekcie diagram zobrazony jest na podstawie procesu dodawania ogłoszenia przez użytkownika.



Rys. 5. Diagram aktywności

3.4.4. Diagram stanu (State Diagram)

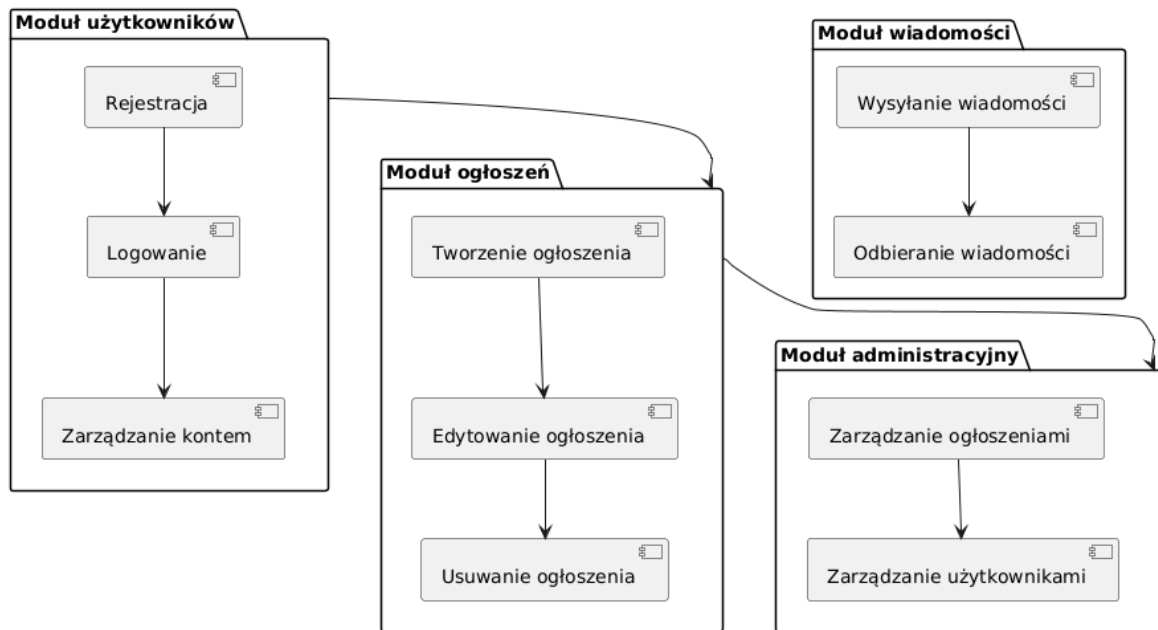
Diagram stanu (rys. 6) przedstawia różne stany obiektu w systemie i przejścia między nimi. Na przykład, diagram stanu dla ogłoszenia może wyglądać następująco:



Rys. 6. Diagram stanu

3.4.5. Diagram komponentów (Component Diagram)

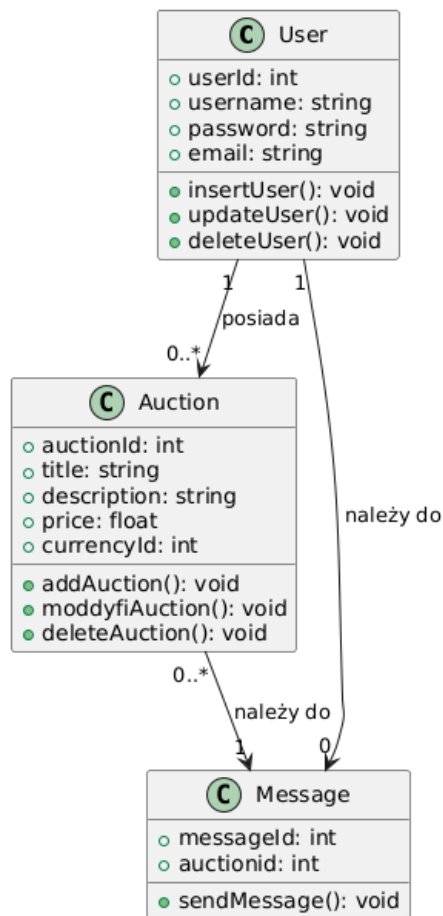
Diagram komponentów (rys. 7) przedstawia podział systemu na poszczególne komponenty. W przypadku strony C2C mogą to być: Moduł użytkowników: Odpowiada za rejestrację, logowanie, zarządzanie kontem. Moduł ogłoszeń: Zarządza procesem dodawania, edytowania, usuwania ogłoszeń. Moduł wiadomości: Obsługuje komunikację między użytkownikami. Moduł administracyjny: Zarządza administracją ogłoszeń, użytkowników, raportami.



Rys. 7. Diagram komponentów

3.4.6. Diagram klas

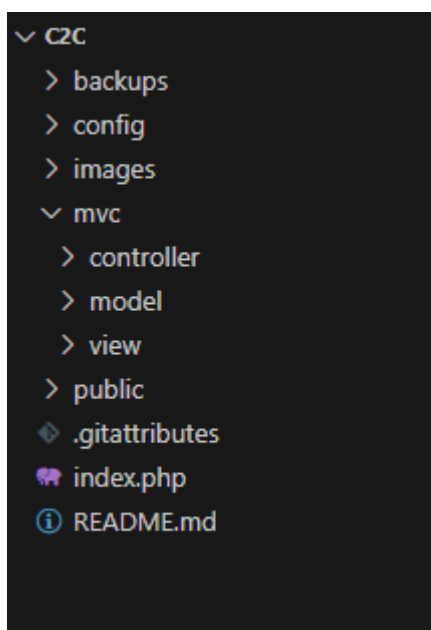
Rysunek nr 8 przedstawia diagram klas. Diagram klas to wizualna reprezentacja struktury systemu w programowaniu obiektowym. Pokazuje klasy, ich atrybuty, metody oraz zależności między nimi, takie jak dziedziczenie czy powiązania. Służy do modelowania systemu, ułatwiając zrozumienie relacji między obiektami w programie.



Rys. 8. Diagram klas.

3.5. Implementacja

A. Struktura projektu jest wykonana w modelu drzewa MVC.



B. W kontrolerze znajdują się pliki z kodem podzielone na główne zadania na stronie www:

- Controller odpowiadający za usunięcie aukcji

```
deleteauction.php X
mvc > controller > deleteauction.php
1 <?php
2
3 include_once '../model/auctionModel.php';
4 $model = new Auction();
5
6 try {
7     $stat = $model->deleteAuction($_GET['auction_id']);
8     // header("location: ../view/auctions/userauctions.php");
9
10 } catch (PDOException $e) {
11     echo "Błąd połączenia: " . $e->getMessage();
12 }
13
14 }
15 >>
```

- Controller odpowiadający za wyszukanie aukcji z danej kategorii lub z daną frazą w tytule

```
forward.php X
mvc > controller > forward.php
1 <?php
2 session_start();
3 //var_dump($_GET['categoryId']);
4 if(isset($_GET['categoryId']) && $_GET['categoryId']!=null){
5     $_SESSION['filter']['categoryId'] = $_GET['categoryId'];
6     header('location: ../view/auctions/auction.php');
7 }
8
9 if(isset($_GET['search'])){
10     $_SESSION['filter']['search'] = $_GET['search'];
11     header('location: ../view/auctions/auction.php');
12 }
13 >>
```

- Controller odpowiadający za wyznaczenie uprawnień

```
logged.php X
mvc > controller > logged.php
1 <?php
2 session_start();
3 print_r($_SESSION["logged"]);
4
5 if (!isset($_SESSION["logged"]) || session_status() != 2){
6     $_SESSION["role"] = "guest";
7     header("location: ../index.php");
8 }else{
9     switch ($_SESSION["logged"]["account_type"]){
10         case 101:
11             $_SESSION["role"] = "admin";
12             break;
13         case 222:
14             $_SESSION["role"] = "user";
15             break;
16     }
17     header("location: ../index.php");
18 }
19
20 if (isset($_SESSION["logged"]["last_activity"])) {
21     $lastActivityTime = $_SESSION["logged"]["last_activity"];
22     $currentTime = time();
23     $sessionTimeout = 1800; //30 minut
24     // $sessionTimeout = 60; //1 minuta
25
26     if ($currentTime - $lastActivityTime <= $sessionTimeout){
27         //echo "Sesja nadal jest aktywna";
28     }else{
29         //echo "Sesja nieaktywna";
30         $_SESSION["error"] = "Sesja zakończona, zaloguj się ponownie";
31         $role = "guest";
32         header("location: ../index.php");
33     }
34 }
35
36 }
37 }
38 }
39 }
40 }
```

