

Warszawa, 14.06.2022 r.

**Politechnika Warszawska
Wydział Elektroniki i Technik Informacyjnych**

Bazy danych i Big Data - Projekt

**Faza #2
Realizacja aplikacji współpracującej z bazą danych**

Temat nr 6: Spółdzielnia mieszkaniowa

Prowadzący: mgr inż. Tomasz Mrozek

**Wykonawcy:
Aleksandra Sidor 303811, Radosław Żelazo 298738 (zespół 6)**

Spis treści

1. Wstęp	1
1.1. Cel projektu	1
1.2. Technologia	1
2. Realizacja projektu	2
2.1. Perspektywy	2
2.1.1. Perspektywa “szeroka”	4
2.1.2. Perspektywa “wąska”	9
2.2. Obsługa błędów	12

1. Wstęp

1.1. Cel projektu

Celem II części przedmiotu BDBT było zaprojektowanie oraz realizacja webowej aplikacji współpracującej z zaimplementowaną w I części bazą danych spółdzielni mieszkaniowej. Aplikacja powinna spełniać wybrane kilka funkcjonalności bazy danych oraz umożliwiać użytkownikom na wybrane działania w obrębie tych funkcjonalności.

1.2. Technologia

Aplikacja powstała środowisku IntelliJ IDEA 2022. 1. 1 z wykorzystaniem Spring Boot. Spring Boot jest to framework oparty na Javie, który pozwala na łatwe i szybkie tworzenie oraz konfigurowanie aplikacji. Wspiera on między innymi transakcje bazodanowe, będące głównym celem projektu. Podłączenie się do bazy możliwe było przy użyciu sterownika Spring Boot JDBC 8.

Wykorzystywany w projekcie silnikiem bazy danych było Oracle w wersji 19c.

Spring Boot zawiera również framework Spring Security, który zapewnia bezpieczeństwo uwierzytelniania i autoryzacji aplikacji internetowych. Spring Security pozwala m.in. na tworzenie perspektyw i jasno określa zakres dostępności funkcji różnym typom użytkowników.

Dodatkowo w tworzeniu perspektyw wykorzystywany był silnik template'ów Thymeleaf - pozwolił on na dynamiczne generowanie zawartości stron HTML po stronie serwera dzięki czemu na podstawie jednego pliku HTML uzupełnionego o polecenia Thymeleaf otrzymać mogliśmy wiele różnych wariantów tej samej strony. Pozwolił on również na ponowne używanie raz napisanego kodu - przykładowo, pasek menu na górze strony napisany jest raz i za pomocą odpowiednich poleceń silnika Thymeleaf wklejany jest do pozostałych stron aplikacji, przy czym w locie modyfikowana jest jego zawartość (np. wyświetlany jest różny zestaw linków, w zależności od tego czy zalogowany jest administrator, czy zwykły użytkownik).

Automatyczne wykrywanie oraz dodawanie bibliotek w obrębie wykorzystywanego frameworku, możliwe było dzięki narzędziu Maven, który wspomaga budowanie projektów Java.

