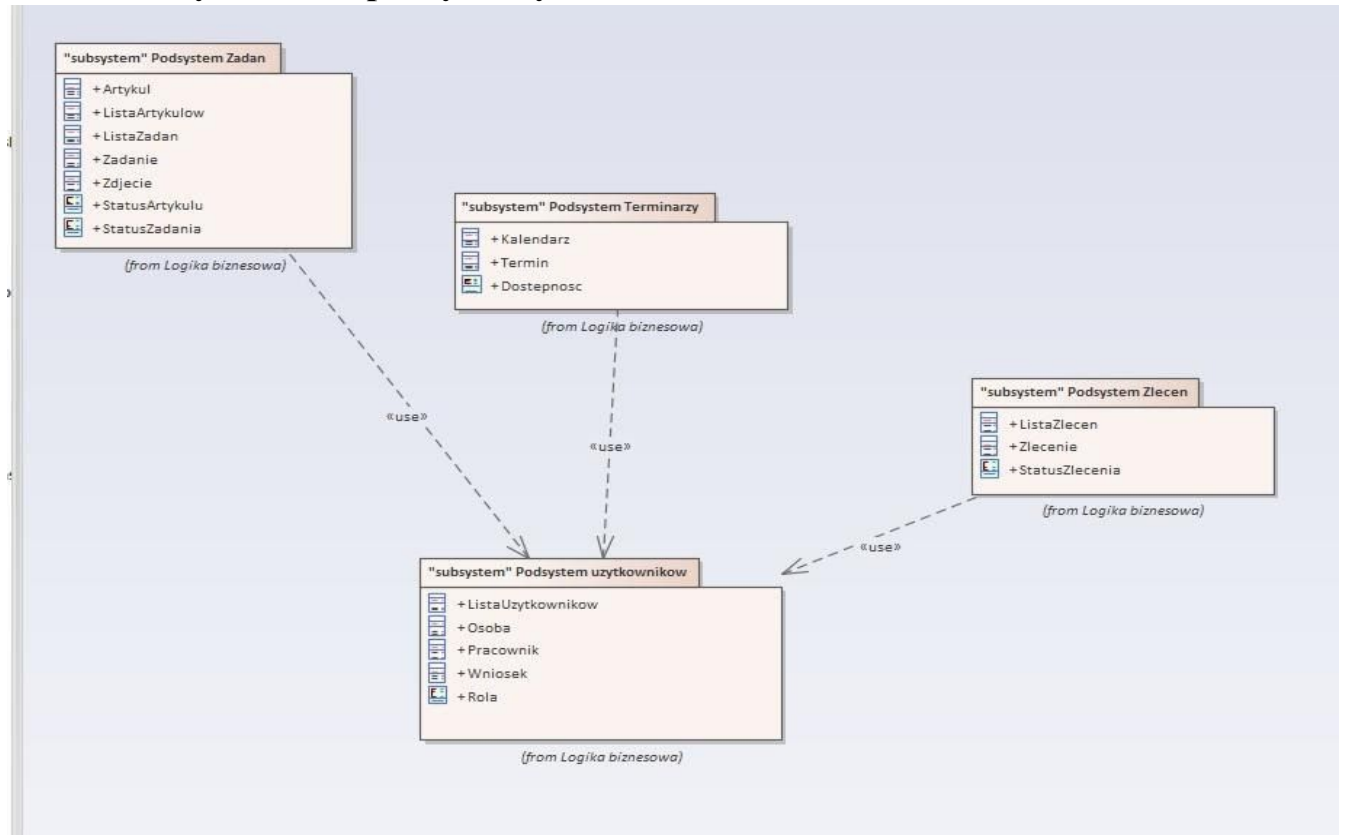


Elementy projektowania

Temat: Dziennik Politechniki

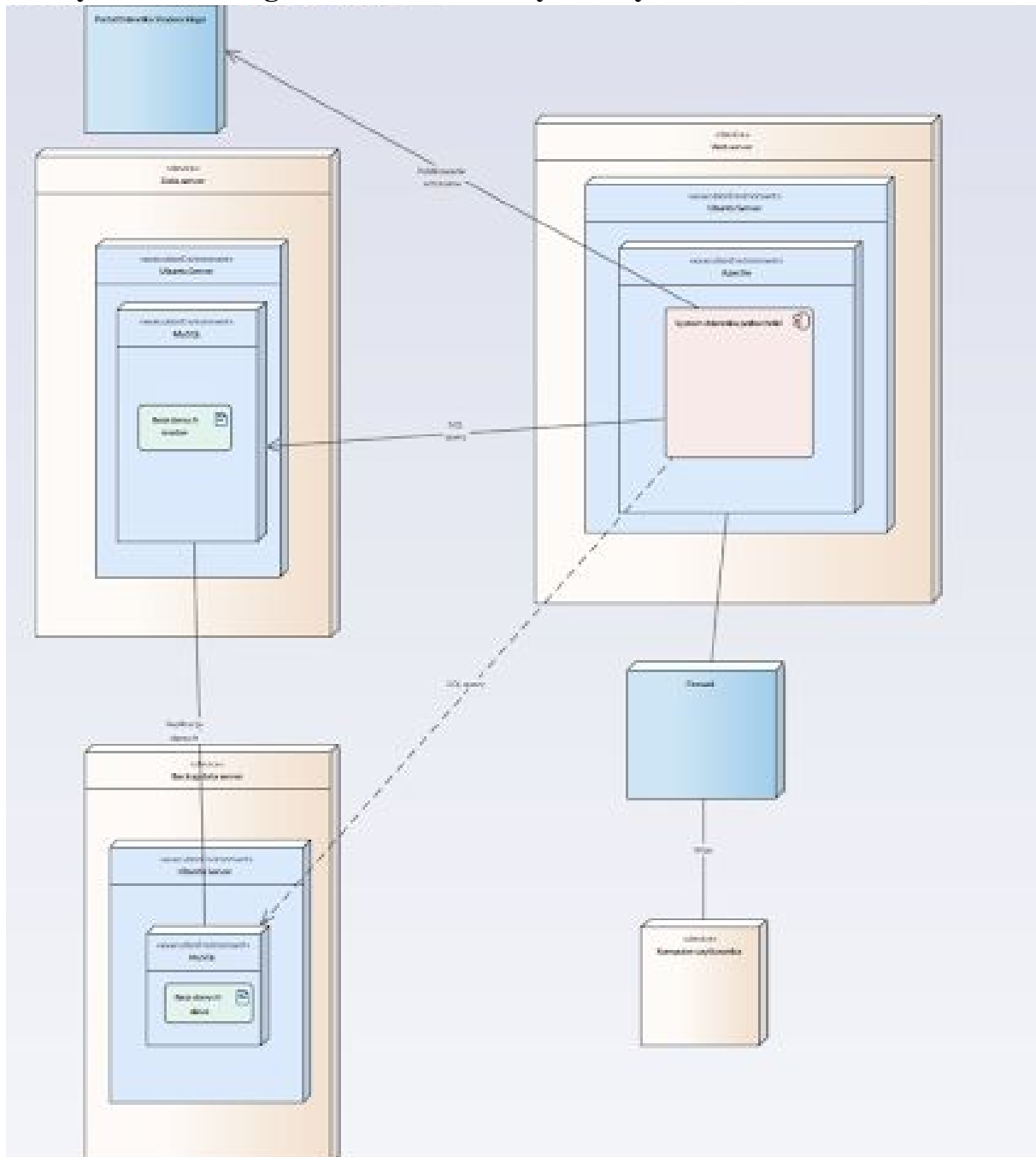
Autorzy: Konrad Paniec, Andrei Shastak, Kseniya Shastak

1. Podział systemu na podsystemy



Podsystem	Opis
Użytkowników	Zawiera klasy odpowiedzialne za składanie wniosków, zarządzanie statusami oraz główną klasę Osoba .
Zlecenia	Zawiera klasy odpowiedzialne za składanie zleceń przez Zleceniodawcę.
Zadania	Zawiera klasy odpowiedzialne za tworzenie, przydzielanie i akceptowanie zadań, dodawanie artykułów oraz zdjęć.
Terminarze	Zawiera klasy odpowiedzialne za obsługę terminarza dostępności oraz rezerwacji sprzętu.

