

POLITECHNIKA WROCŁAWSKA
WYDZIAŁ ELEKTRONIKI

KIERUNEK: INFORMATYKA
SPECJALNOŚĆ: SYSTEMY I SIECI KOMPUTEROWE

PRACA DYPLOMOWA
INŻYNIERSKA

Projekt i implementacja aplikacji webowej
umożliwiającej sprzedaż i wymianę książek

Design and implementation of a web application
allowing the sale and exchange of books

AUTOR:

Jakub Sobieszek

PROWADZĄCY PRACĘ:

Dr inż. Przemysław Ryba, K₃₂Wo₄Do₃

Spis treści

1. Wstęp	7
2. Specyfikacja projektu	8
2.1. Przegląd istniejących rozwiązań	8
2.2. Opis zadania projektowego	9
2.2.1. Opis działania i funkcji systemu	9
2.2.2. Założenia architektoniczne przyjęte podczas realizacji systemu	9
2.3. Wymagania Funkcjonalne	9
2.3.1. Funkcje dostępne dla użytkownika	9
2.3.2. Funkcje dostępne dla administratora	9
2.4. Wymagania niefunkcjonalne	10
2.5. Technologia i narzędzia	10
2.5.1. Język programowania	10
2.5.2. Framework sieciowy	10
3. Projekt aplikacji	12
3.1. Plan aplikacji	12
3.2. Diagramy aplikacji	12
3.2.1. ERD	12
3.2.2. Diagram tabel bazy danych	14
3.2.3. Diagram przypadków użycia	15
3.3. Scenariusze przypadków użycia	16
3.3.1. Dodanie tematu dyskusyjnego	16
3.3.2. Dodanie ogłoszenia transakcyjnego	16
3.3.3. Dodanie odpowiedzi w temacie dyskusyjnym	16
3.3.4. Przeglądanie profilu użytkownika	17
3.3.5. Wysłanie wiadomości prywatnej (rozpoczęcie konwersacji)	17
3.3.6. Edycja awatara	17
3.3.7. Udzielenie odpowiedzi w prywatnej konwersacji	18
3.3.8. Usunięcie konta użytkownika	18
3.3.9. Nadanie/Odebranie praw administracyjnych	18
3.3.10. Dodanie działu tematów dyskusyjnych	19
3.3.11. Usunięcie działu tematów dyskusyjnych	19
3.3.12. Usunięcie tematu dyskusyjnego	19
3.3.13. Usunięcie odpowiedzi Użytkownika w temacie dyskusyjnym	20

4. Aplikacja	21
4.1. Prezentacja serwisu	21
4.1.1. Strona główna	21
4.1.2. Rejestracja	21
4.1.3. Logowanie	22
4.1.4. Tematy dyskusyjne	23
4.1.5. Profil użytkownika i konwersacje	26
4.1.6. Ogłoszenia	29
4.1.7. Wyszukiwanie	31
4.2. Realizacja	32
4.2.1. Rejestracja użytkownika	32
4.2.2. Temat dyskusyjny	34
5. Testy Aplikacji	36
5.1. Plan testów	36
5.2. Przypadki testowe	36
5.2.1. Wątki dyskusyjne	36
5.2.2. Profil Użytkownika	36
5.2.3. Wiadomości	37
5.2.4. Ogłoszenia	37
5.2.5. Wyszukiwanie	37
5.3. Wyniki	37
6. Podsumowanie	38
Literatura	39

Spis rysunków

3.1. Diagram związków encji	13
3.2. Diagram tabel bazy danych aplikacji	14
3.3. Diagram przypadków użycia	15
4.1. Strona główna serwisu.	21
4.2. Strona rejestracji.	22
4.3. Użytkownik wprowadził hasło nie spełniające wymogów.	22
4.4. Formularz logowania	23
4.5. Strona główna z nowymi pozycjami górnej belki	23
4.6. Wybór jednego z działów	24
4.7. Zawartość działu	24
4.8. Utworzenie nowego tematu	24
4.9. Dyskusja w istniejącym temacie	25
4.10. Komentarz widoczny na stronie po dodaniu	26
4.11. Strona profilu użytkownika	26
4.12. Formularz wysłania pierwszej wiadomości (i rozpoczęcia konwersacji)	27
4.13. Strona rozmowy	27
4.14. Ikona panelu konwersacji	27
4.15. Panel konwersacji	28
4.16. Zmiana awatara	28
4.17. Wybór nowego zdjęcia	29
4.18. Nowy awatar	29
4.19. Sekcja ogłoszeń	29
4.20. Ogłoszenie innego użytkownika	30
4.21. Tworzenie nowego ogłoszenia	30
4.22. Własne ogłoszenie	31
4.23. Formularz wyszukiwania	31
4.24. Wyniki	32

Spis listingów

4.1.	Widok rejestracji nowego użytkownika	32
4.2.	Formularz rejestracji użytkownika	33
4.3.	Formularz przekazany do szablonu	33
4.4.	Mapowanie widoku do adresu url serwisu	34
4.5.	Widok funkcyjny tematu dyskusyjnego	34
4.6.	Część szablonu tematu dyskusyjnego wyświetlająca komentarze oraz pole na dodanie nowego komentarza	35

Skróty

UML (ang. *Unified Modeling Language*)

XML (ang. *eXtensible Markup Language*)

IT (ang. *Information Technology*)

WWW (ang. *World Wide Web*)

CRUD (ang. *Create, Read, Update, Delete*)

HTML (ang. *HyperText Markup Language*)

CSS (ang. *Cascading Style Sheets*)

Rozdział 1

Wstęp

Celem pracy jest projekt i implementacja aplikacji sieciowej służącej do sprzedaży, zakupu i wymiany książek, oraz prowadzenia dyskusji w temacie książek i czytelnictwa. Autor zamierza wykorzystać umiejętności nabyte podczas studiów, opanować nowe zagadnienia, a następnie połączyć je ze sobą tak, aby zbudować w pełni funkcjonalny serwis sieciowy. Do budowy aplikacji autor zamierza wykorzystać umiejętności z zakresu tworzenia i pracy z bazą danych, utworzyć serwer komunikujący się z bazą danych, oraz stronami WWW[5], które tworzyć będą interfejs użytkownika. W tym dokumencie opisana zostanie specyfikacja tworzonego serwisu, jego projekt, oraz omówienie gotowej aplikacji razem ze szczegółami jej implementacji.

Drugi rozdział poświęcony jest specyfikacji projektu. Zawiera on przegląd istniejących rozwiązań, założenia przyjęte przy tworzeniu serwisu, wymagania dotyczące aplikacji, oraz opis wykorzystanych narzędzi i technologii. W rozdziale trzecim przedstawiono projekt aplikacji zawierający plan logiczny tworzonego serwisu, diagramy, oraz opisy przypadków użycia. Rozdział czwarty zawiera prezentację i opis interfejsu aplikacji, oraz listingi kodu z opisem realizacji poszczególnych funkcjonalności, dające wgląd w sposób pracy z wykorzystanymi narzędziami. Rozdział piąty to opis testów aplikacji. Przedstawiony jest w nim plan testowania poszczególnych elementów serwisu, oraz wyniki testów. W rozdziale szóstym znajduje się podsumowanie wykonanej pracy.

Rozdział 2

Specyfikacja projektu

2.1. Przegląd istniejących rozwiązań

Jako pierwszy krok w planowaniu i projektowaniu systemu, postanowiono przejrzeć istniejące w sieci podobne projekty.

- <http://www.forumksiazki.com/>
Forum w języku polskim poświęcone miłośnikom czytania. Oferuje możliwość prowadzenia dyskusji w tematach podzielonych na działy i podziały. Strona główna jest czytelna i intuicyjna. Potencjalny użytkownik nie powinien mieć problemu z poruszaniem się po portalu. Forum nie oferuje wbudowanej funkcji ogłoszeń dotyczących sprzedaży czy kupna.
- <https://forum.mp3store.pl/>
Kolejna rodzima strona, tym razem w tematyce sprzętu audio. Układ głównej strony podobny do poprzedniej pozycji. Portal posiada komis, gdzie użytkownicy mogą umieszczać ogłoszenia kupna i sprzedaży, oraz wystawiać sobie komentarze pozwalające ocenić uczciwość i sumienność danej osoby w realizacji transakcji.
- <https://forums.onlinebookclub.org/>
Anglojęzyczne forum. Mimo podobieństwa strony głównej do poprzednich przykładów nie jest ona czytelna wg. autora. Powodem tego jest wyświetlanie zbyt dużej ilości zawartości w poszczególnych działach. Skutkuje to efektem “ściany tekstu” zmuszając użytkownika do poświęcenia większej uwagi w celu znalezienia interesującej zawartości
- <https://www.reading-forum.co.uk/forum/>
Kolejny obcojęzyczny portal cierpiący na ten sam problem co onlinebookclub. Dodatkowym utrudnieniem dla użytkownika są m.in. nazwy działów napisane mniejszą czcionką niż tematy znajdujące się w nich, oraz przedłużenie strony sekcjami nie związanymi bezpośrednio z tematyką forum np. organizacji charytatywnych, czy tematów z czytelnictwem niepowiązanych.

Bazując na podanych przykładach autor obrał za cel utworzenie portalu posiadającego następujące cechy:

- Możliwość tworzenia i udzielania się w tematach dyskusyjnych przez użytkowników
- System prywatnych wiadomości do komunikacji między użytkownikami
- Dedykowany dział transakcyjny

- Podział zawartości na działy i tematy dla sprawnego poruszania się po stronie
- Czytelny układ strony głównej, pozwalający na łatwe identyfikowanie jej elementów
- Intuicyjny interfejs zbudowany tak, aby użytkownik nie potrzebował żadnej pomocy w nawigacji i interakcji z portalem.

2.2. Opis zadania projektowego

2.2.1. Opis działania i funkcji systemu

Aplikacja umożliwiać będzie użytkownikom wystawianie ogłoszeń, sprzedaż, kupno i wymianę książek za pomocą interfejsu strony WWW[5]. System oparty będzie o relacyjną bazę danych. Za pomocą interfejsu, użytkownicy będą mogli zamieszczać ogłoszenia kupna, sprzedaży lub wymiany książek, komunikować się poprzez prywatne wiadomości, zakładać tematy dyskusyjne w odpowiednich działach, oraz udzielać się w nich. Administrator za pomocą wydzielonej dla niego sekcji interfejsu (nieдоступnej dla użytkowników) będzie miał możliwość dodawania/usuwania działów i tematów, modyfikowania treści znajdujących się na forum, oraz modyfikowania uprawnień użytkowników.

2.2.2. Założenia architektoniczne przyjęte podczas realizacji systemu

System zostanie zrealizowany jako serwer komunikujący się z bazą danych, oraz serwisem WWW[5]. Wykorzystana zostanie baza danych SQLite[3]. Serwer napisany zostanie w języku Python[1] z użyciem frameworku sieciowego Django[2]. Interfejs sieciowy zbudowany będzie z użyciem stron HTML[5], stylowania CSS[5] oraz frameworku front-end Bootstrap[4]. Dodatkowo wykorzystane zostaną narzędzia Django[2] przeznaczone do sprawnego komunikacji serwera ze stronami HTML[5], oraz ich wygodnej modyfikacji. Interfejs klienta służyć będzie tylko do prezentacji danych i komunikacji z użytkownikiem, wszystkie operacje na danych realizowane będą po stronie serwera. Przy tworzeniu specyfikacji użyty zostanie UML.