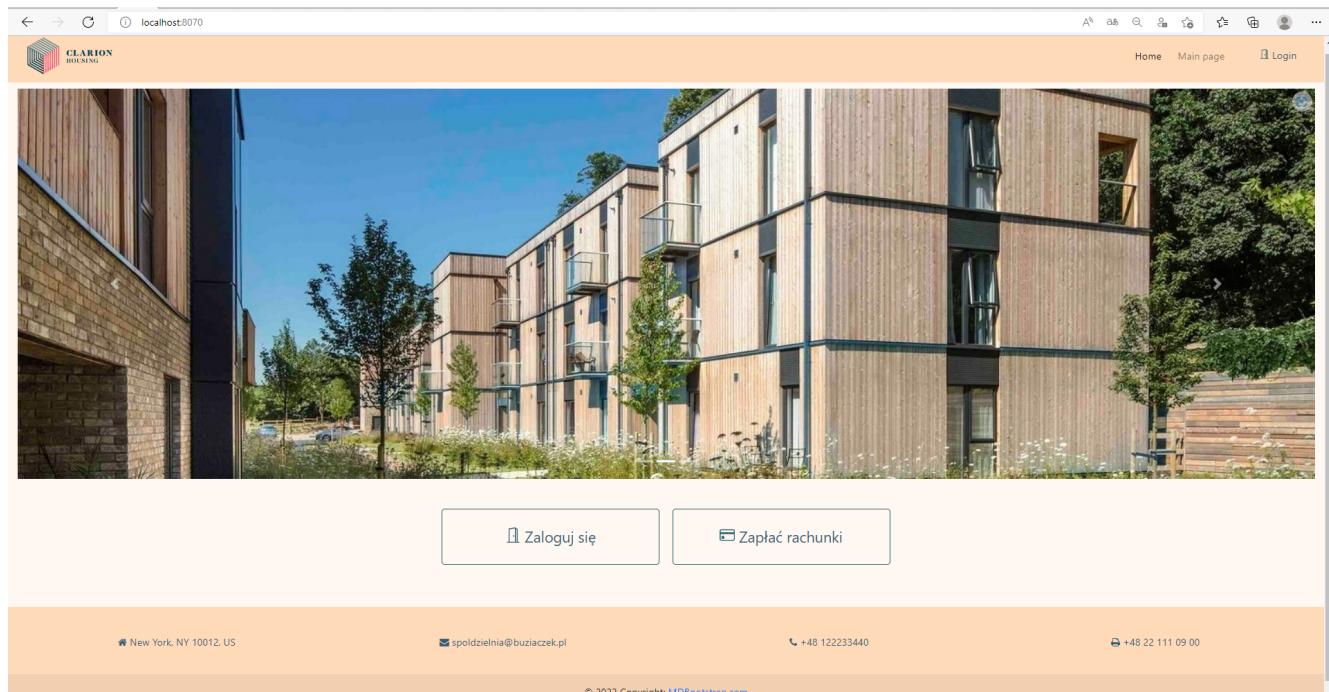
Interfejs graficzny aplikacji zbudowany został w oparciu o framework Bootstrap 5. Jest to biblioteka front-endu, która używa języków HTML, CSS oraz JavaScript. Pozwala na szybsze i efektywniejsze tworzenie aplikacji oraz stron internetowych.