2. Wybór technologii i architektura fizyczna systemu

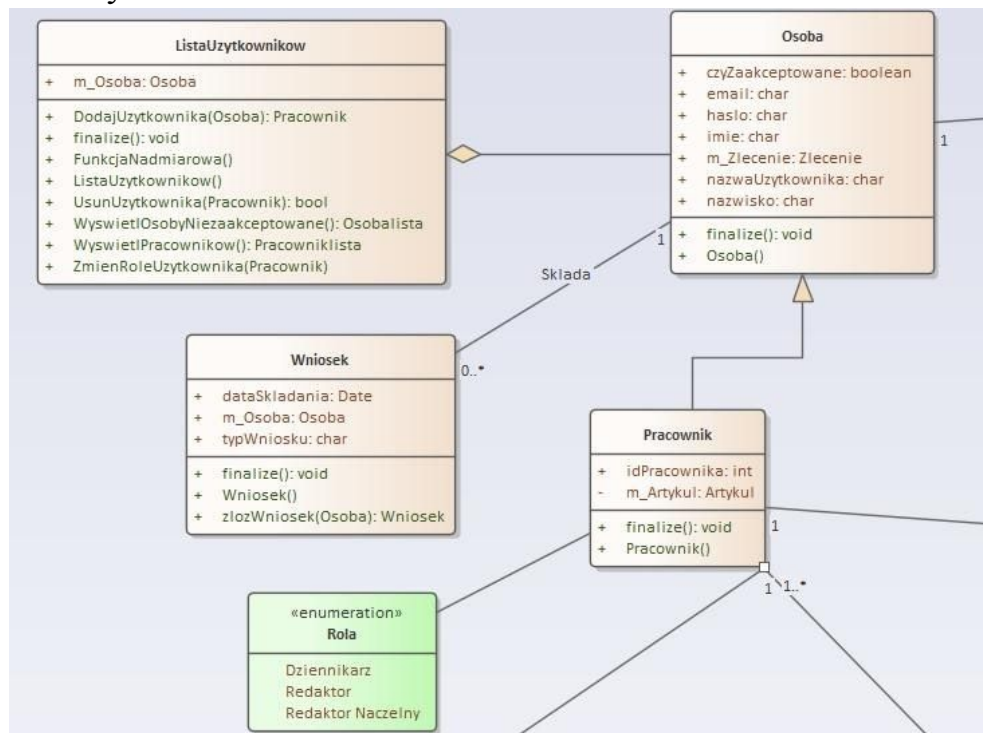


System dla Dziennika Politechniki będzie aplikacją webową napisaną w języku Java EE w wersji 8 z wykorzystaniem open-source'owego framework'u Spring Web MVC. System będzie działał na serwerze Apache, ze względu na to że jest darmowy, open source, prosty w konfiguracji oraz szybki i niezawodny. Użytkownicy będą mogli wysyłać do niego żądania za pośrednictwem przeglądarki internetowej (Mozilla Firefox, Google Chrome, Opera). Użytkownicy będą łączyć się serwerem Apache z użyciem protokołu https oraz firewall w celu zwiększenia bezpieczeństwa. Podstawowym zadaniem firewall będzie filtrowanie połączeń wchodzących i wychodzących oraz tym samym odmawianie żądań dostępu uznanych za niebezpieczne. System może wysłać żądanie o publikację danego artykułu na portal Dziennika Politechniki, który jest oddzielną aplikacją. Dane będą przechowywane na oddzielnych serwerach (w celu zwiększenia bezpieczeństwa) w relacyjnej bazie danych MySQL ze względu na prostotę w obsłudze oraz wysoką dostępność. Baza danych będzie działała na zasadzie master-slave, gdzie slave będzie na bieżąco replikował dane z master'a. Dzięki temu gdy jeden z serwerów ulegnie awarii, system dziennika politechniki nadal będzie miał dostęp do kluczowych dla jego działania danych. Odpowiednie dane przez system mogą być uzyskane za