2.3. Wymagania Funkcjonalne

2.3.1. Funkcje dostępne dla użytkownika

- Założenie konta
- Zalogowanie
- Utworzenie tematu dyskusyjnego
- Dodanie komentarza w dyskusji
- Utworzenie ogłoszenia transakcyjnego (sprzedaży/kupna/zamiany)
- Wysłanie wiadomości innemu użytkownikowi (rozpoczęcie nowej konwersacji)
 - bezpośrednio z poziomu profilu adresata
 - kontakt z użytkownikiem poprzez przekierowanie z wystawionego przez niego ogłoszenia
- Wysłanie odpowiedzi w istniejącej prywatnej konwersacji
- Przeszukiwanie forum pod kątem użytkowników, tematów dyskusyjnych, ogłoszeń, komentarzy

2.3.2. Funkcje dostępne dla administratora

- Dodanie/usunięcie działu na tematy dyskusyjne
- Usunięcie tematu dyskusyjnego, ogłoszenia transakcyjnego

- Usunięcie odpowiedzi użytkownika w temacie dyskusyjnym
- Usunięcie konta użytkownika
- Nadanie/odebranie innemu użytkownikowi praw administratorskich
- Wszystkie operacje dostępne dla zwykłego użytkownika

2.4. Wymagania niefunkcjonalne

- Baza danych SQLite[3]
- Serwer w języku Python[1] z wykorzystaniem frameworku Django[2]
- Interfejs użytkownika zrealizowany za pomocą stron WWW[5]
- Strony WWW[5] zrealizowane poprzez HTML[5]
- Stylowanie stron wykorzystywać będzie CSS[5] oraz Bootstrap[4]
- Wymagane połączenie z internetem

2.5. Technologia i narzędzia

2.5.1. Język programowania

Wybrany językiem programowania jest Python[1]. Jest to język wysokiego poziomu, umożliwiający pisanie kodu m.in. w paradygmacie imperatywnym i obiektowym. Posiada dużą liczbę wbudowanych bibliotek. Charakteryzuje się czytelnością i klarownością składni.

2.5.2. Framework sieciowy

Większość narzędzi do utworzenia aplikacji sieciowej dostarczy framework Django[2]. Utworzony w 2003 roku, Django[2] cechuje się dużą ilością gotowych do użytku funkcjonalności, wysokim poziomem bezpieczeństwa, oraz dużą skalowalnością. Z wykorzystaniem Django[2] utworzone zostały internetowe serwisy Instagram, Pinterest, czy The Washington Times. Posiada wsparcie dla zabezpieczeń przed wieloma popularnymi w sieci zagrożeniami takimi jak: `cross-site request forgery`, `SQL injection`, czy `cross-site scripting`. Dodatkowo posiada wbudowane narzędzia poświęcone szyfrowaniu haseł i autoryzacji użytkownika. Większość dostarczanych funkcjonalności jest konfigurowalna. Użytkownik otrzymuje więc kompletny i gotowy do pracy zestaw narzędzi. Wymagający minimalnej konfiguracji wstępnej, ale umożliwiający dostosowanie do indywidualnych potrzeb, jeśli domyślne ustawienia okażą się niewystarczające.

Django[2] korzysta z architektonicznego wzorca `model-template-view`

- Model - sekcja danych. Modele to klasy pythonowe, tłumaczone za pomocą ORM na tabele w bazie danych. Domyślną bazą danych jest SQLite[3], możliwa jest jednak współpraca z MySQL, PostgreSQL, oraz Oracle. Dzięki takiemu rozwiązaniu przy pisaniu aplikacji korzysta się z modeli prawie tak, jak z tradycyjnych klas, a komponent odpowiedzialny za ORM realizuje odpowiednie operacje na tabelach bazy.
- Template - sekcja prezentacji. Podstawą szablonów jest HTML[5]. Szablony tworzone są jak klasyczne strony z wykorzystaniem HTML[5]. Django[2] dostarcza narzędzia do wygodnej interakcji i modyfikacji stron HTML[5]. Pozwalają one na m.in. na dynamiczne “wstrzykiwanie” elementów do strony, czy kontrolę nad układem i zawartością strony za pomocą skrzynek warunkowych czy pętli.
- View - sekcja funkcjonalności. Widoki zawierają w sobie funkcjonalność stron. Widok jest odpowiedzialny za dostarczenie danych, które zostaną wyświetlone za pomocą szablonu, czy

przetworzenie zawartości formularza, który został przez szablon wyświetlony i wypełniony przez użytkownika.

Całość łączy ze sobą serwer dostarczany razem z frameworkiem, który realizuje komunikację z bazą danych, komunikację poprzez HTTP[5], wystawianie stron przeglądarkom, oraz odbieranie zapytań. Oprócz wymienionych elementów, możliwe jest też wykorzystanie javascript[5], CSS[5] i innych standardowych narzędzi stosowanych przy tworzeniu serwisów sieciowych.

Rozdział 3

Projekt aplikacji

3.1. Plan aplikacji

Aplikacja składać się będzie z 3 głównych sekcji:

- Accounts - sekcja poświęcona kontom użytkowników. Zawierać będzie strony, modele i widoki skupione wokół profili użytkowników i ich funkcjonalności
- Threads - sekcja głównej zawartości forum. Należą do niej strony, modele i widoki poświęcone tematom dyskusyjnym i ogłoszeniom.
- Conversations - sekcja dotycząca funkcjonalności prywatnych konwersacji między użytkownikami. Zawiera w sobie strony, modele i widoki potrzebne do zrealizowania wymiany wiadomości między profilami użytkowników.

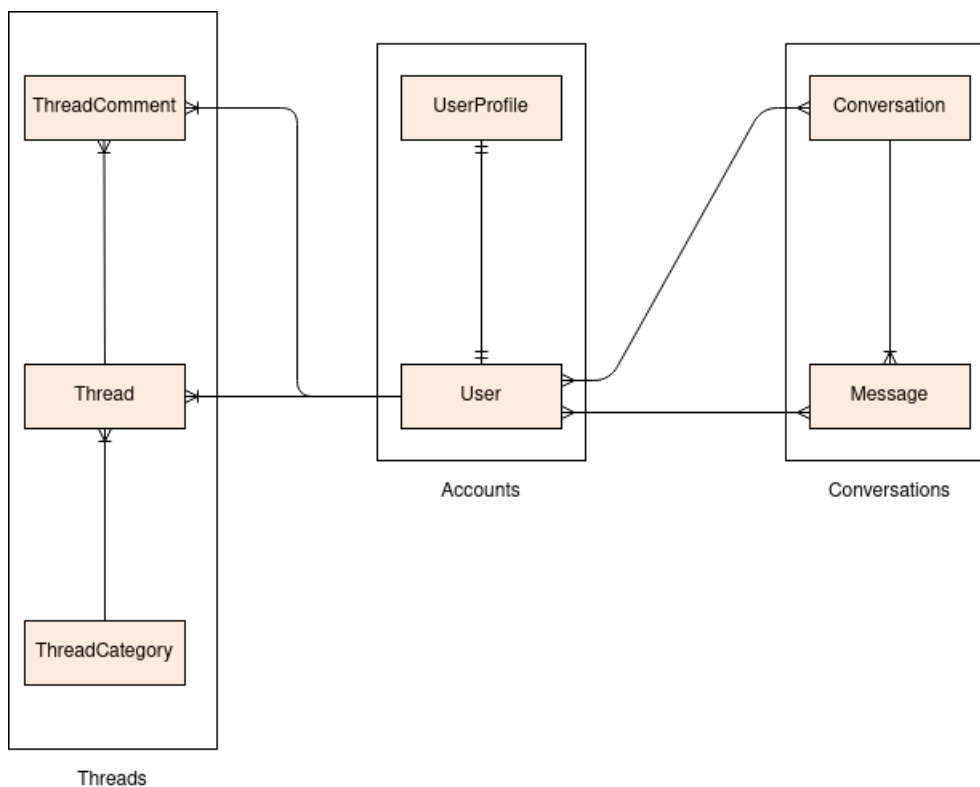
Sekcje te nazywane przez Django[2] aplikacjami, służą głównie wygodzie i przejrzystości tworzenia serwisu. Jest możliwe zrealizowanie całego projektu z wykorzystaniem tylko jednej, głównej aplikacji. Podział na sektory poświęcone konkretnym grupom funkcjonalności pozwala na uzyskanie większej klarowności i czytelności w tworzonej aplikacji. oraz wygodniejszej rozbudowy serwisu w przyszłości.

Django[2] dostarcza wbudowany model User zawierający w sobie podstawowe pola użytkownika, oraz funkcjonalności autoryzacji. Aby zbudować własny model użytkownika można zmodyfikować ten model i nadpisać go, otrzymując w rezultacie swój własny model User, lub utworzyć osobny model przechowujący dodatkowe informacje i połączyć go relacją 1 do 1 z modelem User. Ze względu na brak planów modyfikowania, lub zastępowania domyślnych funkcji modelu User, autor zdecydował skorzystać z drugiej metody rozszerzenia domyślnego modelu.