2. Realizacja projektu

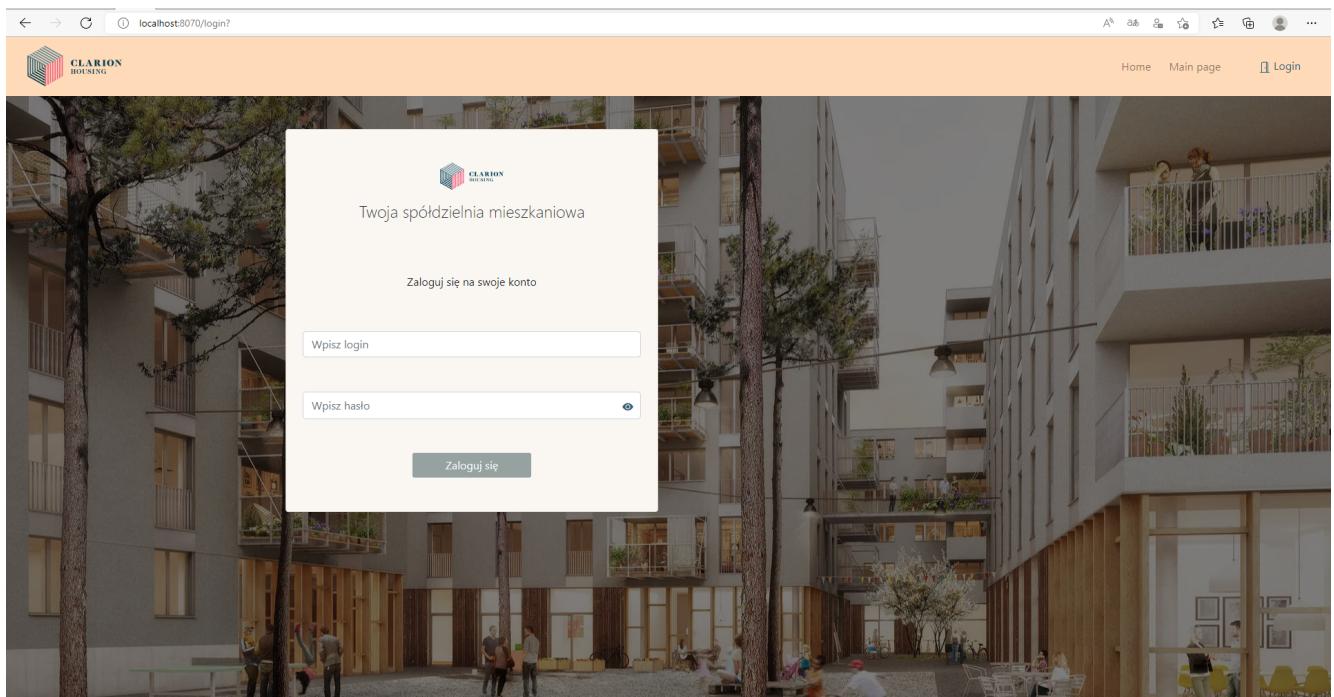
2.1. Perspektywy

W zakresie niniejszej aplikacji, stworzone zostały dwie perspektywy:

- perspektywa “szeroka” - użytkownik z uprawnieniami administratora,
- perspektywa “wąska” - użytkownik z ograniczonymi uprawnieniami.



Zdj. 1. Strona startowa.



Zdj.2. Strona logowania

Dwie powyższej przedstawione strony (*Zdj.1* i *Zdj.2*) nie rozróżniają perspektyw. Oznacza to, że niezależnie od tego, czy użytkownik ma uprawnienia administratora, czy nie, ma pełny dostęp do powyższych stron.

Strona startowa pokazana jest na *Zdj.1*. Na samej górze znajduje się pasek nawigacyjny z przyciskiem przekierowującym do strony logowania.

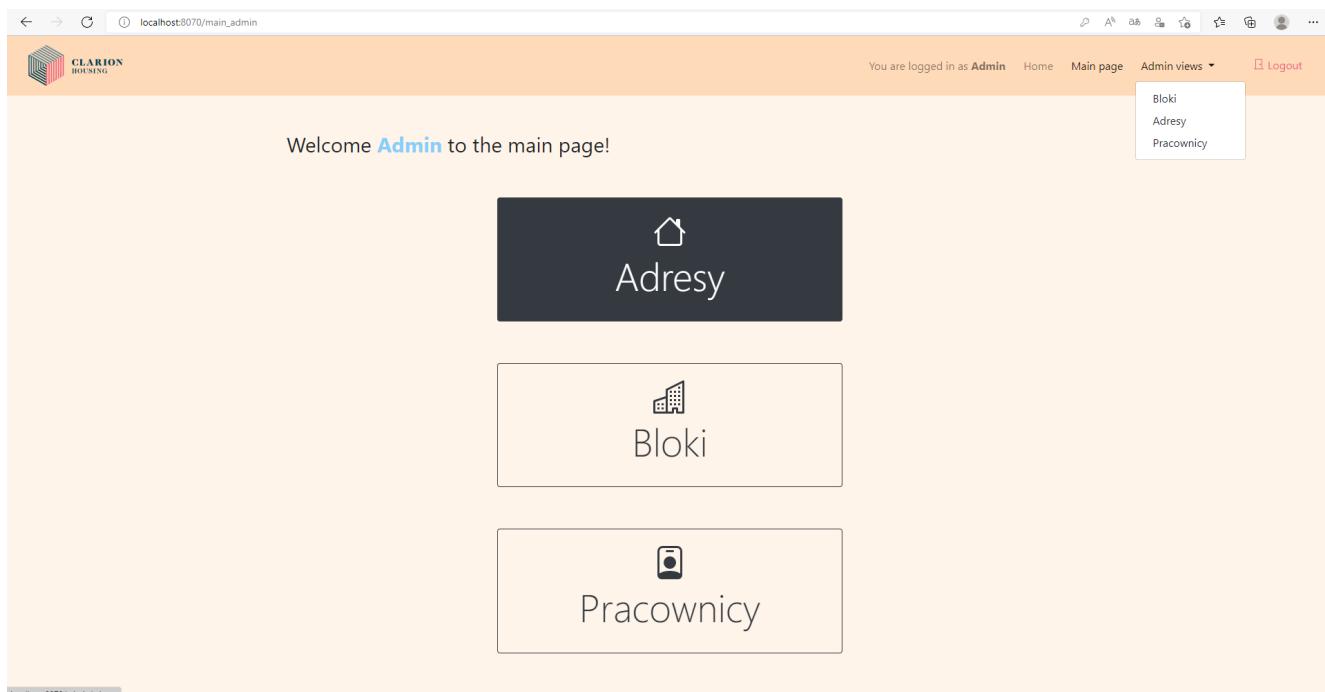
W centralnej części strony, jest karuzela, wraz z przyciskami do przewijania zdjęć. Poniżej znajdują się dwa przyciski przekierowujące użytkownika, albo do strony logowania, albo do strony uiszczenia opłaty za rachunki. W przypadku, gdy użytkownik nie jest jeszcze zalogowany, przycisk ten przekieruje go najpierw do logowania, dopiero później do strony głównej użytkownika.