4. Projekt logiki biznesowej

Podsystem użytkowników



Automatycznie wygenerowany kod :

Plik	Kod
Osoba	<pre>/** * @author student * @version 1.0 * @created 15-Jan-2020 08:46:36 */ public class Osoba { public char imie; public char nazwisko; public char haslo; public char nazwaUzytkownika; public char email; public boolean czyZaakceptowane; public Zlecenie m_Zlecenie; public Osoba(){ }</pre>

	<pre> public void finalize() throws Throwable { } } //end Osoba </pre>
Pracownik	<pre> /** * @author student * @version 1.0 * @created 15-Jan-2020 08:46:36 */ public class Pracownik extends Osoba { public int idPracownika; private Artykul m_Artykul; public Pracownik(){ } public void finalize() throws Throwable { super.finalize(); } } //end Pracownik </pre>
Wniosek	<pre> /** * @author Kseniya * @version 1.0 * @created 15-Jan-2020 08:46:37 */ public class Wniosek { public Date dataSkladania; public char typWniosku; public Osoba m_Osoba; public Wniosek(){ } public void finalize() throws Throwable { </pre>

	<pre> } /** * * @param osoba */ public Wniosek zlozWniosek(Osoba osoba) { return null; } } //end Wniosek </pre>
Rola	<pre> /** * @author Manquak * @version 1.0 * @created 15-Jan-2020 08:46:36 */ public enum Rola { Dziennikarz, Redaktor, Redaktor Naczelny; public Pracownik m_Pracownik; } </pre>
Lista Użytkowników	<pre> /** * @author Einheriarne * @version 1.0 * @created 15-Jan-2020 08:46:36 */ public class ListaUzytkownikow { public Osoba m_Osoba; public ListaUzytkownikow(){ } public void finalize() throws Throwable { } } /** * * @param Osoba </pre>