3.2. Diagramy aplikacji

3.2.1. ERD

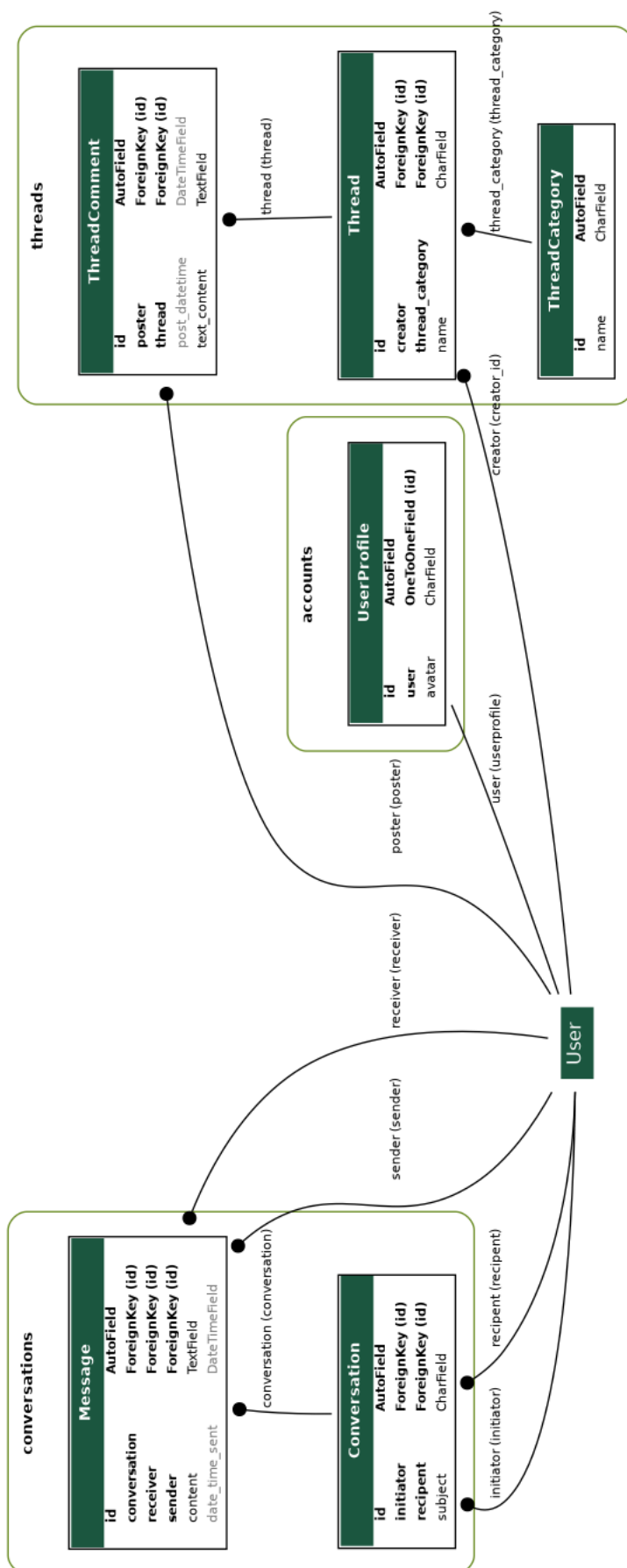
W przypadku sekcji Conversations występują relacje wiele do wielu w przypisaniu użytkownika do wiadomości. Każda konwersacja posiada 2 strony - użytkownika który ją rozpoczął i adresata. Tak samo każda z wiadomości posiada nadawcę i odbiorcę. Użytkownicy mogą prowadzić wiele konwersacji na portalu. Istnieje wiele możliwości realizacji takiej relacji w zależności od konkretnej sytuacji i priorytetów przy tworzeniu serwisu. Z racji na przewidywalną naturę relacji wiele do wielu (użytkownik może wysyłać wiele wiadomości i prowadzić wiele rozmów, ale każda wiadomość i konwersacja odnosi się zawsze do 2 użytkowników), zdecydowano nie implementować tabeli pośredniej, a zrealizować relację poprzez 2 klucze obce



Rys. 3.1: Diagram związków encji

użytkownika w tabelach wiadomości i konwersacji. Zapewnienie poprawności znajdujących się tam kluczy należy do aplikacji i widoków wprowadzających te wartości.

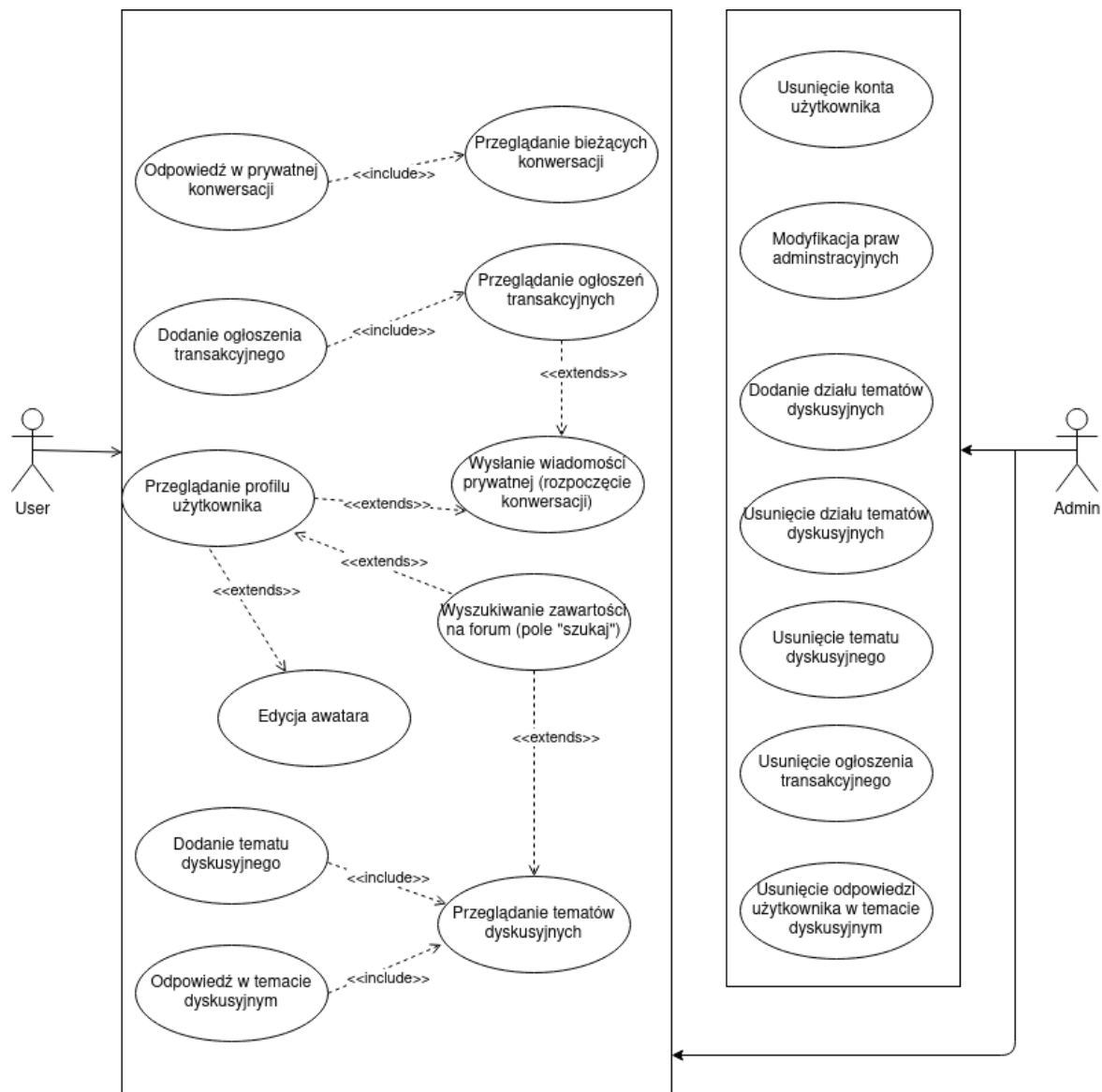
3.2.2. Diagram tabel bazy danych



Rys. 3.2: Diagram tabel bazy danych aplikacji

Tabela User to wbudowany w Django[2] domyślny model użytkownika, zawierający podstawowe pola oraz mechanizmy autoryzacji. na potrzeby pisanej aplikacji można go rozszerzać o dodatkowe informacje i funkcjonalności (tabela UserProfile)

3.2.3. Diagram przypadków użycia



Rys. 3.3: Diagram przypadków użycia

3.3. Scenariusze przypadków użycia

3.3.1. Dodanie tematu dyskusyjnego

Aktor: Użytkownik/Administrator

Cel: Dodanie nowego tematu do dyskusji

Warunki początkowe: Użytkownik utworzył konto i jest zalogowany

Warunki końcowe: Utworzenie nowego tematu dyskusyjnego

Przebieg:

1. Użytkownik wchodzi do działu w którym chce dodać temat dyskusyjny
2. Użytkownik klika przycisk „Dodaj nowy Temat”
3. Użytkownik zostaje przekierowany do formularza utworzenia nowego tematu
4. Użytkownik wpisuje nazwę tematu który chce utworzyć i swoją pierwszą wypowiedź w nim
5. Użytkownik potwierdza utworzenie tematu za pomocą przycisku „Utwórz”
6. Następuje przekierowanie na stronę nowo utworzonej dyskusji.

3.3.2. Dodanie ogłoszenia transakcyjnego

Aktor: Użytkownik/Administrator

Cel: Dodanie nowego ogłoszenia

Warunki początkowe: Użytkownik utworzył konto i jest zalogowany

Warunki końcowe: Utworzenie ogłoszenia transakcyjnego

Przebieg:

1. Użytkownik wchodzi do sekcji sprzedam/kupię/zamienię w zależności od tego jakie ogłoszenie chce dodać
2. Użytkownik klika przycisk „Dodaj nowe ogłoszenie”
3. Użytkownik zostaje przekierowany do formularza utworzenia nowego ogłoszenia
4. Użytkownik wpisuje nazwę ogłoszenia które chce utworzyć i wpisuje opis przedmiotów, warunki transakcji itd.
5. Użytkownik potwierdza utworzenie ogłoszenia za pomocą przycisku „Utwórz”
6. Następuje przekierowanie na stronę nowo utworzonego ogłoszenia.

3.3.3. Dodanie odpowiedzi w temacie dyskusyjnym

Aktor: Użytkownik/Administrator

Cel: Dodanie nowej odpowiedzi

Warunki początkowe: Użytkownik utworzył konto i jest zalogowany

Warunki końcowe: Nowa odpowiedź dodana

Przebieg:

1. Użytkownik przechodzi na stronę dyskusji
2. Użytkownik w formularzu na dole strony wpisuje swoją wypowiedź
3. Użytkownik klika przycisk „Odpowiedz”
4. Następuje przekierowanie z powrotem do strony tematu z dodaną przed chwilą odpowiedzią

3.3.4. Przeglądanie profilu użytkownika

Aktor: Użytkownik/Administrator

Cel: Wyświetlenie profilu użytkownika

Warunki początkowe: Użytkownik utworzył konto i jest zalogowany

Warunki końcowe: Wyświetlony profil użytkownika

Przebieg:

1. Użytkownik klika w przycisk z nazwą swojego konta znajdujący się na górnej belce strony, lub w alias jakiegokolwiek użytkownika (znajdujący się np. przy udzielonej przez niego odpowiedzi w temacie dyskusyjnym, lub w wynikach wyszukiwania na forum)
 - a) Użytkownik przeszedł na stronę własnego profilu
 - Wyświetlony zostaje profil. Użytkownik ma możliwość edycji swojego avatara
 - b) Użytkownik przeszedł na profil innego użytkownika
 - Wyświetlony zostaje profil innego użytkownika. Za pomocą przycisku „Wyślij wiadomość” można wysłać użytkownikowi prywatną wiadomość, rozpoczynając rozmowę.

3.3.5. Wysyłanie wiadomości prywatnej (rozpoczęcie konwersacji)

Aktor: Użytkownik/Administrator

Cel: Rozpoczęcie wymiany wiadomości z użytkownikiem

Warunki początkowe: Użytkownik utworzył konto i jest zalogowany

Warunki końcowe: Rozpoczęto nową konwersację

Przebieg:

1. Użytkownik klika w przycisk „Wyślij wiadomość” znajdujący się na profilu innego użytkownika lub na zamieszczonym przez niego ogłoszeniu.
2. Następuje przeniesienie do formularza rozpoczęcia konwersacji.
3. Użytkownik podaje temat rozmowy oraz pisze pierwszą wiadomość.
4. Po kliknięciu przycisku „Wyślij” następuje przeniesienie na stronę nowo utworzonej konwersacji

3.3.6. Edycja avatara

Aktor: Użytkownik/Administrator

Cel: Zmiana avatara użytkownika

Warunki początkowe: Użytkownik utworzył konto i jest zalogowany.