Strona logowania przedstawiona jest na *Zdj.2*. W górnej jej części, znajduje się ten sam pasek nawigacyjny, co w przypadku strony startowej. Poniżej można wyróżnić specjalny box, zawierający formularz logowania. W przypadku wpisywania hasła, istnieje możliwość jego odkrycia za pomocą przycisku w postaci oka.

2.1.1. Perspektywa “szeroka”

Perspektywa “szeroka” obejmuje tylko i wyłącznie administratora bazy danych spółdzielni mieszkaniowej. Ma on dostęp do danych z tabel *Adresy*, *Blok* oraz *Pracownicy*, których zawartość może przeglądać, modyfikować oraz może dodawać do nich nowe rekordy lub usuwać istniejące.

Poniżej przedstawione strony dostępne są tylko z perspektywy administratora.



Zdj.3. Strona główna admina.

Zdj.3 przedstawia stronę główną administratora. W górnej części umieszczony został pasek nawigacyjny, rozszerzony o dodatkowe elementy - informację o typie zalogowanego użytkownika oraz menu rozwijane pozwalające na nawigację po dostępnych stronach.

W centralnej części strony znajdują się 3 przyciski, przekierowujące administratora do wybranych tabel.

Poniższe zdjęcia przedstawiają strony z tabelami oraz z umożliwionymi działaniami w ich obrębie.



You are logged in as **Admin** [Home](#) [Main page](#) [Admin views](#) [Logout](#)

Pracownicy admin view

ID	Imię	Drugie imię	Nazwisko	Pesel	Data urodzenia	Numer telefonu	Plec	Numer konta	Data zatrudnienia	Data zwolnienia	Ulica	Numer domu	Numer lokalu	Miasto	Kod pocztowy	Actions
1	Andrzej	Maria	Komar	70121211223	12-12-1970	543123123	M	685676341148791569100405	01-08-2014		Małego Pułku	11A	14	Wrocław	02-213	 Update Delete
2	Marzena		Komar	81121211223	12-12-1981	987654321	K	685676341148791569100405	01-10-2018	14-06-2022	Małego Pułku	11A	14	Wrocław	02-213	 Update Delete
3	Agata		Bąk	88010544112	05-01-1988	421234456	K	969362028491556308326614	01-01-2014		Małego Pułku	11B	48	Wrocław	02-213	 Update Delete
4	Paweł		Krawczyk	65040912345	09-04-1965	900800700	M	459015729155365510709658	01-01-2005		Bohaterów	10		Wrocław	02-555	 Update Delete

Add

Zdj.4. Strona admina wyświetlająca tabelę pracowników.