- Controller odpowiadający za logowanie

```
login.php X
mvc > controller > login.php
1 <?php
2 session_start();
3
4 foreach ($_POST as $value){
5     if (empty($value)){
6         $_SESSION["error"] = "Wypełnij wszystkie dane!";
7         echo "<script>history.back();</script>";
8         exit();
9     }
10 }
11
12 $config = include 'C:/xampp/htdocs/in1/c2c/config/config.php';
13 include_once '../model/userModel.php';
14
15 try {
16     $model = new User();
17     $connect = $model -> verifyUser($_POST["email"], $_POST["password"]);
18
19     if ($connect){
20         $user = $connect;
21
22         $pass = password_hash($_POST["pass"], PASSWORD_ARGON2ID);
23
24         if ($user["verified"]==1){
25             if (password_verify($_POST["pass"], $user["password"])) {
26                 $_SESSION["logged"]["firstName"] = $user["firstName"];
27                 $_SESSION["logged"]["lastName"] = $user["lastName"];
28                 $_SESSION["logged"]["session_id"] = session_id();
29                 $_SESSION["logged"]["account_id"] = $user["accountid"];
30                 //echo session_status();
31                 $_SESSION["logged"]["account_type"] = $user["account_type"];
32                 $_SESSION["logged"]["last_activity"] = time();
33                 //print_r($_SESSION["logged"]);
34                 header("location: ../logged.php");
35             } else {
36                 $_SESSION["error"] = "Nie udało się zalogować!";
37                 echo "<script>history.back();</script>";
38             }
39         }elseif ($user["verified"]==0){
40             $_SESSION["error"] = "Konto nie zostało zweryfikowane przez administratorów, proszę czekać";
41             echo "<script>history.back();</script>";
42         }elseif ($user["verified"]==2){
43             $_SESSION["error"] = "Konto zostało odrzucone przez admina, aby dowiedzieć się dlaczego napisz do nas";
44             echo "<script>history.back();</script>";
45         }else{
46             $_SESSION["error"] = "Nieznany błąd";
47             echo "<script>history.back();</script>";
48         }
49     }
50 }
```

- Controller odpowiadający za wysyłki wiadomości

```
mesview.php X
mvc > controller > mesview.php
1 <<?php
2 session_start();
3 include_once '../model/messageModel.php';
4 //print_r($_POST);
5 $model = new Message();
6 try {
7     $stmt = $model -> sendMessage($_POST["auctionid"], $_POST["buyerid"], $_POST["description"], $_POST["answer"]);
8     header("location: ../view/messages/messageview.php?aid=$_POST[buyerid]&bid=$_POST[buyerid]");
9 } catch (PDOException $e) {
10     echo "Błąd połączenia: " . $e->getMessage();
11 }
12 ?>
```

- Controller odpowiadający za weryfikację aukcji przez administratora

```
modauction.php X
mvc > controller > modauction.php
1 <?php
2
3 include_once '../model/auctionModel.php';
4 $model = new Auction();
5
6 try {
7     //var_dump($_GET["verified"]);
8     if($_GET["verified"] == "true"){
9         $verified = 1;
10     } else {
11         $verified = 2;
12     }
13     $stmt = $model -> verifyAuction($verified, $_GET["id"], $_GET["auction_id"]);
14     header("location: ../view/administrativs/admin.php");
15 } catch (PDOException $e) {
16     echo "Błąd połączenia: " . $e->getMessage();
17 }
18 ?>
```

- Controller odpowiadający za weryfikację nowych użytkowników przez administratora

```

modify.php
mvc > controller > modify.php
1 <?php
2
3 include_once '../model/auctionModel.php';
4 $model = new Auction();
5
6 try {
7     $stmt = $model->getAuctionById($_POST["auctionid"]);
8     if($stmt){
9         $title = $stmt["title"] == $_POST["title"] && !$_POST["title"] ? $stmt["title"] : $_POST["title"];
10        $categoryId = $stmt["categoryId"] == $_POST["categoryId"] && !$_POST["categoryId"] ? $stmt["categoryId"] : $_POST["categoryId"];
11        $description = $stmt["description"] == $_POST["description"] && !$_POST["description"] ? $stmt["description"] : $_POST["description"];
12        $used = isset($_POST["used"]) ? $_POST["used"] : false;
13        $price = $stmt["price"] == $_POST["price"] && !$_POST["price"] ? $stmt["price"] : $_POST["price"];
14        $currencyid = $stmt["currencyid"] == $_POST["currencyid"] && !$_POST["currencyid"] ? $stmt["currencyid"] : $_POST["currencyid"];
15        $auctionid = $stmt["auctionid"] == $_POST["auctionid"] && !$_POST["auctionid"] ? $stmt["auctionid"] : $_POST["auctionid"];
16
17        $stmt = $model->updateAuction($title, $categoryId, $description, $used, 1, $price, $currencyid, $auctionid);
18
19        header("Location: ../view/auctions/userauctions.php");
20    }
21 } catch (PDOException $e) {
22     echo "Błąd połączenia: " . $e->getMessage();
23 }
24
25 ?>

```

- Controller odpowiadający za podstawową obsługę aukcji

```

UserController.php
mvc > controller > UserController.php
1 <?php
2
3 require_once '../model/auctionModel.php';
4 require_once '../model/userModel.php';
5
6 class AuctionController {
7
8     private $auctionModel;
9     private $userModel;
10
11     public function __construct() {
12         $this->auctionModel = new Auction();
13         $this->userModel = new User();
14     }
15
16     // Pobierz dane o użytkowniku
17     public function profile() {
18         session_start();
19         if (!isset($_SESSION['logged']) || !isset($_SESSION['account_id'])) {
20             header('Location: ../view/login.php');
21             exit();
22         }
23
24         $user = $this->userModel->getUserById($_SESSION['logged']['account_id']);
25
26         if ($_SERVER['REQUEST_METHOD'] === 'POST') {
27             $updated = $this->userModel->updateUser($_POST);
28             $user = $this->userModel->getUserById($_SESSION['logged']['account_id']);
29         }
30
31         include __DIR__ . '/../view/user_profile.php';
32     }
33
34     // Zweryfikuj dane rejestracji
35     public function validateRegistration($data) {
36         $errors = [];
37
38         foreach ($data as $value) {
39             if (empty($value)) {
40                 $errors[] = "Wypełnij wszystkie dane!";
41             }
42         }
43
44         if (!isset($data['terms'])) {
45             $errors[] = "Zatwierdź regulamin!";
46         }
47
48         if ($data["pass1"] != $data["pass2"]) {

```

- Controller odpowiedzialny za podstawowe dane użytkownika

```

userPasswordController.php
mvc > controller > userPasswordController.php
1  <?php
2
3  require_once 'Database.php';
4
5  class userPasswordController {
6      private $pdo;
7
8      public function __construct() {
9          $database = new Database();
10         $this->pdo = $database->getConnection();
11     }
12
13     public function validateRegistration($data) {
14         $errors = [];
15
16         foreach ($data as $value) {
17             if (empty($value)) {
18                 $errors[] = "Wypełnij wszystkie dane!";
19             }
20         }
21
22         if (isset($data['terms'])) {
23             $errors[] = "Zatwierdź regulamin!";
24         }
25
26         if ($data['pass1'] !== $data['pass2']) {
27             $errors[] = "Hasła są różne!";
28         }
29
30         if ($data['email1'] !== $data['email2']) {
31             $errors[] = "Adresy email są różne!";
32         }
33
34         return $errors;
35     }
36
37     public function validatePasswordUpdate($data) {
38         $errors = [];
39
40         foreach ($data as $value) {
41             if (empty($value)) {
42                 $errors[] = "Wypełnij wszystkie dane!";
43             }
44         }
45
46         if ($data['pass1'] !== $data['pass2']) {
47             $errors[] = "Hasła nie są identyczne!";
48         }
49     }
50

```

- C. W modelu mamy kod podzielony na główne obszary działania aplikacji i implementujemy tam najważniejsze zadania www do wykonania przez bazę danych.

- Model obsługujący zapytania do bazy danych w obszarze administracyjnym

```

administrativeModel.php
mvc > model > administrativeModel.php
1  <?php
2
3  class Administrative {
4      private $db;
5
6      // Konstruktor obiektu
7      public function __construct() {
8          $this->db = Database::getInstance();
9      }
10
11     // Sprawdź uprawnienia usera
12     public function checkAdmin($accountid) {
13         $query = "SELECT account_type FROM accounts WHERE accountid = :accountid";
14         $stmt = $this->db->prepare($query);
15         $stmt->bindParam(':accountid', $this->accountid, PDO::PARAM_INT);
16         $stmt->execute();
17         $user = $stmt->fetch(PDO::FETCH_ASSOC);
18     }
19
20     // Pobierz wszystkie nieweryfikowane aukcje
21     public function getUnverifiedAuctions() {
22         $query = "SELECT
23             a.auctionid, a.title, a.seller, u.firstname, u.lastname, u.city, a.date_start, a.price, c.currency_name
24             FROM auctions a
25             LEFT JOIN accounts u ON a.accountid = u.accountid
26             LEFT JOIN currency c ON a.currencyid = c.currencyid
27             WHERE a.verified = 0";
28         $stmt = $this->db->query($query);
29         return $stmt->fetchAll(PDO::FETCH_ASSOC);
30     }
31
32     // Pobierz wszystkich nowych
33     public function getNewUsers($status = 0) {
34         $query = "SELECT * FROM 'accounts' WHERE 'verified' = :status";
35         $stmt = $this->db->prepare($query);
36         $stmt->bindParam(':status', $status, PDO::PARAM_INT);
37         $stmt->execute();
38         return $stmt->fetchAll(PDO::FETCH_ASSOC);
39     }
40 }
41
42 ?>
43