Warunki końcowe: Awatar zostaje zmieniony.

Przebieg:

1. Użytkownik klika przycisk „Edytuj awatar” na stronie swojego profilu.
2. Następuje przekierowanie na stronę gdzie za pomocą widżetu użytkownik podaje plik graficzny który chce ustawić jako awatar
3. Zatwierdza wybór przyciskiem „zmień awatar”
4. Następuje przekierowanie z powrotem na profil użytkownika

3.3.7. Udzielenie odpowiedzi w prywatnej konwersacji

Aktor: Użytkownik/Administrator

Cel: Wysłanie odpowiedzi w konwersacji innemu użytkownikowi

Warunki początkowe: Użytkownik utworzył konto i jest zalogowany. Istnieje utworzona konwersacja o danym temacie z danym użytkownikiem

Warunki końcowe: Udzielono odpowiedzi w konwersacji

Przebieg:

1. Użytkownik przechodzi do panelu swoich rozmów
2. Wyświetlonych konwersacji które prowadzi z innymi użytkownikami wybiera tą która go interesuje
3. W formularzu na odpowiedź znajdującym się pod ostatnią wysłaną wiadomością wpisuje swoją odpowiedź
4. Potwierdza przyciskiem „Odpowiedz”
5. Następuje przekierowanie na stronę konwersacji z udzieloną przed chwilą odpowiedzią

3.3.8. Usunięcie konta użytkownika

Aktor: Administrator

Cel: Usunięcie konta innego użytkownika

Warunki początkowe: Administrator przeszedł do panelu administratora i zalogował się danymi swojego konta

Warunki końcowe: Profil danego użytkownika zostaje usunięty

Przebieg:

1. Administrator przechodzi do zakładki „Użytkownicy”
2. Z listy wybiera (lub wyszukuje) użytkownika którego konto chce usunąć i klika „Usuń”
3. Potwierdza swój wybór na ekranie potwierdzenia usunięcia
4. Użytkownik zostaje usunięty z bazy danych aplikacji

3.3.9. Nadanie/Odebranie praw administracyjnych

Aktor: Administrator

Cel: Modyfikacja uprawnień innego użytkownika

Warunki początkowe: Administrator przeszedł do panelu administratora i zalogował się danymi swojego konta

Warunki końcowe: Profil danego użytkownika zmienia uprawnienia

Przebieg:

1. Administrator przechodzi do zakładki „Użytkownicy”
2. Z listy wybiera (lub wyszukuje) użytkownika i klika na niego przechodząc do jego szczegółowego panelu
3. W sekcji uprawnień zaznacza lub odznacza pole uprawnień administratorskich.
4. Klika przycisk „Zapisz” potwierdzając modyfikację konta
5. Uprawnienia użytkownika zostają zmienione

3.3.10. Dodanie działu tematów dyskusyjnych

Aktor: Administrator

Cel: Dodanie działu tematów dyskusyjnych

Warunki początkowe: Administrator przeszedł do panelu administratora i zalogował się danymi swojego konta

Warunki końcowe: Nowy dział dyskusyjny utworzony

Przebieg:

1. Administrator przechodzi do zakładki „Działy”
2. Administrator klika przycisk „dodaj nowy”
3. Podaje nazwę działu i potwierdza przyciskiem „Dodaj”
4. Nowy dział na tematy dyskusyjne zostaje utworzony

3.3.11. Usunięcie działu tematów dyskusyjnych

Aktor: Administrator

Cel: Usunięcie działu tematów dyskusyjnych

Warunki początkowe: Administrator przeszedł do panelu administratora i zalogował się danymi swojego konta

Warunki końcowe: dział dyskusyjny Usunięty

Przebieg:

1. Administrator przechodzi do zakładki „Działy”
2. Administrator wybiera (lub wyszukuje) z listy dział i zaznacza go
3. klika „Usuń” i potwierdza wybór na ekranie potwierdzenia
4. Dział zostaje usunięty

3.3.12. Usunięcie tematu dyskusyjnego

Aktor: Administrator

Cel: Usunięcie tematu dyskusyjnego

Warunki początkowe: Administrator przeszedł do panelu administratora i zalogował się danymi swojego konta

Warunki końcowe: Temat dyskusyjny usunięty

Przebieg:

1. Administrator przechodzi do zakładki „Tematy”
2. Administrator wybiera (lub wyszukuje) z listy temat i zaznacza go
3. klika „Usuń” i potwierdza wybór na ekranie potwierdzenia
4. Temat zostaje usunięty

3.3.13. Usunięcie odpowiedzi Użytkownika w temacie dyskusyjnym

Aktor: Administrator

Cel: Usunięcie odpowiedzi w temacie dyskusyjnym

Warunki początkowe: Administrator przeszedł do panelu administratora i zalogował się danymi swojego konta

Warunki końcowe: Odpowiedź w temacie usunięta

Przebieg:

1. Administrator przechodzi do zakładki „Komentarze”
2. Administrator wybiera (lub wyszukuje) z listy komentarz i zaznacza go
3. klika „Usuń” i potwierdza wybór na ekranie potwierdzenia
4. Komentarz zostaje usunięty

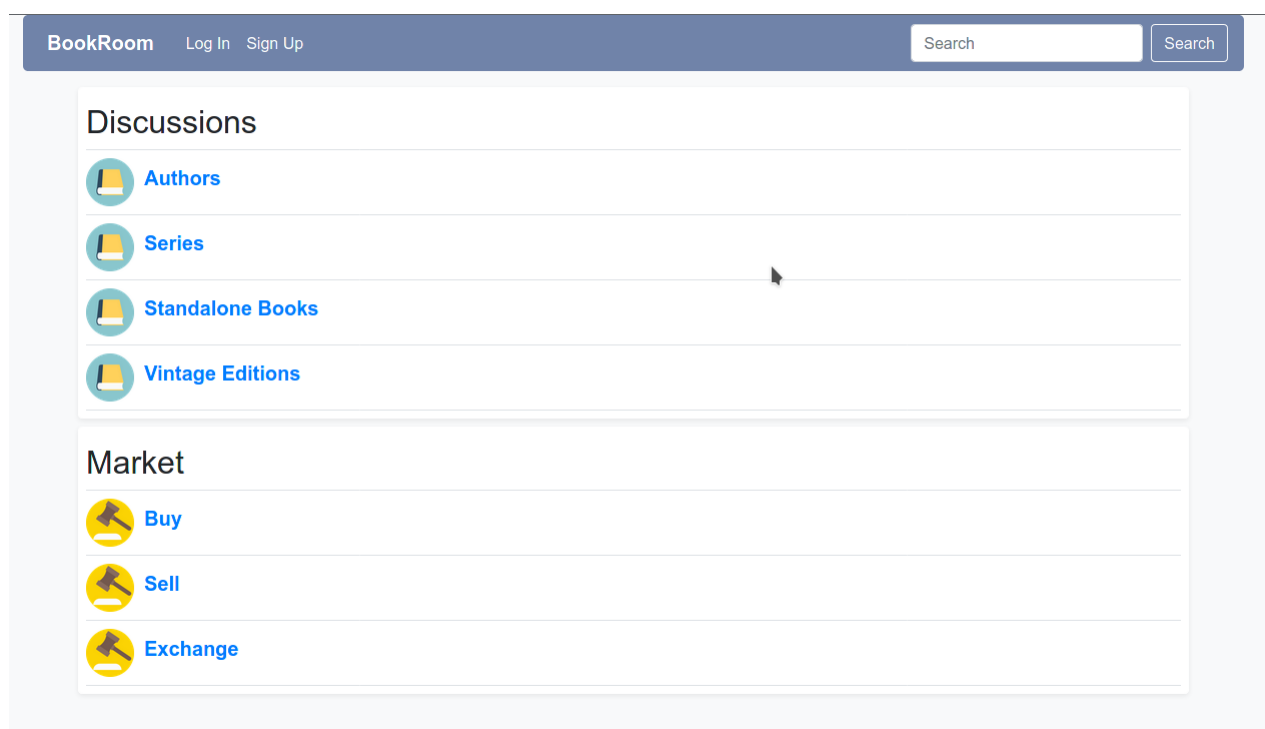
Rozdział 4

Aplikacja

4.1. Prezentacja serwisu

4.1.1. Strona główna

Strona główna serwisu prezentuje użytkownikowi dostępne do przeglądania działy tematyczne, oraz transakcyjne. Górna belka zapewnia dostęp do funkcjonalności związanych z kontem użytkownika. W zależności od aktualnego statusu użytkownika dostępne opcje zmieniają się. Do myślnie, “obca” osoba ma możliwość zalogować się, lub utworzyć konto.



Rys. 4.1: Strona główna serwisu.

4.1.2. Rejestracja

Jako niezarejestrowany użytkownik można przeglądać zawartość tematów, nie ma jednak możliwości udzielać się w nich czy korzystać z jakichkolwiek innych funkcjonalności. Nowy użytkownik musi więc utworzyć konto.

Rys. 4.2: Strona rejestracji.

Zaimplementowane zostały restrykcje dotyczące zawartości pól. Pole adresu e-mail nie przyjmuje wartości, która nie odpowiada wzorcowi adresu mailowego. Hasło musi zostać potwierdzone i spełniać standardy bezpieczeństwa, a nazwa użytkownika składać się z dozwolonych znaków. W przypadku wprowadzenia wartości nie spełniających warunków, strona powiadomi użytkownika o błędzie

Rys. 4.3: Użytkownik wprowadził hasło nie spełniające wymogów.

Jeśli wszystkie dane zostaną wprowadzone poprawnie, konto zostanie utworzone, oraz nastąpi przekierowanie na stronę logowania.

4.1.3. Logowanie

Strona logowania dostępna poprzez górną belkę. Tutaj zostanie też przekierowany użytkownik po rejestracji. Serwis zweryfikuje poprawność wprowadzonych danych i zaloguje użytkownika.

Log In!