```

*/
public Pracownik DodajUzytkownika(Osoba
Osoba){
    return null;
}

/**
 *
 * @param Pracownik
 */
public bool UsunUzytkownika(Pracownik
Pracownik){
    return null;
}

public FunkcjaNadmiarowa(){
    return null;
}

/**
 *
 * @param Pracownik
 */
public ZmienRoleUzytkownika(Pracownik
Pracownik){

}

public Pracowniklista
WyswietlPracownikow(){
    return null;
}

public Osobalista
WyswietlOsobyNiezaakceptowane(){
    return null;
}
} //end ListaUzytkownikow

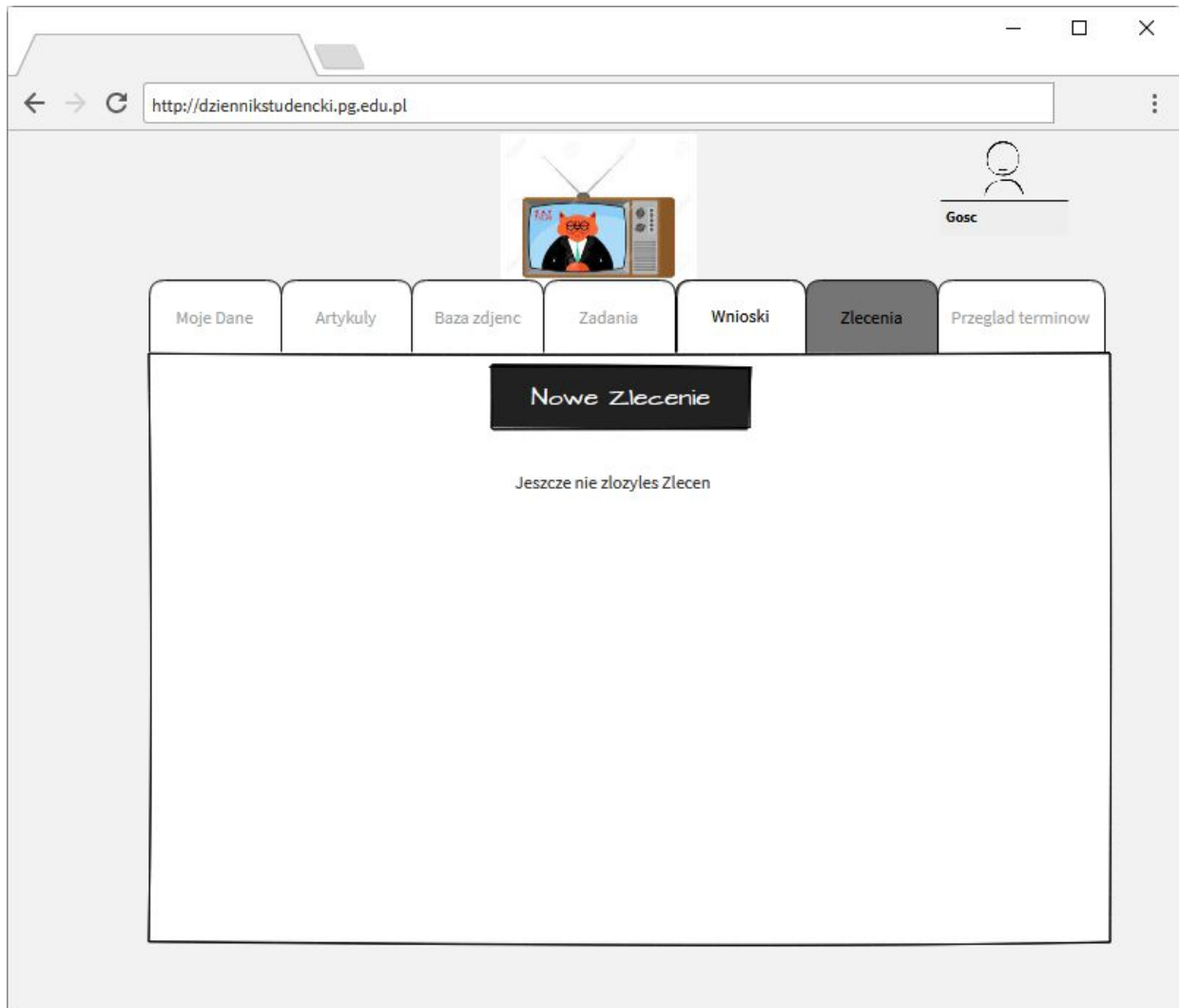
```

5. Projekt interfejsu użytkownika

PU zgłoszenie zlecenia.

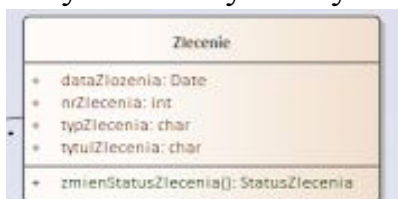
Slajd 1.

Gość składa nowe zlecenie. Dostępne zakładki to zlecenia i wnioski, ponieważ użytkownik posiada status gościa. Aktywna jest zakładka Zlecenia. Na liście zleceń nie ma żadnego zlecenia, bo tworzymy pierwsze zlecenie. Następny krok -- naciśnięcie przycisku “Nowe Zlecenie”



Slajd 2.

Po wciśnięciu przycisku pojawia się okno służące do uzupełnienia danych osobowych oraz statusu zleceniodawcy. Zleceniodawca ma miejsce na wpisanie treści zlecenia. Tytuł oraz typ zlecenia nie muszą być oddzielone jako osobne pola, ponieważ zdecydowaliśmy że użytkownik podaje wszelkie informacje w treści.



tytułZlecenia z klasy Zlecenie jest wykorzystany jako treść zlecenia w UI i jest ograniczony rozmiarem 100 symboli.