You are logged in as **Admin** [Home](#) [Main page](#) [Admin views](#) [Logout](#)

Add new pracownik

Attribute	Input
Imię	<input type="text"/>
Drugie imię	<input type="text"/>
Nazwisko	<input type="text"/>
Pesel	<input type="text"/>
Data_urodzenia	<input type="text"/> dd.mm.rrrr
Numer telefonu	<input type="text"/>
Plec	<input type="button" value="M"/>
Numer konta	<input type="text"/>

Add Cancel

Zdj.5. Dodawanie nowego pracownika.

localhost:8070/admin/pracownicy/1/update

Attribute	Input
ID	<input type="text" value="1"/>
Imię	<input type="text" value="Andrzej"/>
Drugie imię	<input type="text" value="Maria"/>
Nazwisko	<input type="text" value="Komar"/>
Pesel	<input type="text" value="70121211223"/>
Data urodzenia	<input type="text" value="1970-12-12 00:00:00"/>
Numer telefonu	<input type="text" value="543123123"/>
Plec	<input type="text" value="M"/>

Update Cancel

Zdj.6. Modyfikacja pracownika istniejącego w bazie.

ID	Liczba pieter	Liczba lokali	Rok powstania	Winda	Liczba klatek	Adres ID	Actions
1	4	75	2011	4	4	2	<input type="button"/> Update <input type="button"/> Delete
2	6	96	2011	4	4	3	<input type="button"/> Update <input type="button"/> Delete
3	4	70	2012	4	4	4	<input type="button"/> Update <input type="button"/> Delete

Add

Zdj.7. Strona admina wyświetlająca tabelę bloki.

Add new blok

Attribute	Input
Liczba pięter	<input type="text" value="0"/>
Liczba lokali	<input type="text" value="0"/>
Rok powstania	<input type="text" value="0"/>
Winda	<input type="text" value="0"/>
Liczba klatek	<input type="text" value="0"/>
Adresy ID	<input type="text" value="0"/>

[⊕ Add](#)[⊗ Cancel](#)

Zdj.8. Dodawanie nowego bloku.

Update blok

Attribute	Input
ID	<input type="text" value="1"/>
Liczba pięter	<input type="text" value="4"/>
Liczba lokali	<input type="text" value="75"/>
Rok powstania	<input type="text" value="2011"/>
Winda	<input type="text" value="4"/>
Liczba klatek	<input type="text" value="4"/>
Adresy ID	<input type="text" value="2"/>

[⊕ Update](#)[⊗ Cancel](#)

Zdj.9. Modyfikacja bloku istniejącego w bazie.

Adresy admin view

ID	Ulica	Numer domu	Numer lokalu	Miasto	Kod pocztowy	Actions
3	Krakowska	1B		Wrocław	01-112	
4	Krakowska	2		Wrocław	01-112	
5	Krakowska	3		Wrocław	01-112	
6	Małego Pułku	11A	14	Wrocław	02-213	
7	Małego Pułku	11B	48	Wrocław	02-213	
8	Bohaterów	10		Wrocław	02-555	
9	Fabryczna	1		Bielchatów	91-111	

Add

Zdj.10. Strona admina wyświetlająca tabelę adresów.

Add new adres

Attribute	Input
Ulica	<input type="text"/>
Numer domu	<input type="text"/>
Numer lokalu	<input type="text"/>
Miasto	<input type="text"/>
Kod pocztowy	<input type="text"/>

Add

Cancel

Zdj.11. Dodawanie nowego adresu.

Update adres

Attribute	Input
ID	<input type="text" value="7"/>
Ulica	<input type="text" value="Małego Pułku"/>
Numer domu	<input type="text" value="11A"/>
Numer lokalu	<input type="text" value="23"/>
Miasto	<input type="text" value="Wrocław"/>
Kod pocztowy	<input type="text" value="02-213"/>

 Update Cancel

Zdj. 12. Modyfikacja adresu istniejącego w bazie.