Username

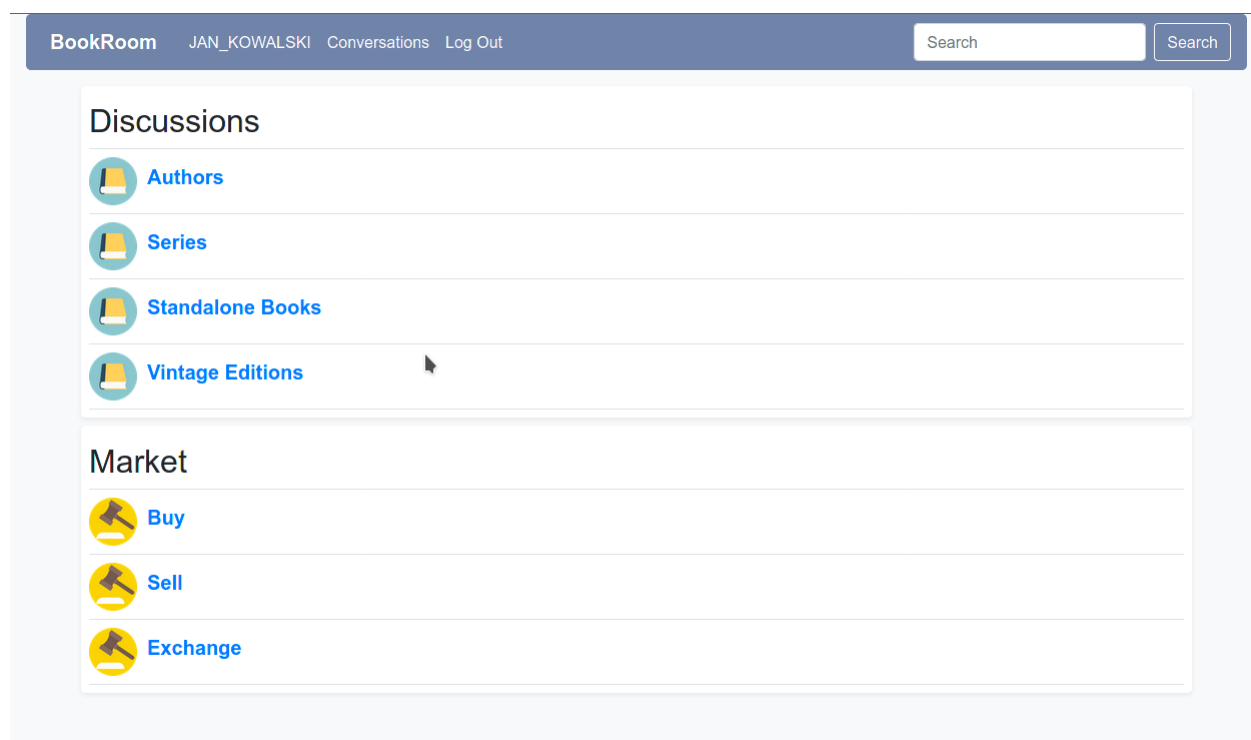
Jan_Kowalski

Password

Log In

Rys. 4.4: Formularz logowania

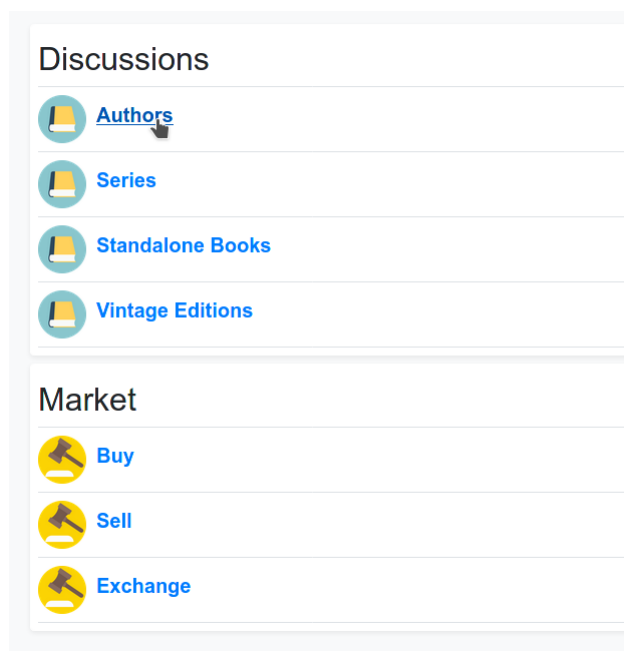
Po zalogowaniu zmieniają się opcje dostępne na belce. Użytkownik może przejść do swojego profilu, otworzyć panel konwersacji, lub wylogować się.



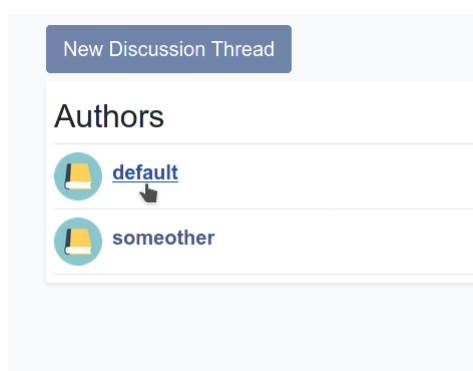
Rys. 4.5: Strona główna z nowymi pozycjami górnej belki

4.1.4. Tematy dyskusyjne

Po przejściu do interesującego użytkownika działu wyświetlone zostaną tematy do niego przynależące. Użytkownik może czytać i odpowiadać w istniejących tematach, lub założyć nowy.

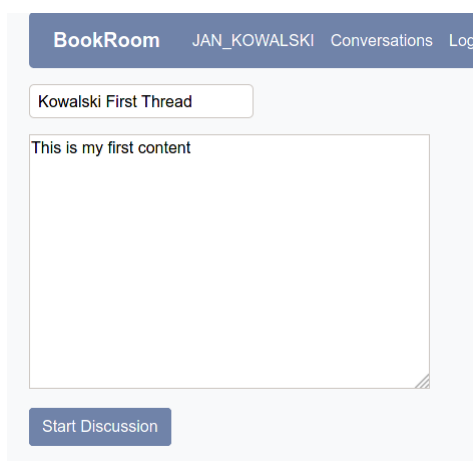


Rys. 4.6: Wybór jednego z działów



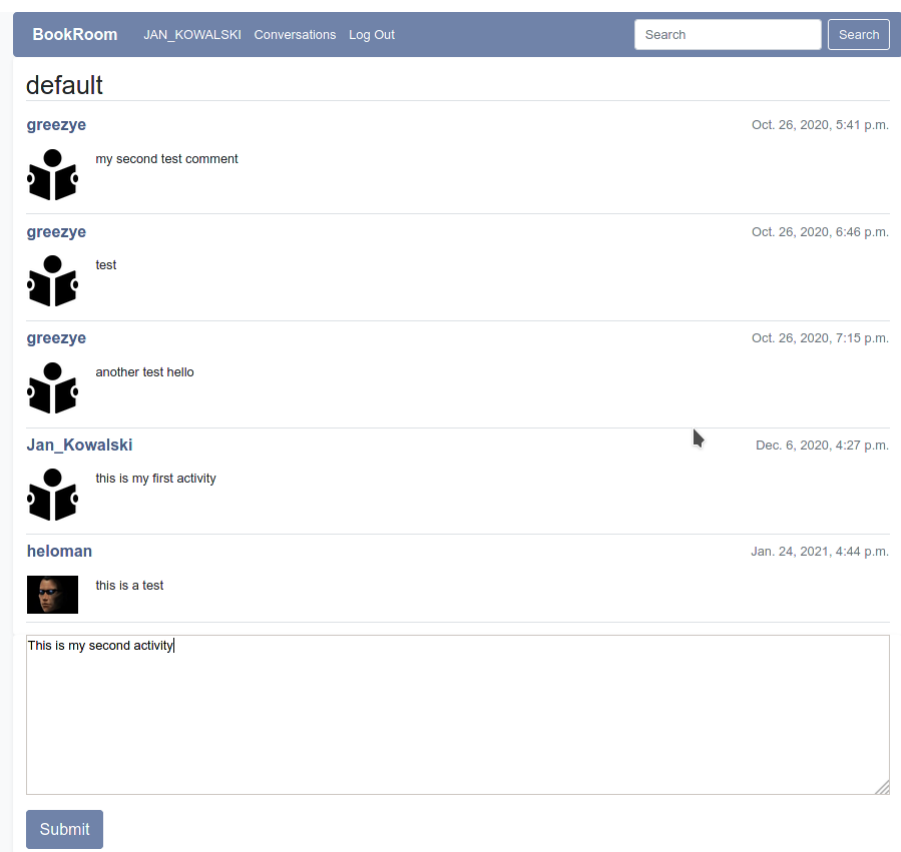
Rys. 4.7: Zawartość działu

Użytkownik może wybrać jeden z tematów, lub utworzyć nowy.



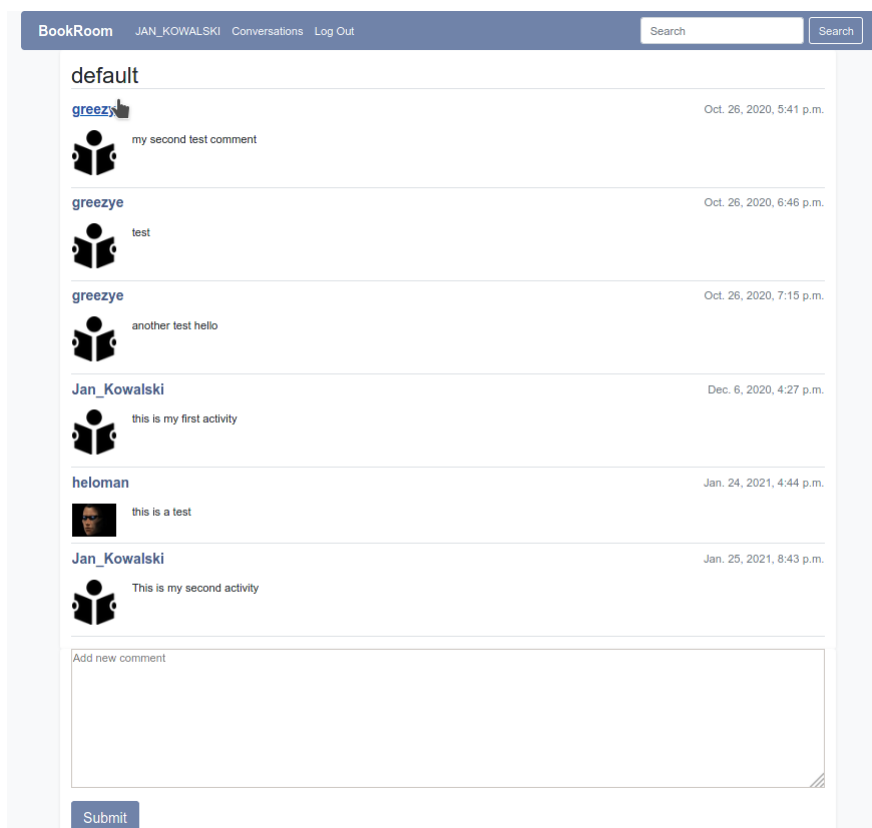
Rys. 4.8: Utworzenie nowego tematu

Jeśli użytkownik zdecyduje się utworzyć nowy temat, musi nadać mu nazwę, oraz umieścić pierwszą wypowiedź.