Po wykonaniu wszystkich działań, gość wysłał zlecenie za pomocą przycisku “wyślij”

The screenshot shows a web browser window with the URL <http://dziennikstudencki.pg.edu.pl>. The page features a navigation bar with tabs: 'Moje Dane', 'Artykuły', 'Baza zdjenc', 'Zadania', 'Wnioski', 'Zlecenia' (selected), and 'Przegląd terminow'. A user profile icon labeled 'Gosc' is in the top right. The main content area is titled 'Nowe Zlecenie'. It contains a form with the following fields: 'Imie' (required), 'Nazwisko' (required), 'Username', 'Password' (required), and a 'Jestem' dropdown menu with options: 'Gosciem', 'Dziennikarzem', 'Redaktorem', and 'Redaktorem DS.Finansow'. A text area contains the text 'Proszę o przyjęcie, nadaniu statusu... bla bla bla 100 znakow'. Below the text area is a checkbox labeled 'Zatwierdzam dane' which is checked. A 'Wyslij' button is at the bottom right. A note at the bottom left states '* pole obowiązkowe'.

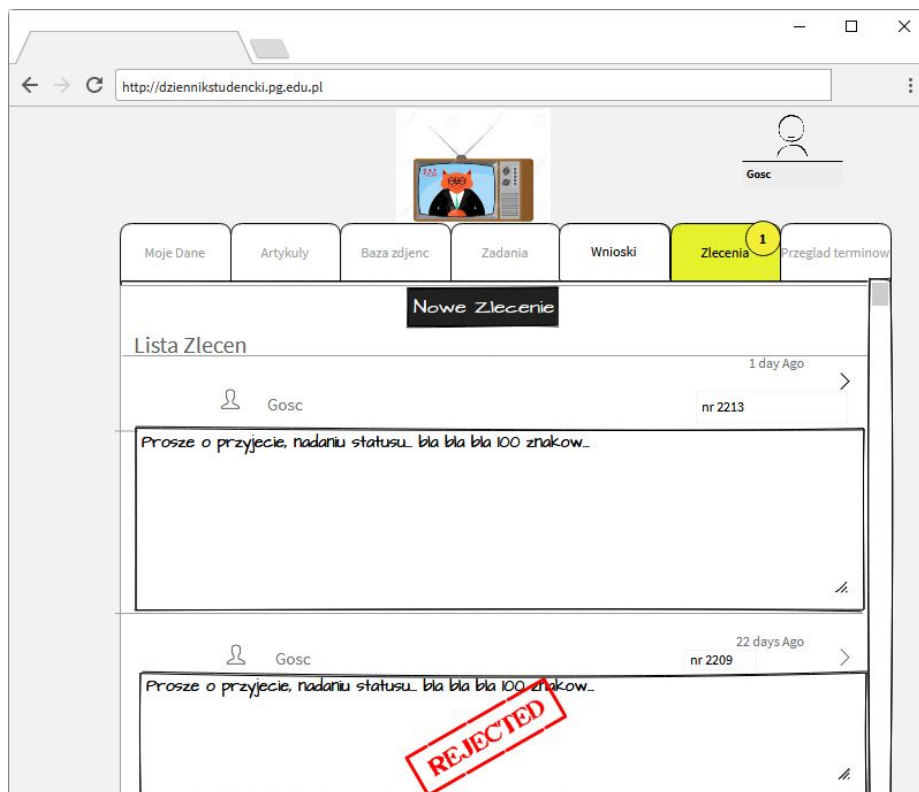
Slajd 3.

Po wysłaniu zlecenie jest dodane do listy zleceń. Zlecenia otrzymują swój numer i datę.

The screenshot shows the same web browser window, but the 'Zlecenia' tab is selected, displaying the 'Lista Zleceń' page. The page has a 'Nowe Zlecenie' button at the top. Below it, the list shows a single entry for 'Gosc' with the number 'nr 2209' and a timestamp '1 min Ago'. The text of the request is visible in a box: 'Proszę o przyjęcie, nadaniu statusu... bla bla bla 100 znakow...'. A right arrow icon is next to the entry.