Zdj. 4, Zdj. 7 oraz Zdj. 10 przedstawiają strony wyświetlające zawartość tabel. Po prawej stronie w każdym wierszu widnieją dwa przyciski służące do modyfikacji lub usunięcia danego rekordu z bazy danych.

Pod każdą z tych tabel umieszczone zostały przyciski przekierowujące do strony tworzenia nowego rekordu.

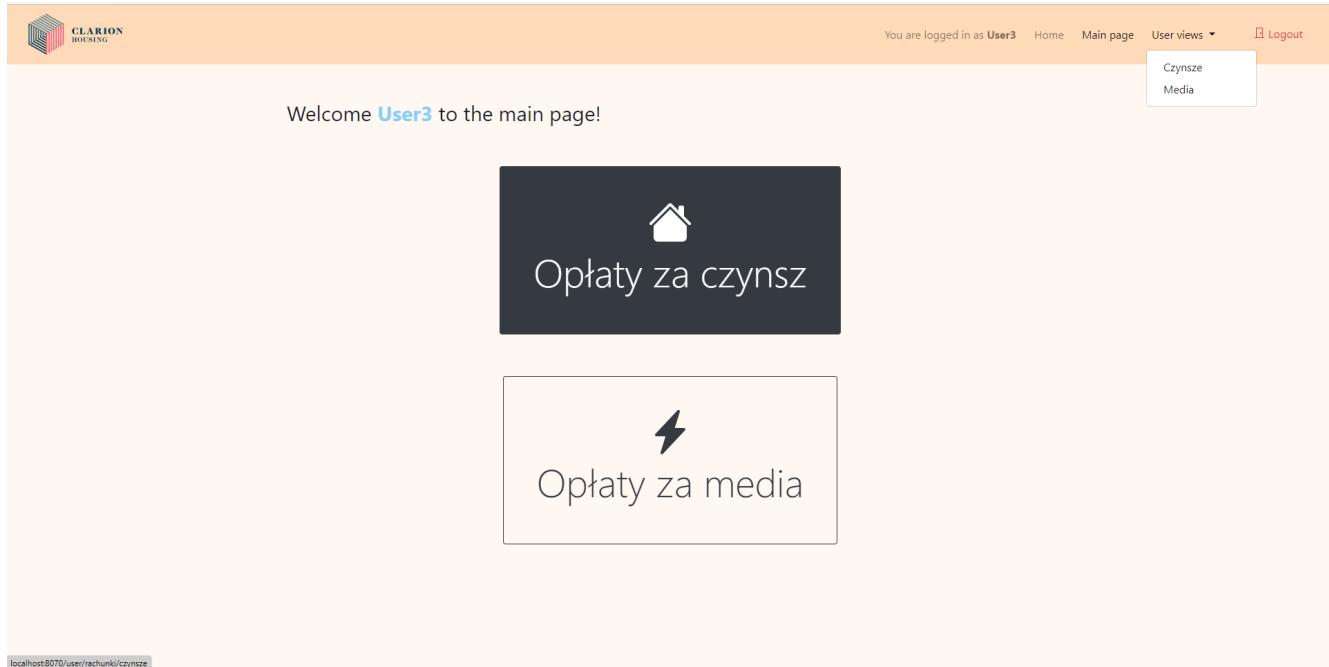
Strony tworzenia rekordów zilustrowane są na Zdj. 5, Zdj. 8 oraz Zdj. 11. Każdy input ma typ odpowiadający danemu atrybutowi w bazie danych.

Pod każdą tabelą tworzenia umieszczone zostały dwa przyciski. Przycisk *Add* dodaje do bazy utworzony rekord, natomiast przycisk *Cancel* przekierowuje użytkownika z powrotem na stronę wyświetlającą tabelę oraz nie dodaje nowego wiersza do tabeli w bazie.