```

- Model obsługujący zapytania do bazy danych w obszarze aukcji

```
mv > model > auctionModel.php
1 <?php
2
3 require_once 'databaseConnection.php';
4
5 class Auction {
6     private $db;
7
8     // Konstruktor obiektu
9     public function __construct() {
10         $this->db = Database::getInstance();
11     }
12
13     // Pobranie wszystkich aukcji
14     public function getAllAuction(){
15         $query = "SELECT * FROM auctions a
16             LEFT JOIN accounts u ON a.accountid = u.accountid
17             LEFT JOIN currency c ON a.currencyid = c.currencyid
18             WHERE a.selled = 0 AND a.verified = 1";
19         $stmt = $this->db->query($query);
20         $stmt->execute();
21         return $stmt;
22     }
23
24     // Pobranie aukcji do banneru
25     public function getBannerAuction(){
26         $query = "SELECT title, auctionid FROM auctions WHERE sold = 0 AND verified = 1 ORDER BY auctionid DESC LIMIT 4";
27         $stmt = $this->db->query($query);
28         return $stmt;
29     }
30
31     // Pobranie wszystkich aukcji aktualnych
32     public function getAllAuctionUser($accountId){
33         $query = "SELECT auctions.*, accounts.accountid, category.name AS category_name, auctions.verified
34             FROM auctions
35             INNER JOIN accounts ON auctions.accountid = accounts.accountid
36             LEFT JOIN category ON auctions.categoryid = category.categoryid
37             WHERE accounts.accountid = :accountId AND auctions.verified < 2 AND auctions.selled = 0";
38         $stmt = $this->db->prepare($query);
39         $stmt->bindParam(':accountId', $accountId);
40         $stmt->execute();
41         return $stmt;
42     }
43
44     // Pobranie wszystkich aukcji aktualnych
45     public function getAllAuctionUserInactive($accountId){
46         $query = "SELECT auctions.*, accounts.accountid, category.name AS category_name, auctions.verified
47             FROM auctions
48             INNER JOIN accounts ON auctions.accountid = accounts.accountid
```

- Model obsługujący połączenie do bazy danych

```
mv > model > databaseConnection.php
1 <?php
2
3 class Database {
4
5     private static $instance = null;
6
7     private $pdo;
8
9     private function __construct() {
10
11         $config = require __DIR__ . '/../config/config.php';
12
13         $dsn = "mysql:host={$config['host']};dbname={$config['dbname']};charset=utf8mb4";
14
15         try {
16             $this->pdo = new PDO($dsn, $config['user'], $config['password'], [
17                 PDO::ATTR_ERRMODE => PDO::ERRMODE_EXCEPTION,
18                 PDO::ATTR_DEFAULT_FETCH_MODE => PDO::FETCH_ASSOC,
19                 PDO::ATTR_EMULATE_PREPARES => false
20             ]);
21         } catch (PDOException $e) {
22             die("Błąd połączenia z bazą danych: " . $e->getMessage());
23         }
24     }
25
26
27
28     public static function getInstance() {
29         if (!self::$instance) {
30             self::$instance = new Database();
31         }
32         return self::$instance->pdo;
33     }
34
35 }
36
```

- Model obsługujący zapytania do bazy danych w obszarze związanym z użytkownikami

39

D. W widokach natomiast znajdują się pliki widziane bezpośrednio przez personę po wejściu na stronę www.

- Widok administratora

```
admin.php
mvc > view > administrators > admin.php
1 <?php
2
3 require_once __DIR__."/../main/header.php";
4 require_once __DIR__."/../model/administrativeModel.php";
5 require_once __DIR__."/../model/auctionModel.php";
6
7 $administratorPanel = new Administrative();
8 $accountId = $_SESSION['logged']['account_id'];
9
10
11 <ul class="nav nav-tabs" id="myTab" role="tablist">
12 <li class="nav-item" role="presentation">
13 <button class="nav-link active" id="auctiontab" data-bs-toggle="tab" data-bs-target="#auction" type="button" role="tab" aria-controls="auction" aria-selected="true">Aukcje
14 </li>
15 <li class="nav-item" role="presentation">
16 <button class="nav-link" id="newusertab" data-bs-toggle="tab" data-bs-target="#newuser" type="button" role="tab" aria-controls="newuser" aria-selected="false">Konta - weryfikacja
17 </li>
18 <li class="nav-item" role="presentation">
19 <button class="nav-link" id="activeusertab" data-bs-toggle="tab" data-bs-target="#activeuser" type="button" role="tab" aria-controls="activeuser" aria-selected="false">Konta - aktywacja
20 </li>
21 </ul>
22
23 <div class="tab-content" id="myTabContent">
24
25 <div class="tab-pane fade show active" id="auction" role="tabpanel" aria-labelledby="auction"><br>
26
27 <?php
28
29 // Pobranie wszystkich aukcji z podstawowymi informacjami
30 $auctions = $administratorPanel->getUnverifiedAuctions();
31 foreach ($auctions as $auction) {
32
33     echo <<< TABLELISTA
34     <div class="row box p-3">
35         <img class="aimg" src=__DIR__."/images/nofoto.jpg" />
36
37         <div class="box-text">
38
39             <div style="width: 50%; float: left; ">
40                 <h3>
41                     <a class="atitle link-dark" style="text-decoration: none;" href="auction.php?auction_id=$auction[auctionid]">$auction[title]</a>
42                 </h3>
43             </div>
44
45             <div style="overflow: hidden; text-align: right;">
46                 <h3>$auction[price]</h3>$auction[currency_name]<br>
47                 <a href="...">Zatwierdź</a> <a href="...">Odrzuć</a>
48             </div>
49         </div>
50     </div>
51 }
```

- Widoki aukcji (listy/dodawania/modyfikacji/listy aukcji użytkownika)

```
addauction.php
mvc > view > auctions > addauction.php
1 <?php
2
3 $config = include 'C:/xampp/htdocs/inf2/C2C/config/config.php';
4 require '../main/header.php';
5 require '../model/auctionModel.php';
6 $model = new Auction();
7
8 $used = false;
9 $private = true;
10
11
12 $account_id = $_SESSION['logged']['account_id'];
13
14
15
16 <form method="POST" action="...">
17 <table class="table">
18 <tbody>
19 <tr>
20 <td><label for="title">Tytuł:</label></td>
21 <td><input class="form-control" type="text" name="title" id="title" required</td>
22 </tr>
23 <tr>
24 <td><label for="categoryid">Kategoria:</label></td>
25 <td>
26 <select class="form-control" name="categoryid" id="categoryid" required>
27 <?php
28 $stmt = $model->getCategories();
29 $category1 = $stmt->fetchAll();
30 foreach ($category1 as $category){
31     echo "<option value='$category[categoryid]'">$category[name]</option>";
32 }
33 </select>
34 </td>
35 </tr>
36 <tr>
37 <td><label for="description">Opis:</label></td>
38 <td><textarea class="form-control" name="description" id="description" required</td>
39 </tr>
40 <tr>
41 <td><label class="form-check-label" for="used">Używany:</label></td>
42 <td><input class="form-check-input" type="checkbox" name="used" id="used" ></td>
43 </tr>
44 <tr>
45 <td colspan="2"><label for="price">Cena:</label></td>
46 </tr>
47 </tbody>
48 </table>
```



```

1 <?php
2 include_once "../main/header.php";
3 include_once "../model/auctionModel.php";
4 $model = new Auction();
5
6 try {
7
8     if(isset($_SESSION['info']) && $_SESSION['info'] <> null){
9
10         echo "<div class='alert alert-success' role='alert'>".$_SESSION['info']</div>";
11         $_SESSION['info']=null;
12     }
13
14     if (isset($_GET['auction_id'])) {
15         $auctionId = $_GET['auction_id'];
16
17         // Pobranie informacji o wybranej aukcji
18         $auction = $model->getAuctionById($auctionId);
19
20         if ($auction) {
21             ?>
22             <div class="row col-md-1">
23                 <button style="border: 0px;" onclick="goBack()">
24                     <svg xmlns="http://www.w3.org/2000/svg" width="50" height="50" fill="dark" class="bi bi-arrow-left-short" viewBox="0 0 16 16">
25                         <path fill-rule="evenodd" d="M12 8a.5.5 0 0 1-.5.5H5.707l2.146.5.5 0 1-.708.708l-3-3a.5.5 0 0 1 0-.708l3-3a.5.5 0 0 1 1-.708.708.707 7.5H11.5a.5.5
26                     </svg>
27                     <!-->> do widoku wszystkich aukcji-->
28                 </button>
29             </div>
30             <div class="row col-md-11">
31                 <div class="col-md-6">
32                     
33                 </div>
34
35                 <div class="p-3 mb-2 col-md-6" style="text-align: right;">
36
37                     <h4>
38                         <b>Sprzedający:</b> <?php echo $auction['login']; ?><br><br>
39                         <b>Telefon:</b> <?php echo ($_SESSION['role']=="guest" ? "Zaloguj się aby zobaczyć nr telefonu" : $auction['phone']); ?> <br><br>
40                     </h4>
41                     <a class="btn btn-secondary" <?php echo "href='../../messages/newmessage.php?auction_id=$auctionId'"; ?> >
42                         <svg xmlns="http://www.w3.org/2000/svg" width="16" height="16" fill="currentcolor" class="bi bi-send" viewBox="0 0 16 16">
43                             <path d="M15.854 14.589 11.541 5.819 14.547a.75.75 0 0 1 1.329 1.241 3.178 3.178 0 0 1 1.184.75.75 0 0 1 1.124 1.331 15.314 15.314 0 0 1 .541
44                     </svg> Napisz wiadomość </a>
45                     <br><br><br>
46                     <h2><?php echo $auction['price']." ".$auction['currency_name']; ?></h2>
47
48             </div>