Rys. 4.9: Dyskusja w istniejącym temacie

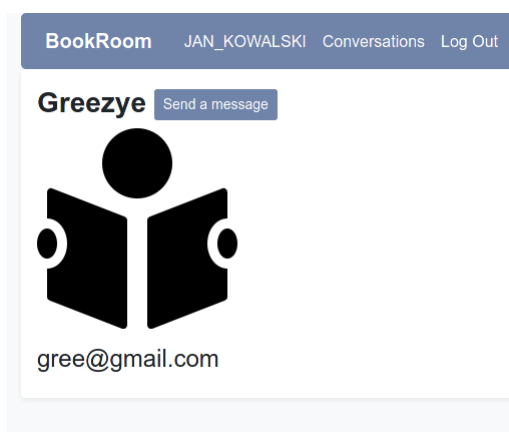
W temacie dyskusyjnym, prezentowane są dotychczasowe odpowiedzi w dyskusji. Pod ostatnią odpowiedzią znajduje się formularz do włączenia się w dyskusję.



Rys. 4.10: Komentarz widoczny na stronie po dodaniu

4.1.5. Profil użytkownika i konwersacje

Klikając na nick użytkownika (np. przy udzielonej przez niego odpowiedzi w temacie) można przejść do jego strony profilowej. Wyświetlane są tam informacje o użytkowniku.



Rys. 4.11: Strona profilu użytkownika

Z profilu użytkownika można rozpocząć z nim konwersację.

The screenshot shows a web interface for starting a new conversation. At the top, a dark blue header bar contains the text 'BookRoom', 'JAN_KOWALSKI', 'Conversations', and 'Log Out'. Below the header, the main heading is 'Start new conversation with greezye'. There is a text input field with the placeholder text 'Book series for sale'. Below the input field is a large text area containing placeholder text: 'Lorem ipsum dolor sit amet, consectetur adipiscing elit. Ut tincidunt nisi ullamcorper erat luctus hendrerit. Ut id luctus sem. Proin mauris lorem, mattis ac diam nec, porttitor condimentum ligula.' At the bottom of the text area is a blue button labeled 'Send'.

Rys. 4.12: Formularz wysłania pierwszej wiadomości (i rozpoczęcia konwersacji)

Po wysłaniu pierwszej wiadomości następuje przekierowanie na panel utworzonej konwersacji.

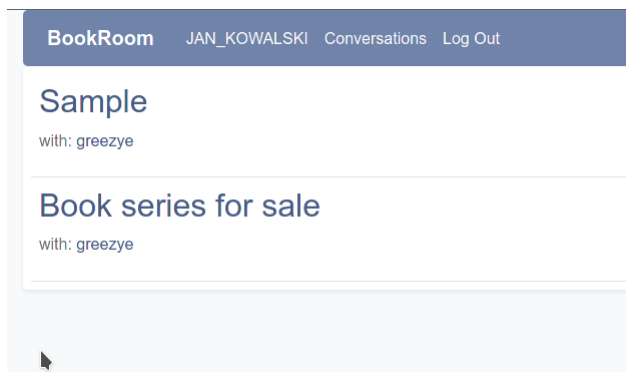
The screenshot shows the 'Conversation with greezye' panel. The header bar is the same as in the previous screenshot. Below the header, the heading is 'Conversation with greezye' and the subject is 'Subject: Book series for sale'. The main content area shows a message from 'Jan_Kowalski' with a book icon and placeholder text: 'Lorem ipsum dolor sit amet, consectetur adipiscing elit. Ut tincidunt nisi ullamcorper erat luctus hendrerit. Ut id luctus sem. Proin mauris lorem, mattis ac diam nec, porttitor condimentum ligula.' Below the message is a large text input field for replying. At the bottom of the input field is a blue button labeled 'Reply'.

Rys. 4.13: Strona rozmowy

kolejne odpowiedzi będą pojawiać się pod ostatnią wysłaną wiadomością. W każdej chwili użytkownik może otworzyć panel wszystkich prowadzonych konwersacji.

The screenshot shows the 'Discussions' panel. The header bar is the same as in the previous screenshots. Below the header, the heading is 'Discussions'. There are two items listed: 'Authors' and 'Series', each with a book icon and a blue label.

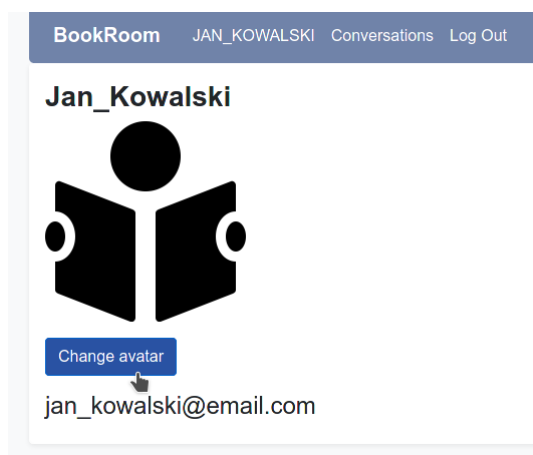
Rys. 4.14: Ikona panelu konwersacji



Rys. 4.15: Panel konwersacji

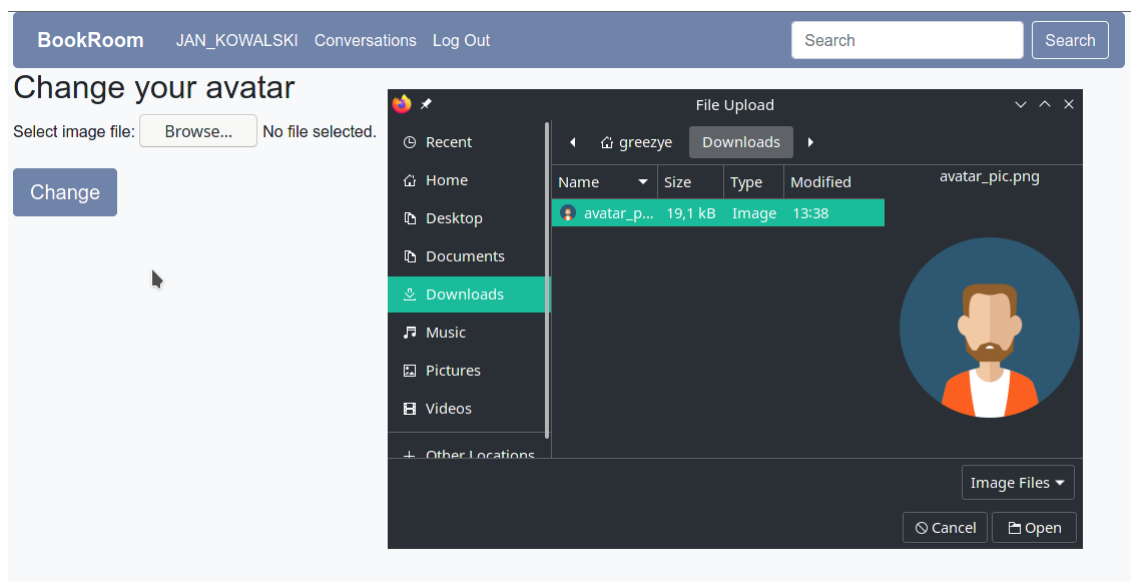
Użytkownik może przejrzeć zawartość każdej prowadzonej konwersacji i udzielić w nich odpowiedzi.

Po przejściu na stronę własnego profilu (np. poprzez kliknięcie nazwy użytkownika na górnej belce), użytkownik ma możliwość zmiany swojego zdjęcia profilowego.

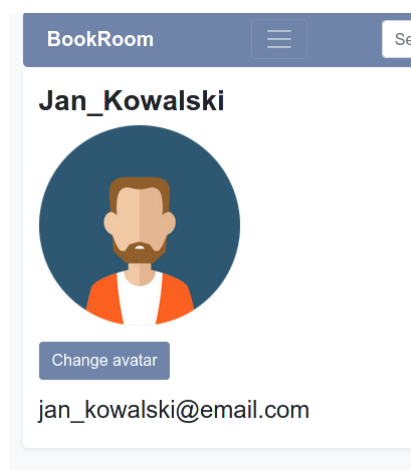


Rys. 4.16: Zmiana awatara

Należy wybrać nowy obraz korzystając z przeglądarki plików, oraz zatwierdzić wybór.



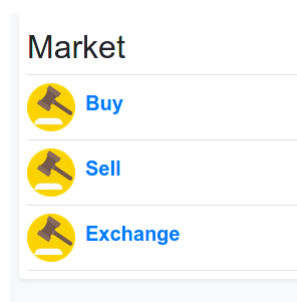
Rys. 4.17: Wybór nowego zdjęcia



Rys. 4.18: Nowy awatar

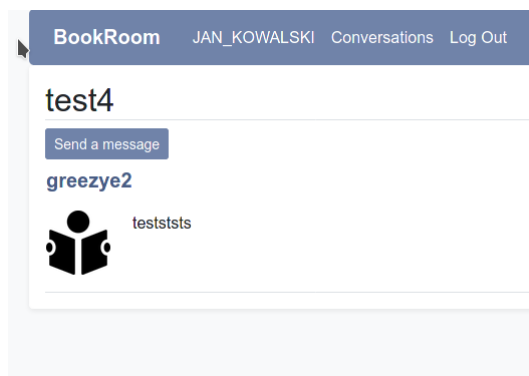
4.1.6. Ogłoszenia

Z poziomu strony głównej dostępne są trzy działy ogłoszeń transakcyjnych: Sprzedam, kupię i zamienię. Ich działanie jest podobne do tematów dyskusyjnych, ale posiadają swoje unikalne funkcje.

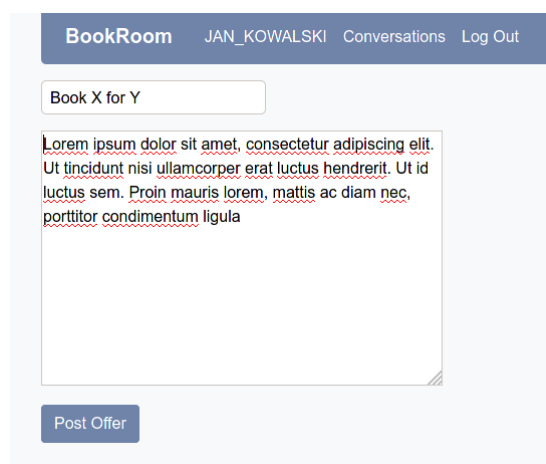


Rys. 4.19: Sekcja ogłoszeń

W ogłoszeniu komentarze dodawane mogą być tylko przez autora. Każdy inny użytkownik po otwarciu ogłoszenia widzi przycisk 'Skontaktuj się' który pozwala od razu rozpocząć konwersację z autorem ogłoszenia, dotyczącą transakcji. Autor ma możliwość również zakończyć ogłoszenie po zrealizowanej transakcji przyciskiem 'Zamknij ogłoszenie'



Rys. 4.20: Ogłoszenie innego użytkownika



Rys. 4.21: Tworzenie nowego ogłoszenia

The screenshot shows a web interface for a transaction titled "Book X for Y". At the top, a dark blue header bar contains the text "BookRoom", the username "JAN_KOWALSKI", and links for "Conversations" and "Log Out". Below the header, the title "Book X for Y" is displayed. Underneath the title is a button labeled "End Transaction Offer". The user's name "Jan_Kowalski" is shown, followed by a circular profile picture of a man with a beard and a placeholder text "Lorem ipsum dolor sit amet, consectetur adipiscing elit. Ut tincidunt nisi u lorem, mattis ac diam nec, porttitor condimentum ligula". Below this is a text input field with the placeholder "Add new comment" and a "Submit" button at the bottom.