Zdj. 6, Zdj. 9 oraz Zdj. 12 przedstawiają strony modyfikacji istniejących w bazie rekordów. Po zmianie danych użytkownik może je zatwierdzić przy użyciu przycisku *Update* lub może nie zapisać zmian i wrócić na stronę wyświetlającą tabelę za pomocą przycisku *Cancel*.

2.1.2. Perspektywa “wąska”

Perspektywa “wąska” obejmuje użytkowników aplikacji, którzy w bazie danych istnieją jako *Mieszkańcy*. W obrębie aplikacji, ten typ użytkownika ma wgląd tylko i wyłącznie w swoje dane dotyczące rachunków za opłaty i media. *Mieszkaniec* może również uiścić opłatę za niezapłacone rachunki, obsługując w ten sposób operację modyfikacji danych w bazie.



Zdj.13. Strona główna użytkownika.

Powyższy screen(Zdj.13) przedstawia stronę główną z perspektywy użytkownika z ograniczonym dostępem do modyfikacji bazy. Podobnie jak w stronie głównej administratora, tu również na pasku administracyjnym umieszczona została lista rozwijana pozwalająca na nawigację po aplikacji.

W centralnej części strony znajdują się dwa przyciski przekierowujące użytkownika do tabel z danymi o jego opłatach za czynsz oraz opłatach za media.

Rachunki - czynsze user view

Numer lokalu	Data wystawienia rachunku	Wysokość czynszu [zł]	Oplaty dodatkowe [zł]	Kwota do zapłaty [zł]	Za okres od	Za okres do	Oplacony?	Actions
45	2022-03-04 00:00:00	800.0	0.0	800.0	2022-02-01 00:00:00	2022-02-28 00:00:00	N	
45	2022-02-04 00:00:00	800.0	0.0	800.0	2022-01-01 00:00:00	2022-01-31 00:00:00	T	
45	2022-02-04 00:00:00	800.0	0.0	800.0	2021-12-01 00:00:00	2021-12-31 00:00:00	T	

Zdj.14.Strona użytkownika wyświetlająca tabelę rachunków za czynsze.

Rachunki - media user view

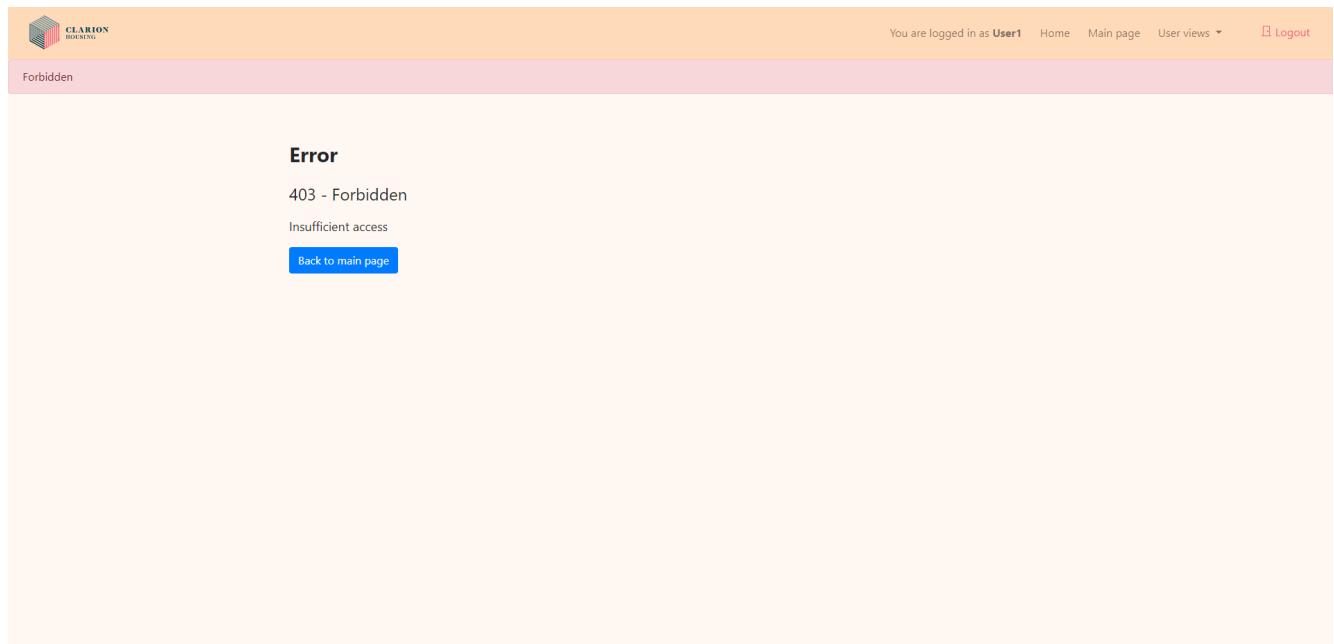
Numer lokalu	Data wystawienia rachunku	Kwota do zapłaty [zł]	Za okres od	Za okres do	Medium	Oplacony?	Actions
45	2022-03-04 00:00:00	221.0	2022-02-01 00:00:00	2022-02-28 00:00:00	Prąd	T	
45	2022-02-04 00:00:00	231.0	2022-01-01 00:00:00	2022-01-31 00:00:00	Prąd	T	
45	2022-02-04 00:00:00	228.0	2021-12-01 00:00:00	2021-12-31 00:00:00	Prąd	T	
45	2022-03-04 00:00:00	148.42	2022-02-01 00:00:00	2022-02-28 00:00:00	Gaz	T	
45	2022-02-04 00:00:00	173.02	2022-01-01 00:00:00	2022-01-31 00:00:00	Gaz	T	
45	2022-02-04 00:00:00	120.54	2021-12-01 00:00:00	2021-12-31 00:00:00	Gaz	T	