```

```

1 <?php
2
3 if (isset($_SESSION['logged']) && $_SESSION['logged']) {
4     header('location: ../../../login.php');
5     exit();
6 }
7
8 try {
9
10     $accountId = $_SESSION['logged']['account_id'];
11
12     // Pobranie istniejących informacji o użytkowniku
13     $statement = $model->getUserById($accountId);
14     $user = $statement->fetch(PDO::FETCH_ASSOC);
15
16     if ($user) {
17
18         echo <<<TABLEUSER
19         <table class="table">
20             <thead>
21                 <tr>
22                     <th scope="col"><h2>Informacje o użytkowniku</h2></th>
23                     <th scope="col"><h2>Zmień dane użytkownika</h2></th>
24                 </tr>
25             </thead>
26
27             <tbody>
28                 <tr>
29                     <td>
30                         <b>Login:</b> $user[login]
31                         <br>
32                         <b>Typ konta:</b> $user[type_name]
33                         <br>
34                         <b>Zweryfikowany:</b> $user[verifiedtext]
35                         <br>
36                     </td>
37
38                     <td>
39                         <form method="POST" action="">
40                             <div class="form-row">
41                                 <label for="firstname">Imię:</label>
42                                 <input class="form-control" type="text" name="firstname" id="firstname" value="$user[firstname]" maxlength="50" placeholder="Max 50 znaków" required><br>
43
44                                 <label for="lastname">Nazwisko:</label>
45                                 <input class="form-control" type="text" name="lastname" id="lastname" value="$user[lastname]" maxlength="150" placeholder="Max 150 znaków" required><br>
46
47                                 <label for="email">Email:</label>
48                                 <input class="form-control" type="email" name="email" id="email" value="$user[email]" maxlength="250" placeholder="Wprowadź poprawny email" required><br>
49
50                                 <label for="phone">Telefon:</label>
51                                 <input class="form-control" type="tel" name="phone" id="phone" value="$user[phone]" pattern="[0-9]{9}" placeholder="Numer telefonu (9 cyfr)" required><br>
52

```

- Widoki związane z obsługą (logowania/rejestracji/przypominania hasła)

```

forgot-password.php X
mvc > view > main > forgot-password.php
20 <body class="hold-transition login-page">
21 <div class="login-box">
22 <div class="card card-outline card-secondary">
23 <div class="card-header text-center">
24 <a href="..." index.php" class="h1"><b>Studencki</b></a>
25 </div>
26 <div class="card-body">
27 <form action="..." controller/recovery.php" method="post">
28 <div class="input-group mb-3">
29 <input type="text" class="form-control" placeholder="Login" name="login" required>
30
31 <div class="input-group-append">
32 <div class="input-group-text">
33 <span class="fas fa-user"></span>
34 </div>
35 </div>
36
37 </div>
38 <div class="input-group mb-3">
39 <input type="email" class="form-control" placeholder="Email" name="email" required>
40 <div class="input-group-append">
41 <div class="input-group-text">
42 <span class="fas fa-envelope"></span>
43 </div>
44 </div>
45 </div>
46
47 <div class="input-group mb-3">
48 <input type="password" class="form-control" placeholder="Hasło" name="pass1" required>
49 <div class="input-group-append">
50 <div class="input-group-text">
51 <span class="fas fa-lock"></span>
52 </div>
53 </div>
54 </div>
55
56 <div class="input-group mb-3">
57 <input type="password" class="form-control" placeholder="Powtórz hasło" name="pass2" required>
58 <div class="input-group-append">
59 <div class="input-group-text">
60 <span class="fas fa-lock"></span>
61 </div>
62 </div>
63 </div>
64 <div class="row">
65 <div class="col-12">
66 <button type="submit" class="btn btn-secondary btn-block">Zmień hasło</button>
67 </div>

```

- Widoki związane z obsługą elementów stałych (header/stopka)

```

header.php X
mvc > view > main > header.php
16 <head>
17 <meta name="viewport"
18 <content="width=device-width, user-scalable=no, initial-scale=1.0, maximum-scale=1.0, minimum-scale=1.0">
19 <meta name="description"
20 <content="Strona stworzona w celach nauki programowania www. Studencki C2C to strona stworzona, aby przećwiczyć tworzenie kodu w HTML|PHP|JavaScript w powiązaniu z SQL.">
21 <link rel="icon"
22 <type="image/png" href="..." echo $config['baseurl']. '/images/icon.png'; ?>
23
24 <link href="https://cdn.jsdelivr.net/npm/bootstrap@5.3.0/dist/css/bootstrap.min.css" rel="stylesheet" integrity="sha384-9ndCyUa1bzAi12FUX018Cjncaps5m75np7ef0486qhlNu22cdeR0021uK"
25 <link rel="stylesheet" href="..." echo $config['baseurl']. '/public/css/style.css'; ?>
26
27 <title>Studencki C2C</title>
28 </head>
29
30
31
32
33
34 <div role="navigation" class="navbar navbar-expand-lg navbar-dark bg-dark">
35 <div class="container">
36 <div class="navbar-brand" href="..." echo $config['baseurl']. '/index.php'; ?> class="bi" width="40" height="40">
37 <button class="navbar-toggler" type="button" data-bs-toggle="collapse" data-bs-target="#menu" aria-controls="menu" aria-expanded="false" aria-label="Toggle navigation">
38 <span class="navbar-toggler-icon"></span>
39 </button>
40 <div class="collapse navbar-collapse text-secondary justify-content-end" id="menu">
41 <?php
42 <if ($SESSION["role"]=="guest"){
43 <?>
44 <ul class="navbar-nav">
45 <li class="nav-item">
46 <a href="..." echo 'mvc/view/main/login.php'; ?><button type="button" class="btn btn-outline-light me-2">Zaloguj</button></a>
47 </li>
48 <li class="nav-item">
49 <a href="..." echo 'mvc/view/main/register.php'; ?><button type="button" class="btn btn-light">Zarejestruj się</button></a>
50 </li>
51 </ul>
52
53 <?php
54 <elseif ($SESSION["role"]=="user"){
55 <?>
56 <ul class="navbar-nav">
57 <li class="nav-item">
58 <a class="nav-link" href="mvc/view/auctions/userauctions.php">Twoje aukcje</a>
59 </li>
60 <li class="nav-item">
61 <a class="nav-link" href="mvc/view/messages/messageList.php">Wiadomości</a>
62 </li>
63 <li class="nav-item">
64 <a class="nav-link" href="mvc/view/faq/faq.php">Pytania i odpowiedzi</a>
65 </li>
66 </ul>
67 </div>