Rys. 4.22: Własne ogłoszenie

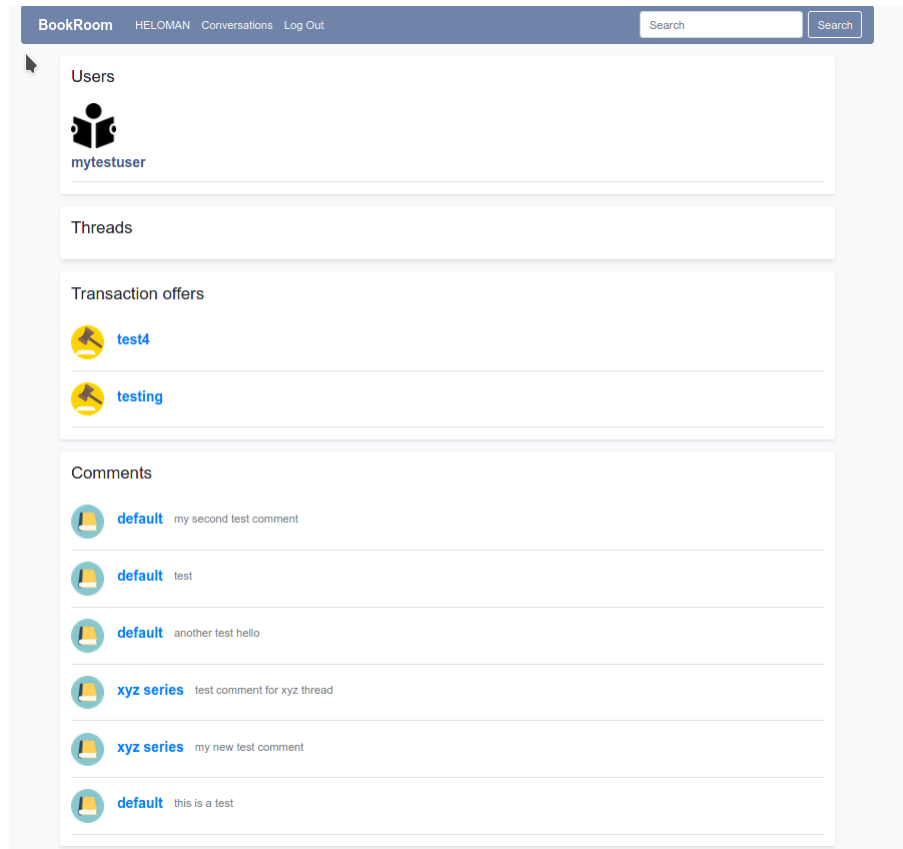
4.1.7. Wyszukiwanie

Serwis oferuje wyszukiwanie zawartości za pomocą pola znajdującego się na górnej belce.

The screenshot shows the top of the BookRoom application. A dark blue header bar contains the text "BookRoom", the username "HELOMAN", and links for "Conversations" and "Log Out". On the right side of the header is a search bar with the text "test" and a "Search" button. Below the header, the word "Discussions" is displayed. Underneath "Discussions" is a list of four categories, each with a circular icon containing a book and the category name: "Authors", "Series", "Standalone Books", and "Vintage Editions".

Rys. 4.23: Formularz wyszukiwania

Po kliknięciu 'szukaj' następuje przekierowanie na stronę, zwracającą pasujące wyniki z różnych kategorii.



Rys. 4.24: Wyniki

4.2. Realizacja

Sekcja ta, ma na celu zaprezentować kilka przykładów działania aplikacji za pomocą fragmentów kodu i towarzyszących im opisów. Celem autora jest przedstawić wybrane z narzędzi dostarczanych przez Django[2], w celu przedstawienia w jaki sposób operuje framework i jak został wykorzystany przy tworzeniu aplikacji.

4.2.1. Rejestracja użytkownika

Listing 4.1: Widok rejestracji nowego użytkownika

```
class CreateUserForm(CreateView):
    form_class = forms.UserCreateForm
    success_url = reverse_lazy('accounts:login')

    template_name = 'accounts/signup.html'

    def form_valid(self, form):
        super(CreateUserForm, self).form_valid(form)

        user = self.object
        userprofile = UserProfile(user=user)
        userprofile.save()

        return HttpResponseRedirect(self.success_url)
```

Do zrealizowania rejestracji użytkownika wykorzystane zostały wysokopoziomowe narzędzia widoków klasowych, dostarczane przez Django[2]. Są one mniej intuicyjne niż klasyczne

widoki funkcyjne omawiane w dalszej części. Dla pewnych standardowych operacji (głównie operacje typu CRUD) pozwalają jednak na szybką realizację w kilku liniach kodu. Większość pracy wykonywana jest przez gotowe narzędzia, które należy tylko skonfigurować i ew. trochę zmodyfikować.

W tej sytuacji wykorzystywany jest widok `CreateView`. Jak nazwa sugeruje, realizuje on operację `create`. Widok ten (po imporcie z bibliotek Django[2]) przekazywany jest jako klasa nadrzędna, po której dziedziczy tworzony przez programistę widok klasowy. Dzięki temu, widok ma zapewnione wszystkie funkcje, potrzebne do utworzenia nowej instancji modelu, a następnie przetłumaczenia go i zapisania w odpowiedniej tabeli bazy danych. Dostosowanie widoku do indywidualnych potrzeb, odbywa się za pomocą nadpisania odziedziczonych atrybutów, oraz przeładowania metod klasy nadrzędnej.

`form_class` to atrybut do którego przypisać należy utworzony wcześniej formularz, który w celu rejestracji wypełni użytkownik. Formularz ten, na podobnej zasadzie został utworzony poprzez dziedziczenie klasy nadrzędnej formularza i odpowiednią modyfikację.

Listing 4.2: Formularz rejestracji użytkownika

```
class UserCreateForm(UserCreationForm):
    class Meta:
        model = get_user_model()
        fields = ('username', 'email', 'password1', 'password2')

    def __init__(self, *args, **kwargs):
        super().__init__(*args, **kwargs)
        self.fields['username'].label = 'Display Name'
        self.fields['email'].label = 'Email Address'
```

Wewnętrzna klasa `Meta` pozwala na konfigurację elementów formularza. W tym wypadku przekazana zostaje informacja, jaki model będzie tworzony za pomocą formularza. `get_user_model` zwraca aktualnie używany model użytkownika - tutaj wbudowany model `User`. Następnie definiowane są pola jakie formularz będzie zawierał. Szczegóły konfiguracji i wszystkie dostępne opcje znajdują się w dokumentacji frameworku. Na końcu zdefiniowana zostaje metoda `__init__` wywoływana przy tworzeniu instancji klasy.

Formularz przekazany do widoku informuje o tym jakie pola zawierać powinien formularz wyświetlany na stronie, jakiego modelu dotyczyć będzie operacja, oraz przekazuje narzędzia potrzebne do zrealizowania tych operacji. Atrybuty `success_url` oraz `template_name` informują kolejno o tym, gdzie ma nastąpić przekierowanie po wykonaniu operacji utworzenia użytkownika, oraz jaki szablon jest powiązany z tym widokiem.

W przypadku gdyby aplikacja korzystała tylko z domyślnego modelu użytkownika, byłby to koniec konfiguracji widoku. Należy jednak utworzyć jeszcze instancję modelu. `UserProfile` i powiązać ją z świeżo utworzoną instancją `User`. To zadanie pełni przeładowanie metody `form_valid`. Metoda ta wykonywana jest kiedy wszystkie dane zostały wprowadzone do formularza poprawnie i można przejść do etapu ich przetworzenia i zapisania. Na początku wywoływana jest ta sama metoda, ale dla klasy nadrzędnej. Ta sama operacja została by przeprowadzona gdyby nie nastąpiło przeładowanie metody `form_valid`, a następnie nastąpiło by przekierowanie do `success_url`. W tej sytuacji celem jest wykonanie dodatkowych operacji po utworzeniu użytkownika, a przed przekierowaniem. Z instancji klasy (`self`) wyłuskiwany jest utworzony przed chwilą model użytkownika i przekazywany jako argument przy tworzeniu instancji modelu `UserProfile`. Następnie wywoływana jest metoda `save` instancji `UserProfile`, która zapisuje obiekt do bazy danych. W ten sposób do utworzonego wcześniej (i zapisanego w bazie danych) użytkownika, utworzona została i przypisana, instancja modelu rozszerzającego. Na końcu następuje przekierowanie na odpowiedni adres.

Jedyne co należy jeszcze wykonać to przekazać formularz do odpowiedniego szablonu.

Listing 4.3: Formularz przekazany do szablonu

```
<form method="post">
    {% csrf_token %}
    {% bootstrap_form form %}
    <input type="submit" class="btn btn-primary" value="Sign Up">
</form>
```

Domyślna nazwa “form” może zostać zmieniona na inną jeśli jest taka potrzeba. Zakończeniem procesu jest mapowanie widoku do adresu url w serwisie. Mapowanie odbywa się w przeznaczonym do tego pliku `urls.py`.

Listing 4.4: Mapowanie widoku do adresu url serwisu

```
path('signup/', views.CreateUserForm.as_view(), name='signup')
```

4.2.2. Temat dyskusyjny

Do zrealizowania strony tematu dyskusyjnego został wykorzystany klasyczny widok funkcyjny.

Listing 4.5: Widok funkcyjny tematu dyskusyjnego

```
def thread_view(request, pk):
    thread = Thread.objects.get(pk=int(pk))
    if request.method == 'GET':
        form = forms.AddComment(auto_id=False)

        comments = ThreadComment.objects.filter(thread=thread.pk)

        return render(request, 'threads/thread.html', context={'thread':
            ↪ thread,
                                                                'comments':
            ↪ comments
            ↪ ,
            'form': form
            ↪ })

    elif request.method == 'POST':
        form = forms.AddComment(request.POST)

        if form.is_valid():
            thread_comment = ThreadComment(thread=thread,
                                           poster=request.user,
                                           text_content=form.cleaned_data['
            ↪ text_content'])

            thread_comment.save()

        return redirect('threads:thread', pk=pk)
```

Widok pełni różne funkcje w zależności od tego jakie zapytanie http zostanie do niego skierowane. Dla zapytania GET, przygotowuje dane potrzebne do wyświetlenia strony tematu dyskusyjnego. Zapytanie POST skutkuje dodaniem nowego komentarza do tematu.