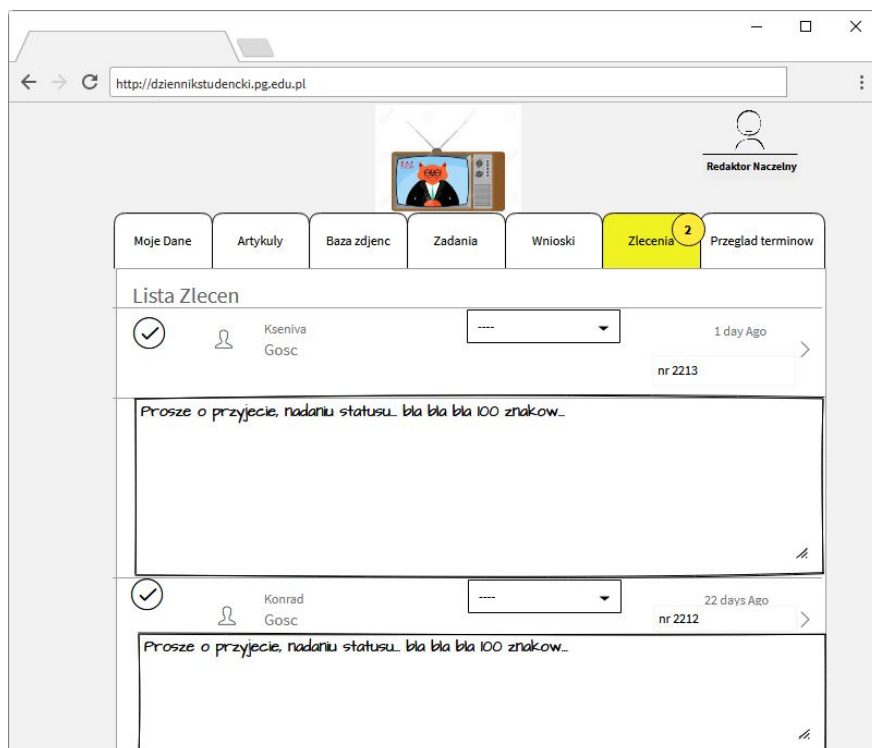
Slajd 4.

Gość widzi listę własnych zleceń, jedno z nich było rozpatrzone i jego zleceniu został nadany status rejected.



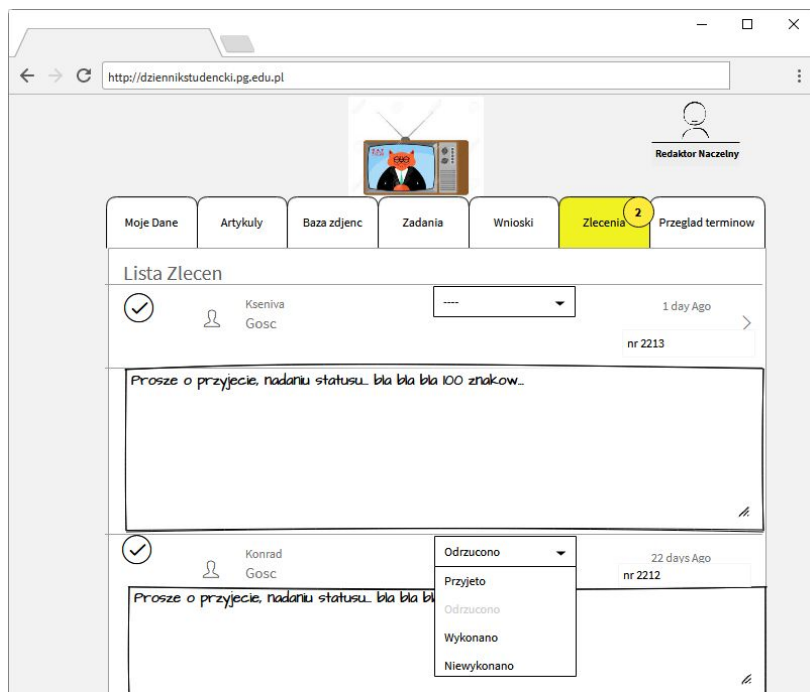
Slajd 5.