```

```

terms.php X
mvc > view > main > terms.php
1 <?php
2 include_once "../header.php";
3 ?>
4 <div class="h1">Regulamin</div>
5 <br><br>
6 <pre><b>1. Definicje</b>
7
8 Cena dynamiczna - cena, której wysokość jest zmienna, w odpowiedzi na popyt rynkowy. Aktualna wysokość ceny dynamicznej jest dostępna podczas uzupełniania formularza Ogłoszenia lub pu
9 Cena końcowa - cena uwzględniająca podatek VAT i wszystkie pozostałe opłaty, za którą Przedmiot lub Usługa są oferowane dla Kupujących.
10 Dostawa - usługa przewozu Przedmiotów objętych Transakcją, świadczona przez profesjonalnych przewoźników wybranych za pośrednictwem usługi świadczonej przez (i) Alsendo sp. z o.o. all
11 Dzień - oznacza następujące kolejno po sobie 24 godziny.
12 Gość - osoba fizyczna korzystająca z Serwisu, nieposiadająca Konta lub do niego niezalogowana.
13 Grupa OLX - Grupa OLX sp. z o.o. z siedzibą w Poznaniu (ul. Królowej Jadwigi 43, 61-872, Poznań), wpisana do rejestru przedsiębiorców prowadzonego przez Sąd Rejonowy Poznań-Nowe M
14 Kandydat - Gość lub Użytkownik przeglądający Ogłoszenia w kategorii Praca lub korzystający z dedykowanych funkcjonalności określonych w Załączniku nr 1 "OLX Praca".
15 Kategoria - kategoria tematyczna (np. Motoryzacja czy Elektronika), do której przypisane jest Ogłoszenie. Kategoria może dzielić się na powiązane tematycznie podkategorie.
16 Kategorie Outletowe - szczególny rodzaj Kategorii, w których publikowane są Ogłoszenia dotyczące Produktów Outletowych oraz które zostały wyraźnie oznaczone w Serwisie nazwą "Outlet".
17 Konsument - Użytkownik będący osobą fizyczną, dokonujący w Serwisie czynności niezwiązanych bezpośrednio z jego działalnością gospodarczą lub zawodową.
18 Konto - zbiór danych powiązanych z danym Użytkownikiem, obejmujący informacje o jego aktywności w Serwisie, w tym informacje podawane przez Użytkownika w Serwisie. Zasady dotyczące K
19 Kupujący - Użytkownik dokonujący ze Sprzedawcą Transakcji z wykorzystaniem funkcjonalności Serwisu.
20 Limit - przewidziana dla wybranych Kategorii liczba bezpłatnych Ogłoszeń, które Użytkownik może opublikować w Serwisie w danym czasie. Limity w poszczególnych Kategoriach mogą być wył
21 Ogłoszenie - propozycja sprzedaży lub świadczenia Przedmiotu sporządzona przez Sprzedawcę lub zaproszenie do uczestnictwa w procesie rekrutacyjnym sporządzone przez Pracodawcę i o
22 Operator Płatności - podmiot świadczący usługi płatnicze na rzecz Użytkowników w ramach Płatności OLX:
23 1. Adyen N.V. zarejestrowany w Holandii, pod numerem 34259528 z siedzibą przy Simon Carmiggeltstraat 6-50, 1011 DJ, Amsterdam, Holandia, lub
24 2. PayU S.A. z siedzibą w Poznaniu (ul. Grunwaldzka 186, 60-166 Poznań), wpisaną do rejestru przedsiębiorców Krajowego Rejestru Sądowego prowadzonego przez Sąd Rejonowy Poznań-Nowe M
25 Operator Płatności obsługujący Transakcję z Przesyłką OLX i Płatnościami OLX zostanie określony w procesie zamówienia.
26 Płatności OLX - usługa świadczona przez Operatora Płatności, umożliwiająca Kupującym dokonywanie płatności za Przedmiot w ramach Serwisu oraz ewidencjonowanie środków pieniężnych pocl
27 Portal OLX - bezpłatna funkcjonalność Konta opisana w punkcie 9 Regulaminu, ułatwiająca dokonywanie płatności na rzecz Grupy OLX.
28 Pracodawca - Użytkownik publikujący Ogłoszenie w Kategorii Praca w Serwisie lub korzystający z dedykowanych funkcjonalności określonych w Załączniku nr 1 "OLX Praca".
29 Przedmiot - towary lub usługi objęte Ogłoszeniem; w ramach Ogłoszeń w Kategorii Praca za Przedmiot Ogłoszenia uważa się opis i wymagania dotyczące stanowiska objętego Ogłoszeniem Pra
30 Przesyłki OLX - usługa świadczona z wykorzystaniem funkcjonalności Serwisu, umożliwiająca zamówienie w Serwisie Dostawy Przedmiotów.
31 Regulamin - niniejszy Regulamin wraz z załącznikami, określający zasady korzystania z Serwisu. Aktualna wersja Regulaminu jest dostępna na stronie Serwisu w każdym czasie, w formie w
32 Rejestracja - proces utworzenia przez Użytkownika Konta, po podaniu danych Użytkownika, akceptacji Regulaminu i aktywacji Konta.
33 Serwis - prowadzona przez Grupę OLX internetowa platforma online dostępna w domenie olx.pl oraz w aplikacji mobilnej OLX.pl.
34 Sprzedawca - Użytkownik publikujący Ogłoszenie w Serwisie i dokonujący z Kupującym Transakcji.
35 Transakcja - Każda umowa zawierana pomiędzy Użytkownikami lub Użytkownikiem a Gościem, dotycząca Przedmiotu.
36 Usługi Dostawy OLX - odpłatne usługi Dostawy świadczone za pośrednictwem Grupy OLX na rzecz Użytkownika zgodnie z niniejszym Regulaminem, w tym Załącznikiem nr 9 "Usługa Dostawy OLX".
37 Usługi Odpłatne - odpłatne usługi świadczone przez Grupę OLX na rzecz Użytkownika, obejmujące m.in.: opublikowanie płatnego Ogłoszenia lub pakietu Ogłoszeń, Usługi Promowania, Usługi
38 Usługa Serwisowa - odpłatna usługa świadczona przez Grupę OLX polegająca na udostępnieniu Kupującemu możliwości skorzystania z fakultatywnych usług: Przesyłki OLX i Płatności OLX przy
39 Usługi Promowania - odpłatne usługi promowania Ogłoszeń określone w Załączniku nr 2 "Usługi Promowania".
40 Użytkownik - osoba fizyczna, osoba prawna lub jednostka organizacyjna nieposiadająca osobowości prawnej, której ustawa przyznaje zdolność prawną, korzystająca z Serwisu po zalogowaniu
41 Użytkownik Biznesowy - Użytkownik Serwisu będący osobą fizyczną, osobą prawną lub jednostką organizacyjną nieposiadającą osobowości prawnej, której ustawa przyznaje zdolność prawną, i
42 Wynagrodzenie za Usługę Serwisową - wynagrodzenie na rzecz Grupy OLX z tytułu udostępnienia Kupującemu przez Grupę OLX możliwości skorzystania z fakultatywnych usług: Przesyłki OLX i
43
44 <b>2. Postanowienia Ogólne</b>
45
46
47
48
49
50
51
52

```

- Widoki powiązane z wiadomościami

```

messageList.php X
mvc > view > messages > messageList.php
6 if (!isset($_SESSION['logged']) || $_SESSION['logged'] != null) {
7     header("Location: ../../index.php");
8 }
9
10 $account_id = $_SESSION['logged']['account_id'];
11
12 try {
13     //Zapytanie dla sprzedawanych przedmiotów (nagłówek)
14     $stmtSell = $model->getSellMessages($account_id);
15
16     // Zapytanie dla kupowanych (nagłówek)
17     $stmtBuy = $model->getBuyMessages($account_id);
18
19 } catch (PDOException $e) {
20     echo "Błąd połączenia lub aktualizacji bazy danych: " . $e->getMessage();
21 }
22
23
24
25
26 <div class="nav nav-tabs" id="myTab" role="tablist">
27 <li class="nav-item" role="presentation">
28 <button class="nav-link active" id="messageBuy-tab" data-bs-toggle="tab" data-bs-target="#buy" type="button" role="tab" aria-controls="buy" aria-selected="true">Kupuj</li>
29 </li>
30 <li class="nav-item" role="presentation">
31 <button class="nav-link" id="messageSell-tab" data-bs-toggle="tab" data-bs-target="#sell" type="button" role="tab" aria-controls="sell" aria-selected="false">Sprzedaj</li>
32 </li>
33 </div>
34
35 <div class="tab-content" id="myTabContent">
36
37 <div class="tab-pane fade show active" id="buy" role="tabpanel" aria-labelledby="messageBuy-tab">
38
39 <div class="row">
40 <div class="col">
41 <div class="box">
42 <div class="box-text">
43 <div style="width: 50%; float: left; ">
44 <div style="width: 50%; float: right; ">
45 <div style="width: 50%; float: left; ">
46 <div style="width: 50%; float: right; ">
47 <div style="width: 50%; float: left; ">
48 <div style="width: 50%; float: right; ">
49 <div style="width: 50%; float: left; ">
50 <div style="width: 50%; float: right; ">
51 <div style="width: 50%; float: left; ">
52 <div style="width: 50%; float: right; ">

```

C2. Efekty realizacji projektu

Do efektów realizacji projektu należy:

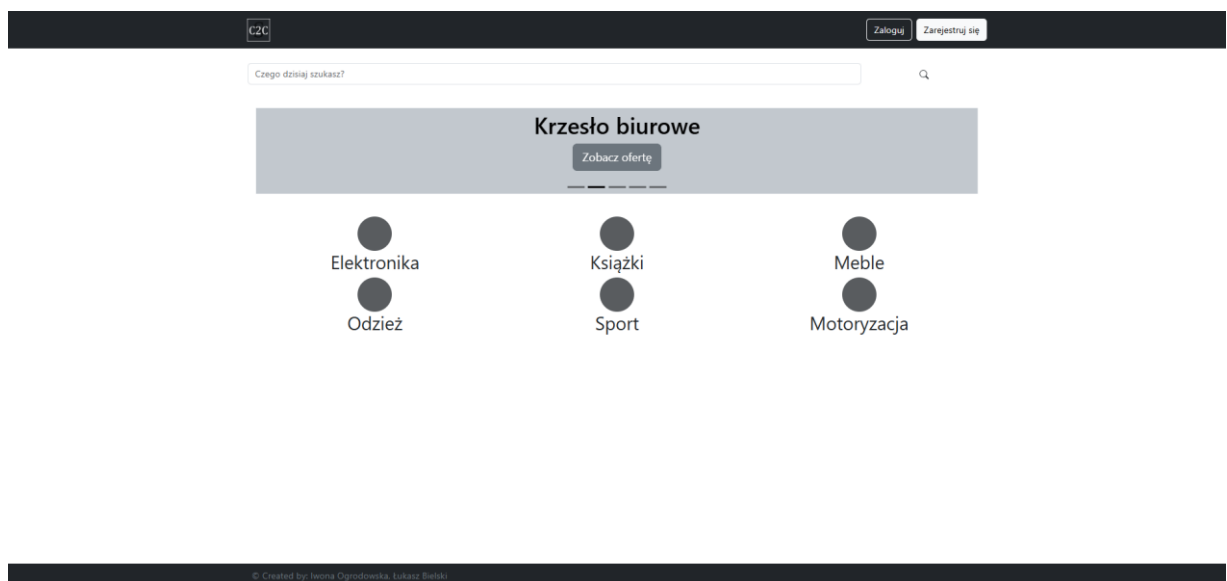
Dostrzeżenie roli systemów C2C na postawy konsumenckie, czyli mechanizmów wpływających na zmianę nawyków konsumpcyjnych, poprzez porównanie podejścia do zakupu nowych produktów z ideą ponownego wykorzystania oraz wymiany używanych dóbr.