Zdj.15.Strona użytkownika wyświetlająca tabelę rachunków za media.

Zdj.14 oraz *Zdj.15* obrazują tabele z bazy dotyczące tylko i wyłącznie rachunków aktualnie zalogowanego użytkownika. W przypadku, gdy któryś z rachunków nie jest opłacony, przy jego wierszu w tabeli widnieje przycisk, który umożliwia uiszczenie opłaty. W momencie jego przyciśnięcia, użytkownik modyfikuje bazę poprzez zmianę atrybutu *czy_oplacony* na wartość T . W rzeczywistej aplikacji po kliknięciu w przycisk *Payment* użytkownik przekierowywany byłby do strony pośrednika płatności, a wartość *czy_oplacony* w bazie ulegałaby zmianie dopiero po otrzymaniu informacji o pomyślnym sfinalizowaniu transakcji.

2.2. Obsługa błędów

W przypadku wystąpienia błędu, czy to po stronie użytkownika (błędy z kodem 4XX), czy po stronie serwera (błędy z kodem 5XX) użytkownik przekierowywany jest na stronę błędu dostosowaną stylistycznie do reszty aplikacji. Z poziomu strony błędu użytkownik ma możliwość powrotu do strony głównej aplikacji poprzez wcisnięcie przycisku *Back to the main page* lub poprzez skorzystanie z paska nawigacyjnego na górze strony.



Zdj.16. Przykładowa obsługa błędu w przypadku próby dostępu do niedozwolonej strony - w tym przypadku próba dostępu do strony admina z poziomu użytkownika.

Błędy wynikające bezpośrednio z naruszenia założonej logiki aplikacji bazodanowej wyświetlają dodatkowo komunikat informujący co było przyczyną błędu.

Attribute	Input
Liczba pięter	6
Liczba lokali	100
Rok powstania	1999
Winda	4
Liczba klatek	4
Adresy ID	9999999

[Add](#) [Cancel](#)

Zdj.17a. Próba dodania bloku z nieistniejącym w bazie adresem.

Given Adres_id does not exist

Error
400 - Bad Request

[Back to main page](#)

Zdj.17b. Strona błędu wyświetlając komunikat informujący o naruszeniu założonej logiki aplikacji