Widok na te same zlecenia ze strony Redaktora Naczelnego. Redaktor widzi listę zgłoszonych, nieopracowanych zleceń.



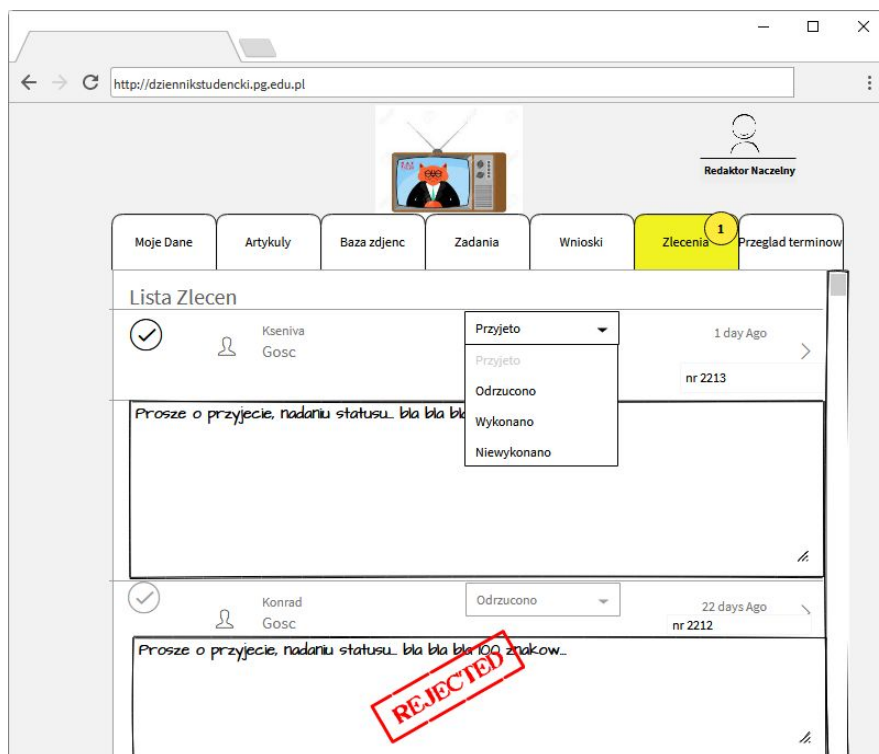
Slajd 6.

Redaktor naczelny wybrał status “odrzucono” dla jednego ze zleceń, ale jeszcze nie zatwierdził swojej decyzji.



Slajd 7.

Redaktor naczelny zatwierdził poprzednie zlecenie i wybrał status “przyjęto” dla nowego, ale jeszcze nie zatwierdził swojej decyzji.



Slajd 8.

Wszystkie zlecenia są rozpatrzone i zostały zatwierdzone.

The screenshot shows a web browser window with the URL <http://dziennikstudencki.pg.edu.pl>. The page features a navigation bar with tabs: Moje Dane, Artykuły, Baza zdjenc, Zadania, Wnioski, Zlecenia (selected), and Przegląd terminow. Below the navigation bar, there is a header section with a logo of a television set and a user profile labeled "Redaktor Naczelny".

The main content area displays a list of requests under the heading "Lista Zleceń". Each request entry includes a status icon (checkmark or X), a user profile, a status dropdown menu, a date, and a red stamp indicating the final decision.

Status	User	Status	Date	Decision
✓	Kseniwa Gosc	Przyjęto	1 day Ago	APPROVED
✓	Konrad Gosc	Odrzucono	22 days Ago	REJECTED

Each request entry also includes a text box with the content: "Proszę o przyjęcie, nadaniu statusu... bla bla bla 100 znakow..." and a reference number (nr 2213 and nr 2212 respectively).