Kompleksowa koncepcja funkcjonalności platformy C2C, dotycząca opracowania szczegółowego modelu funkcjonalnego, który integruje rozwiązania technologiczne, społeczno-ekologiczne i użytkowe, umożliwiające promowanie odpowiedzialnej i zrównoważonej konsumpcji.

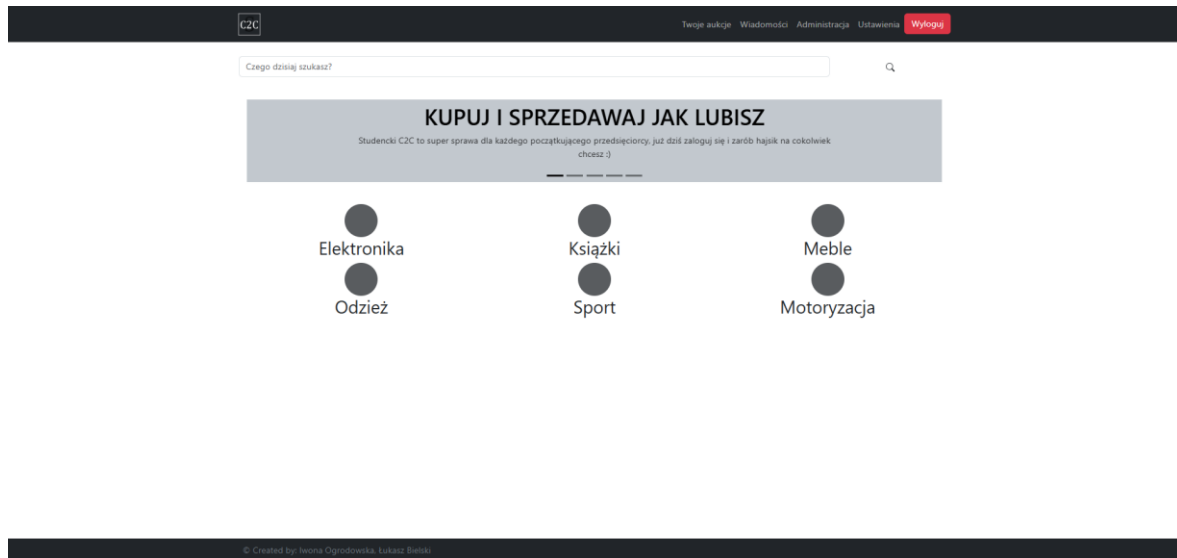
Identyfikacja kluczowych cech i funkcji platformy, polegająca na wyodrębnieniu najważniejszych funkcjonalności, które platforma powinna posiadać, aby skutecznie wspierać ideę zrównoważonej konsumpcji, m.in. mechanizmy weryfikacji autentyczności ofert i użytkowników.

Zaprojektowanie interfejsu użytkownika sprzyjającego intuicyjnemu korzystaniu z platformy:

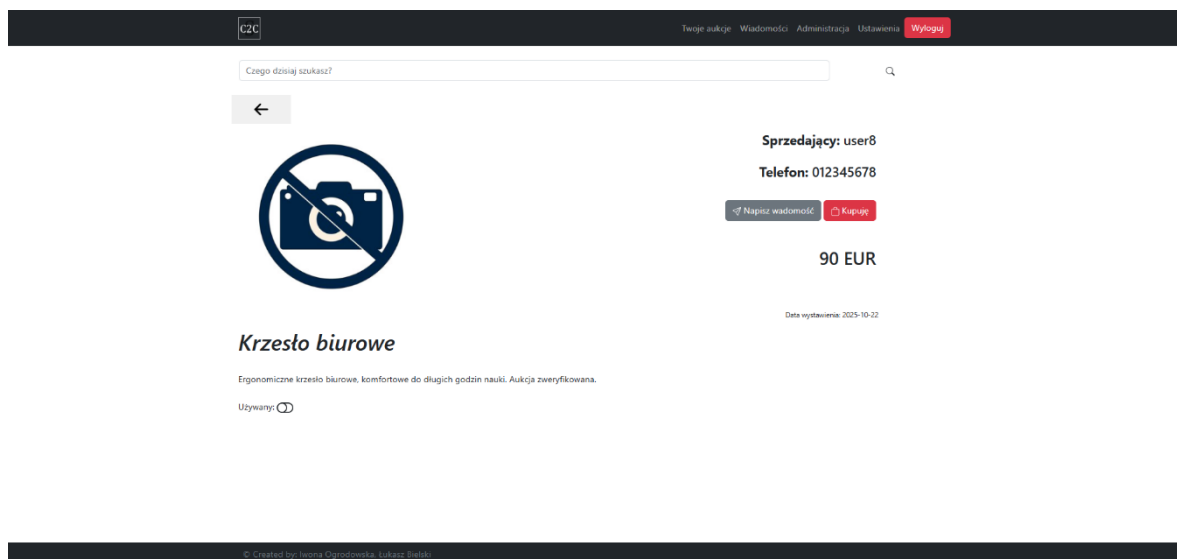
- Widok główny



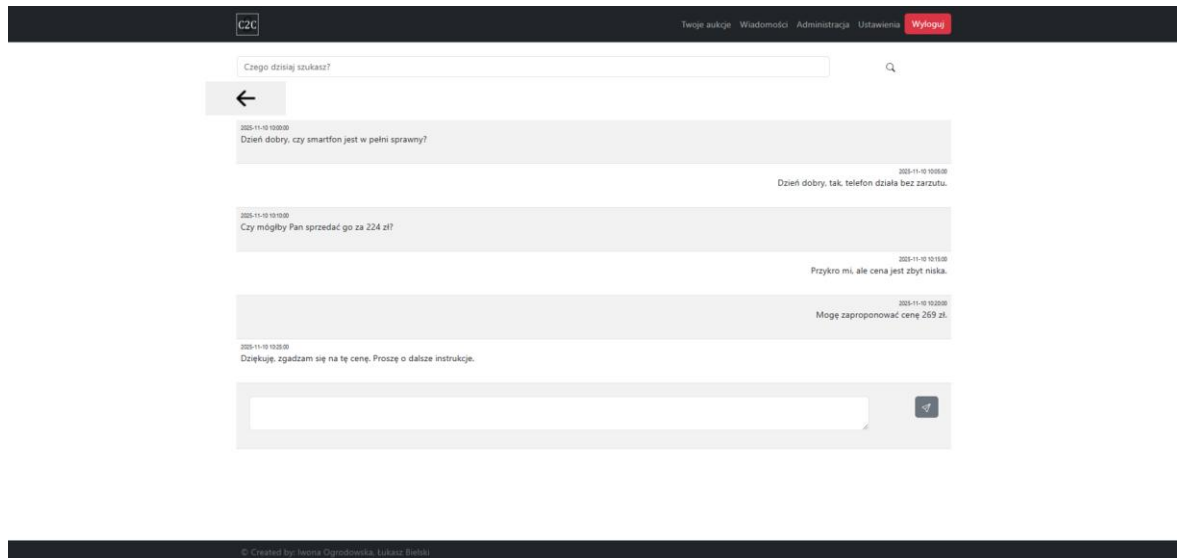
- Widok główny Administratora



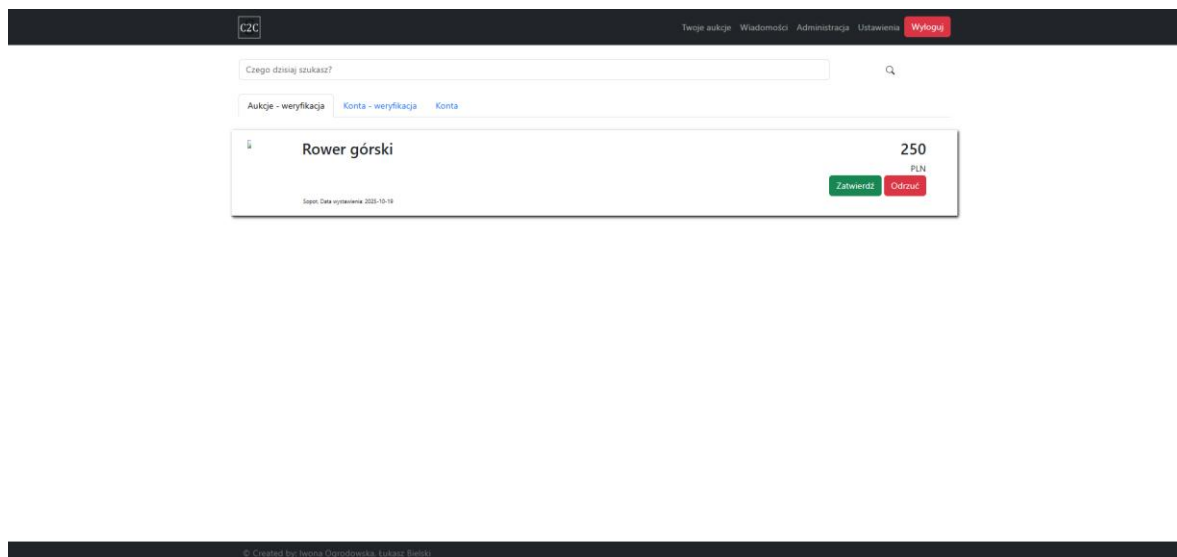
- Widok oferty (zalogowanego użytkownika)