Kiedy użytkownik kliknie w link do tematu dyskusyjnego, zostanie wysłane zapytanie GET na adres przypisany do widoku `thread_view`. Do widoku jako argumenty zostaną przekazane: obiekt zapytania, oraz zawarty w adresie url klucz główny odwiedzanego wątku. Pierwszą operacją będzie wyszukanie w bazie danych odpowiedniego wiersza (bazując na przekazanym kluczu) i zwrócenie instancji modelu tematu dyskusyjnego utworzonej z tych danych (`thread = Thread.objects.get(pk=int(pk))`). Następnie w sekcji poświęconej zapytaniu GET, utworzona zostanie instancja pustego formularza, oraz pobrane z bazy danych komentarze, znajdujące się w temacie dyskusyjnym. Ostatnim krokiem jest zwrócenie wygenerowanej strony. Odbywa

się to za pomocą funkcji `render`, do której przekazywane jest zapytanie HTTP, referencja do szablonu, który powinien zostać użyty do utworzenia strony, oraz kontekst strony. Kontekst to słownik obiektów, które będzie można po przekazaniu “wstrzyknąć” w zawartość strony odwołując się do zdefiniowanych dla nich nazw.

Listing 4.6: Część szablonu tematu dyskusyjnego wyświetlająca komentarze oraz pole na dodanie nowego komentarza

```
{% for comment in comments %}
    <p><a href={% url 'accounts:user_profile_page' pk=comment.poster.pk
    ↪ %}>{{ comment.poster }}</a> said: {{ comment.text_content }}</p>
    <p></p>
    {% if comment.poster == user %}
        <a href="#" type="button" class="btn btn-primary btn-md">Delete
        ↪ Comment</a>
    {% endif %}
    <p>-----</p>
    <p></p>
{% endfor %}

<div class="basic">
    {% include "threads/add_comment_subtemplate.html" %}
</div>
```

Widoczna jest pętla FOR, skrzynka warunkowa, oraz referencja elementów kontekstu poprzez nazwy nadane w widoku. Końcowy efekt to strona wyświetlająca temat dyskusyjny razem ze znajdującymi się w nim komentarzami, oraz polem na dodanie nowego komentarza.

Jeśli użytkownik wprowadzi nowy komentarz, po zatwierdzeniu dodania, zostanie wysłane zapytanie na ten sam adres url. Tym razem będzie to zapytanie POST. Wywołany zostanie ten sam widok, ale wykona się jego sekcja poświęcona tej metodzie zapytania. Na podstawie informacji wprowadzonych w formularzu, utworzona zostanie instancja komentarza i zapisana do bazy danych. Na końcu nastąpi przekierowanie, w tym wypadku do tego samego widoku. Przekierowanie wysyła zapytanie GET. Zostanie więc wykonana procedura dla tego zapytania i strona tematu zostanie wyświetlona, zaktualizowana o dodany komentarz.

Jak prezentują przytoczone przykłady, Django[2] oferuje wiele wysoko poziomowych narzędzi, których zrozumienie i nauka użycia może zająć dużo czasu. Jednak po opanowaniu tych zagadnień możliwe jest, aby szybko i za pomocą niewielkiej ilości kodu, implementować złożone elementy tworzonej aplikacji.

Rozdział 5

Testy Aplikacji

5.1. Plan testów

Ze względu na rozmiar projektu oraz jego charakter, autor zdecydował się na testy manualne aplikacji. Testowane były poszczególne funkcje serwisu, kompozycja i wyświetlanie elementów interfejsu, przebieg operacji wykonywanych na stronie, oraz odzwierciedlenie wprowadzanych zmian w bazie danych. Testy skupiały się na sprawdzeniu poprawności działania interfejsu użytkownika oraz jego komunikacji z serwerem i bazą danych, przygotowana została lista przypadków testowych, które następnie były weryfikowane.

5.2. Przypadki testowe

5.2.1. Wątki dyskusyjne

- Tworzenie nowego wątku dyskusyjnego - poprawność wprowadzonych danych; czy nowy wątek został dodany i nastąpiło poprawne przekierowanie.
- Dodawanie Komentarzy - działanie formularza dodania komentarza; czy następuje poprawne przekierowanie po dodaniu komentarza; czy komentarz został dodany do dyskusji.
- Interfejs - poprawne wyświetlanie i zawartość nazw użytkowników, awatarów, zawartości tekstowej w tematach dyskusyjnych.
- Autoryzacja - czy możliwość tworzenia nowych tematów i udzielania się w istniejących jest dostępna tylko dla zalogowanego użytkownika - odpowiadające za to elementy interfejsu powinny być ukryte jeśli użytkownik nie jest zalogowany.
- Baza Danych - czy po dodaniu tematu/komentarza baza danych zostaje odpowiednio zmodyfikowana; czy po administracyjnym usunięciu tematu, należące do niego komentarze zostają kaskadowo usunięte.

5.2.2. Profil Użytkownika

- Wyświetlanie elementów - zawartość wszystkich elementów na stronie; modyfikacja interfejsu w zależności od przeglądającego użytkownika (niezalogowany, przeglądanie własnego i cudzego profilu).
- Zmiana awatara - przebieg procedury zmiany zdjęcia; zmiana odzwierciedlona na profilu i w zawartości użytkownika w serwisie.
- Usunięty użytkownik - po administracyjnym usunięciu użytkownika, test poprawności wyświetlania związanej z nim zawartości na serwisie (udzielone komentarze, utworzone wątki itd.).

- Baza Danych - czy wprowadzone zmiany są poprawnie odzwierciedlone w bazie danych.

5.2.3. Wiadomości

- Tworzenie nowej konwersacji - test poprawności przekierowań do formularza tworzenia rozmowy (z profilu użytkownika i jego ogłoszenia); poprawne utworzenie nowej konwersacji.
- Panel konwersacji - czy wszystkie rozmowy wyświetlane są poprawnie; czy można przeglądać poszczególne rozmowy.
- Rozmowa - Wysyłanie wiadomości w konwersacji; poprawne wyświetlanie w prezentacji rozmowy.
- Interfejs - poprawne wyświetlanie elementów rozmowy, informacji o użytkownikach na stronie.
- Baza Danych - czy wprowadzone zmiany są poprawnie odzwierciedlone w bazie danych.

5.2.4. Ogłoszenia

- Tworzenie nowych ogłoszeń - przebieg procedury tworzenia ogłoszenia.
- Interfejs - czy odpowiednie elementy strony są wyświetlane lub ukrywane w zależności od przeglądającego użytkownika (formularz nowego komentarza, przycisk zakończenia widoczny tylko dla twórcy ogłoszenia, możliwość kontaktu z wystawiającym ogłoszenie tylko dla innych użytkowników itp.).
- autoryzacja - możliwość tworzenia ogłoszeń tylko dla zalogowanych użytkowników.
- Baza Danych - czy wprowadzone zmiany są poprawnie odzwierciedlone w bazie danych.

5.2.5. Wyszukiwanie

- Wyszukiwanie zawartości - czy następuje poprawne przekierowanie na stronę z wynikami pasującymi do wyszukiwanej frazy; czy można przejść do każdego z wyników.
- Interfejs - poprawne wyświetlanie wyników, kompozycja i zawartość elementów na stronie z wynikami.

5.3. Wyniki

Przy pierwszej serii testów, powodzeniem zakończone zostały testy dotyczące działania poszczególnych funkcji, zmian wprowadzanych do bazy danych oraz prezentacji interfejsu. Testy autoryzacji zakończyły się powodzeniem ze strony serwera (np. niezalogowany użytkownik nie mógł utworzyć nowej zawartości), ze strony interfejsu wykryto błędy z wyświetlaniem zastrzeżonej zawartości (np. niezalogowany użytkownik dalej miał dostęp do formularzy). Niepowodzeniem zakończyły się również testy dotyczące zachowania systemu po usunięciu użytkownika, którego zawartość dalej istnieje w serwisie (strony interfejsu nie potrafiły obsłużyć brakujących informacji). Na podstawie zgromadzonych informacji wprowadzono stosowne poprawki i druga tura testów zakończona została pełnym sukcesem.

Rozdział 6

Podsumowanie

Budowa aplikacji sieciowej “od początku do końca” okazała się niezwykle ciekawym i satysfakcjonującym wyzwaniem. Do zrealizowania projektu należało posiadać wiedzę z różnych dziedzin informatyki. Tworzenie i praca z bazą danych, inżynierią oprogramowania, opanowanie i wykorzystanie frameworków sieciowych, a nawet tworzenie stron za pomocą HTML CSS i Bootstrap to tylko główne z umiejętności, które potrzebne były, aby zbudować działającą aplikację. Opanowanie poszczególnych zagadnień, oraz odpowiednie połączenie ze sobą różnych elementów aplikacji okazało się zadaniem trudnym, które zostało zostało ukończone przez autora niniejszej pracy. Został zbudowany sprawny i działający serwis spełniający założenia postawione na pierwszym etapie projektu. Zadanie dostarczyło tyle trudności co satysfakcji i przyjemności. Tworzenie kompletnego serwisu, pracując nad każdym jego elementem, pozwoliło zmierzyć się z różnymi wyzwaniami. Problemy rozwiązywane w trakcie pracy pochodziły z różnych dziedzin, przez co nie tylko ich rozwiązanie, ale też metodyka i proces ich rozwiązywania były bardzo zróżnicowane. Dzięki realizacji zadania projektowego, autor miał okazję nabyć wiedzę z obszarów wcześniej nieznanych, oraz udoskonalić umiejętności już posiadane. Na osobną uwagę zasługuje to, że projekt taki jest większy niż suma jego części składowych. W ocenie autora nawet ważniejszą umiejętnością, niż nabyta biegłość w tworzeniu poszczególnych elementów aplikacji, jest wiedza o tym, jak fragmenty te współpracują i komunikują się ze sobą aby stworzyć wspólną całość. Dzięki realizacji zadania projektowego, autor wzbogacił zasób swoich umiejętności związanych z programowaniem i tworzeniem aplikacji sieciowych. Przede wszystkim posiadał szersze i pełniejsze zrozumienie funkcjonowania złożonych serwisów webowych, oraz doświadczył głębokiej satysfakcji utworzenia kompletnego dzieła, nad którego każdym elementem pracował osobiście.

Aplikacja tworzona była tak, aby możliwa była jej dalsza rozbudowa. Wprowadzenie zautomatyzowanego systemu realizacji i przetwarzania zamówień jest główną funkcją, którą autor planuje dodać. Oprócz tego zamierza uzupełnić istniejące funkcje o dodatkowe usprawnienia takie jak opcje powiadomień e-mail i w przeglądarce, możliwość subskrybowania określonych treści, czy statystyki związane z aktywnością sprzedażową, oraz komentarze transakcji i punkty reputacji użytkownika. Następnie można skierować prace na dodanie zupełnie nowych autonomicznych funkcji, takich jak np. grupowy chatbox.

Literatura

- [1] Dokumentacja języka Python - <https://www.python.org/doc/>
- [2] Dokumentacja frameworku Django - <https://docs.djangoproject.com/en/3.1/>
- [3] Dokumentacja SQLite - <https://www.sqlite.org/docs.html>
- [4] Dokumentacja frameworku Bootstrap - <https://getbootstrap.com/docs/4.6>
- [5] Portal z materiałami deweloperskimi Mozilla - <https://developer.mozilla.org/en-US/>