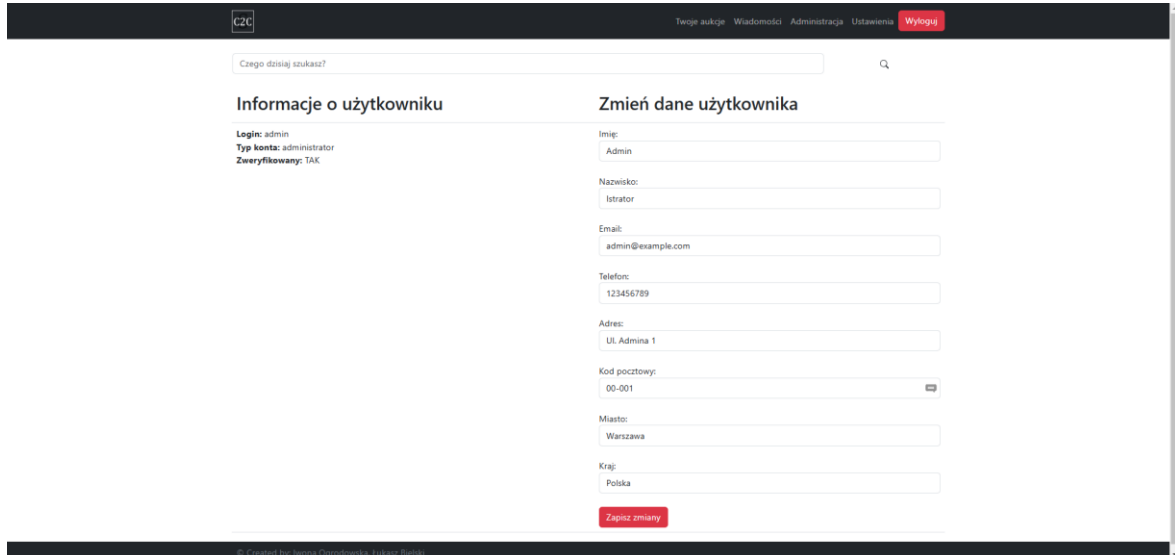
- Widok wymiany wiadomości między sprzedającym a kupującym



- Widok Administratora do autoryzacji ofert

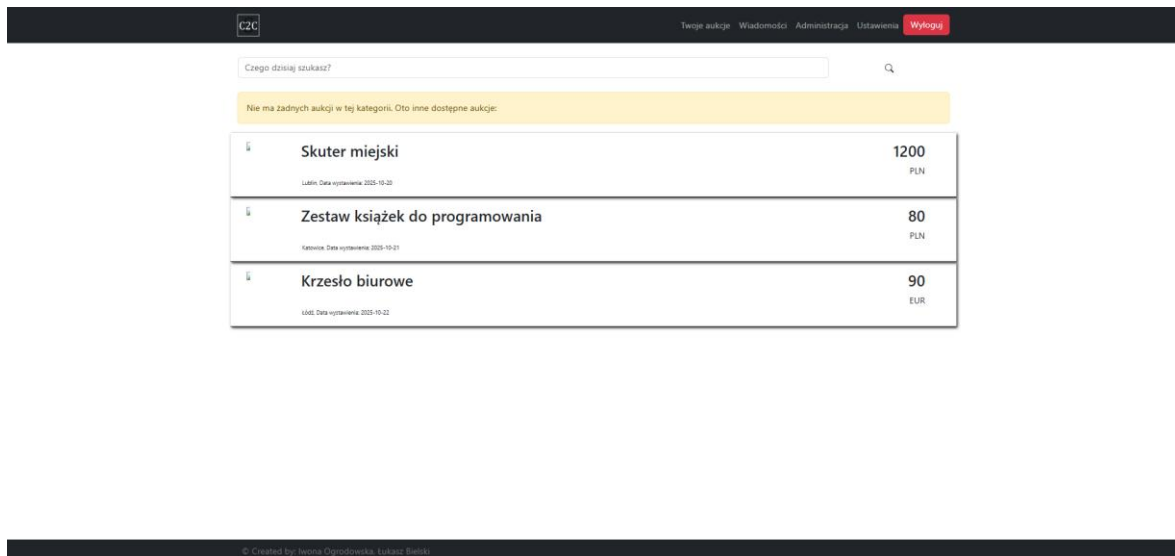


- Widok edycji konta użytkownika






The screenshot shows the 'Zmień dane użytkownika' (Edit user data) page. It features a search bar at the top and a navigation menu with links to 'Teraz aukcje', 'Wiadomości', 'Administracja', 'Ustawienia', and a red 'Wyloguj' button. The main content area is divided into two columns. The left column, titled 'Informacje o użytkowniku', displays user details: 'Login: admin', 'Typ konta: administrator', and 'Zweryfikowany: TAK'. The right column, titled 'Zmień dane użytkownika', contains a form with fields for 'Imię' (Admin), 'Nazwisko' (Istrator), 'Email' (admin@example.com), 'Telefon' (123456789), 'Adres' (Ul. Admina 1), 'Kod pocztowy' (00-001), 'Miasto' (Warszawa), and 'Kraj' (Polska). A red 'Zapisz zmiany' button is at the bottom right of the form. A footer note at the bottom of the page reads: '© Created by: Iwona Ogrodowska, Lukasz Bielicki'.

- Widok kategorii, w których nie ma ofert

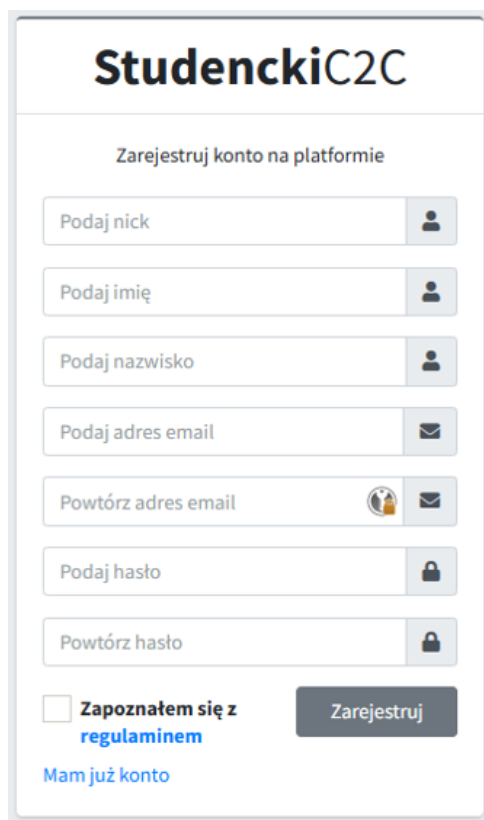


The screenshot shows the category view page. It features a search bar at the top and a navigation menu with links to 'Teraz aukcje', 'Wiadomości', 'Administracja', 'Ustawienia', and a red 'Wyloguj' button. A yellow banner message states: 'Nie ma żadnych aukcji w tej kategorii. Oto inne dostępne aukcje:'. Below this, a table lists three items:

	Skuter miejski <small>Łódź, Data wystawienia: 2025-10-20</small>	1200 PLN
	Zestaw książek do programowania <small>Katowice, Data wystawienia: 2025-10-21</small>	80 PLN
	Krzesło biurowe <small>Włocławek, Data wystawienia: 2025-10-22</small>	90 EUR


A footer note at the bottom of the page reads: '© Created by: Iwona Ogrodowska, Lukasz Bielicki'.


- Ekran rejestracji nowego konta





StudenckiC2C



Zarejestruj konto na platformie


Podaj nick 


Podaj imię 

Podaj nazwisko 

Podaj adres email 

Powtórz adres email  

Podaj hasło 

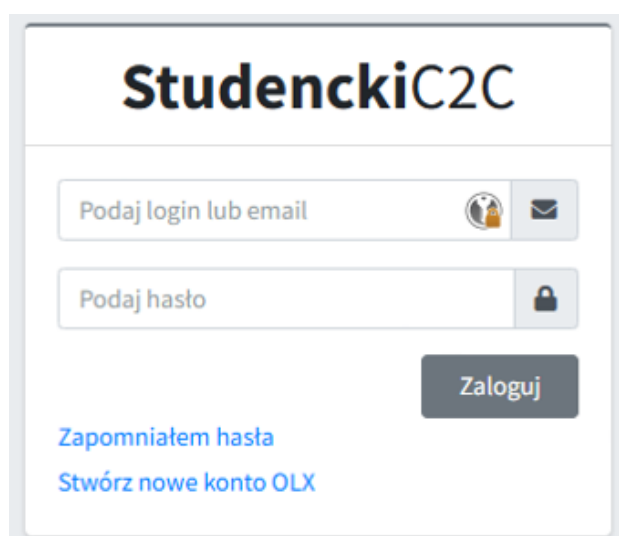
Powtórz hasło 

☐ Zapoznałem się z [regulaminem](#)



[Mam już konto](#)


Zarejestruj

- Ekran logowania



StudenckiC2C

Podaj login lub email  

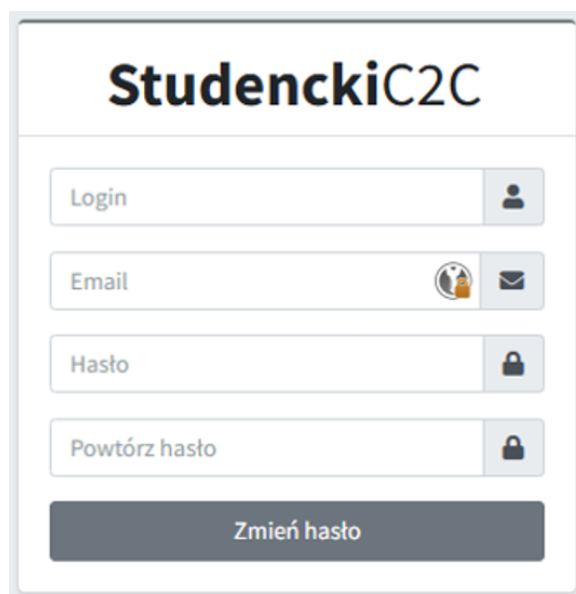
Podaj hasło 

Zaloguj

[Zapomniałem hasła](#)

[Stwórz nowe konto OLX](#)

- Ekran odzyskiwania hasła



StudenckiC2C

Login

Email

Hasło

Powtórz hasło

Zmień hasło

C3. Użyteczność projektu

Projekt ma potencjał, aby realnie wpłynąć na zmianę postaw konsumpcyjnych, promując ideę zrównoważonego rozwoju oraz umożliwiając użytkownikom świadomy wybór pomiędzy kupowaniem nowych a wykorzystywaniem używanych dóbr. Dzięki integracji technologii, edukacji i społecznych inicjatyw, platforma może stać się istotnym narzędziem w walce z nadmiernym konsumpcjonizmem. Obecna forma projektu ukierunkowana jest na grupę docelową jaką stanowią studenci. Użytkownicy mogą korzystać z platformy do zakupu lub wymiany przedmiotów, co przyczynia się do oszczędności finansowych w tej grupie docelowej. Dzięki informacjom i funkcjonalnościom edukacyjnym, platforma wspiera zmianę nawyków konsumpcyjnych powodując wzrost świadomości ekologicznej.

Projekt może stać się elementem strategii CSR (Społeczna Odpowiedzialność Biznesu), w której przedsiębiorstwa podejmują działania wykraczające poza czysto ekonomiczne cele, dbając jednocześnie o dobro społeczne i środowisko naturalne a także wsparciem dla inicjatyw proekologicznych i gospodarki o obiegu zamkniętym. Projekt może być adaptowany i rozwijany na różnych rynkach, przyczyniając się do globalnej zmiany modeli konsumpcji, nie tylko wśród studentów.

Propozycja strategii wdrożenia i popularyzacji obejmuje: Etap I. Testy i Walidacja: Przeprowadzenie pilotażowego wdrożenia platformy na ograniczonej grupie

użytkowników (studentów) w celu zebrania opinii i weryfikacji funkcjonalności. Testowanie kluczowych modułów (mechanizmów weryfikacji autentyczności ofert i użytkowników) oraz analiza wpływu na postawy konsumenckie.

Etap II. Wdrożenie i Rozwój: Skalowanie rozwiązania na szerszy rynek z jednoczesnym monitorowaniem i optymalizacją funkcji platformy. Wprowadzenie systemu feedbacku umożliwiającego ciągłe doskonalenie oferty i dostosowywanie funkcjonalności do zmieniających się potrzeb użytkowników.

Etap III. Kampania Promocyjna: Opracowanie i realizacja strategii marketingowej, w tym działań w mediach społecznościowych. Współpraca z organizacjami pozarządowymi, instytucjami edukacyjnymi i sektorem publicznym w celu popularyzacji idei platformy.

Etap IV. Monitorowanie Efektów: Regularna analiza wpływu platformy na zachowania konsumentów i konsumpcjonizm. Regularne aktualizacje platformy (dostosowywanie technologii do aktualnych trendów).

C4. Autoewaluacja zespołu projektowego

Iwona Ogrodowska

Współtworząc projekt byłam odpowiedzialna za opracowanie koncepcji funkcjonalności platformy oraz strategii wdrożenia innowacyjnych rozwiązań. Skupiłam się na aspektach technologicznych. Określiłam na podstawie literatury wpływ platform C2C na postawy konsumentów oraz ich podejście do kupowania nowych produktów w porównaniu do wykorzystywania przedmiotów używanych lub wymiany. Dokonałam weryfikacji dostępnych platform internetowych opartych na systemie sprzedażowym C2C.

Zdobyte przeze mnie umiejętności i kompetencje to rozwój umiejętności projektowania systemów informatycznych z uwzględnieniem aspektów ekologicznych oraz zdobycie doświadczenia w tworzeniu strategii implementacji technologicznych rozwiązań, a przede wszystkim praktycznego wykorzystania programowania obiektowego (OOP).

Łukasz Bielski

Współtworząc projekt byłem odpowiedzialny za przegląd literatury i wybór najnowszych doniesień literaturowych w Polsce i na świecie, określenie na podstawie literatury wpływu platform C2C na postawy konsumentów oraz ich podejście do kupowania nowych produktów w porównaniu do wykorzystywania przedmiotów

używanych lub wymiany, przygotowanie bazy danych i opracowaniu użyteczności projektu.

Zdobyte przeze mnie umiejętności i kompetencje to rozwinięcie kompetencji analitycznych w zakresie weryfikacji trendów konsumenckich w kontekście tworzenia nowych technologii. Umiejętność łączenia aspektów społecznych z technologicznymi. Przede wszystkim rozwój umiejętności projektowania systemów informatycznych z wykorzystaniem MySQL – systemu zarządzania relacyjnymi bazami danych.

C5. Wykorzystane materiały i bibliografia związana z realizacją projektu

1. Jastrzębska, E. (2019). Konsument w gospodarce o obiegu zamkniętym. *Studia i Prace Kolegium Zarządzania i Finansów*, (172), 53-69.
2. Kos-Łabędowicz, J. (2015). Aukcje internetowe jako źródło informacji o produktach. *Studia Ekonomiczne*, 232, 103-112.
3. Kawecki, T. (2012). Handel C2C-czy to tylko aukcje? *E-mentor*, 44(2), 61-64.
4. Rilfi, M. R. M. (2021). Product attribute extraction from C2C social media messages.
5. Sikora, M., Dzieża, G., & Lwowski, Z. (2011). Ograniczenia w realizacji transakcji internetowych. *Studia i Materiały Polskiego Stowarzyszenia Zarządzania Wiedzą*, 49, 157-171.
6. Pangsy-Kania, S. (2024). Czy za pieniądze można kupić szczęście? – Konsumpcjonizm jako przyczyna kryzysów. *Optimum. Economic Studies*, 116(2), 20-42.
7. Dąbrowska, A. (2015). Postawy polskich konsumentów – od konsumpcjonizmu do zrównoważonej konsumpcji. *Handel Wewnętrzny*, 355(2), 88-100.
8. Cebulska, A., Pieczarka, K. (2024). Znaczenie aspektów technicznych, funkcjonalnych i użytkowych platform sprzedażowych. *Wyższej Szkoły Zarządzania i Przedsiębiorczości z siedzibą w Wałbrzychu*, 5.
9. Leloch, N., & Kowalska, M. Zrównoważony rozwój e-commerce w dobie przemian społeczno-gospodarczych. *Wybrane Problemy Logistyki I Usług Turystycznych Analiza Sektorowa, Zrównoważony Rozwój, Rachunkowość*, 101.
10. Wójcik, A., Wolski, M., & Smółka, J. B. (2019). Analiza wydajnościowa frameworka Symfony do tworzenia nowoczesnych aplikacji webowych na podstawie wybranych wersji. *Journal of Computer Sciences Institute*, 13, 293-297.

11. Welling, L., & Thomson, L. (2003). PHP and MySQL Web development. Sams publishing.
12. brandly360.com
13. <https://biznes.olx.pl/blog>
14. Todryk, G. (2014). Architektura aplikacji internetowych. Zeszyty Naukowe WSEI seria: Transport i Informatyka, 4(1).
15. Jabłońska, P. (2018). Tworzenie modułowych aplikacji JavaScript–porównanie rozwiązania otwartego i komercyjnego. Journal of Computer Sciences Institute, 7.