**Medical Appointment Booking System**

Greyg Władysiak, Hubert Mądry

K32

1. **Business description of the system**

The medical appointment booking system was created with the aim of streamlining the process of scheduling visits at healthcare facilities. The main motivation behind developing the software was the need to automate and simplify communication between patients and medical staff, as well as to reduce the number of phone registrations and queues at clinics.

The system allows patients to quickly and conveniently book appointments with selected doctors, browse available specializations, and manage their reservations. Doctors have access to their appointment schedules, can view the list of patients, and manage their availability. The system administrator has tools for managing users and monitoring the overall operation of the system.

1. **The most important functions of the system are**
2. Registration and login for patients and doctors
3. Browsing available doctors and specializations
4. Booking and cancelling appointments by patients
5. Viewing the appointment schedule by doctors
6. Communication with the administrator
7. Secure management of user data

1. **Functional requirements**

|  |  |  |
| --- | --- | --- |
| **ID** | **Function Name** | **Description** |
| F01 | User Registration | A user (patient or doctor) can create an account by providing required data. |
| F02 | User Login | A user can log into the system using a username and password. |
| F03 | Browse Doctors and Specializations | A patient can view a list of doctors and their specializations. |
| F04 | Appointment Booking | A patient can select and book an appointment. |
| F05 | Appointment Cancellation | A patient can cancel a previously booked appointment. |
| F06 | View Doctor Schedule | A doctor can view their appointment schedule. |
| F07 | View Patient List | A doctor can view the list of patients booked for appointments. |
| F08 | Manage Availability | A doctor can manage their available appointment slots. |
| F09 | Message to Administrator | A user can send a message to the system administrator. |
| F10 | Edit Account | A user can edit their account details. |
| F11 | Delete Account | A user can delete their account. |

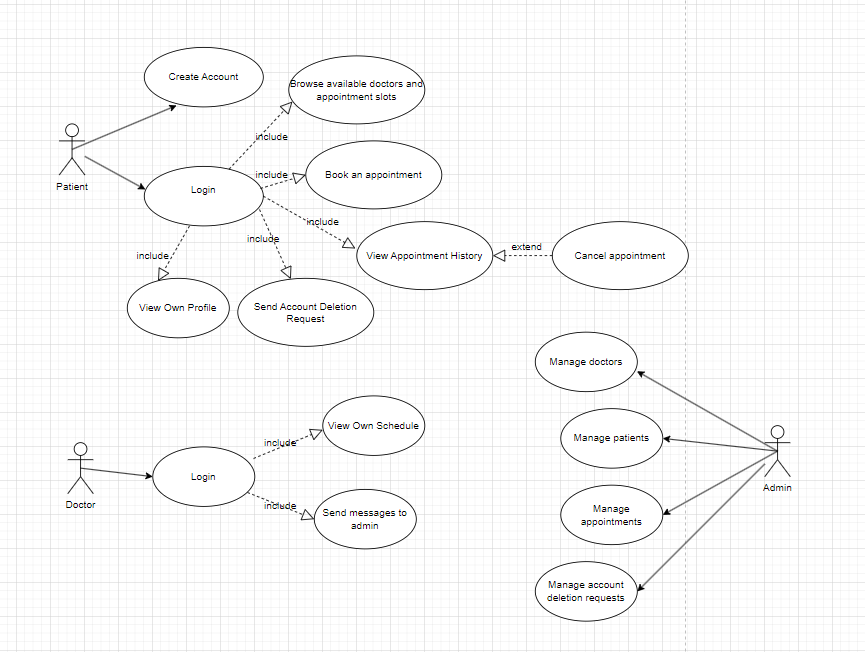
1. **Non-functional requirements**

|  |  |  |
| --- | --- | --- |
| **ID** | **Category** | **Requirement** |
| NF01 | Security | User data is securely stored (e.g., encrypted passwords). |
| NF02 | Security | Only authorized users have access to data. |
| NF03 | Performance | The system handles simultaneous logins and bookings without noticeable delays. |
| NF04 | Availability | The system is accessible 24/7 via a web browser. |
| NF05 | Usability | The user interface is simple, clear, and user-friendly for all age groups. |

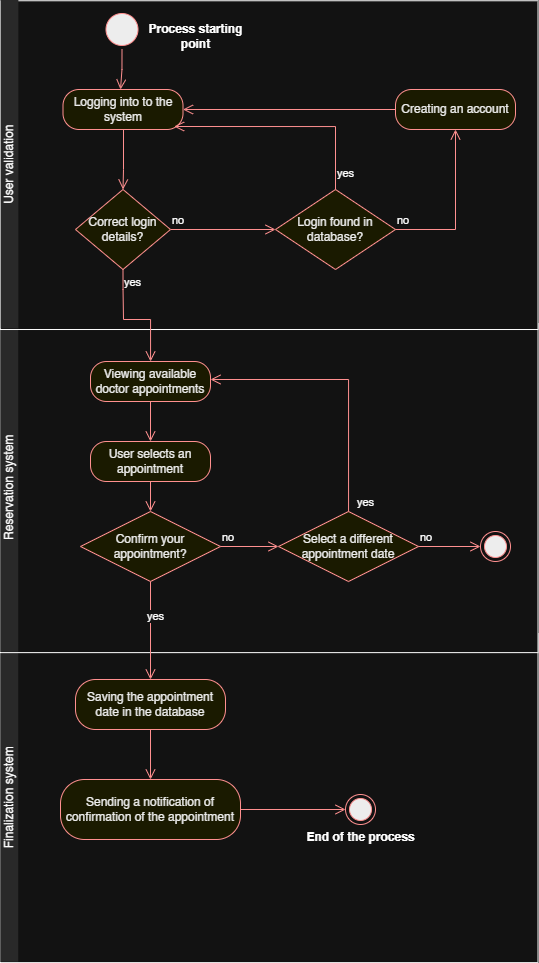
1. **Use cases**

|  |  |  |
| --- | --- | --- |
| **ID** | **Role** | **Use Case Description** |
| US01 | Patient | As a patient, I want to register so I can book appointments online. |
| US02 | Patient | As a patient, I want to browse doctors and their specializations to choose the right specialist. |
| US03 | Patient | As a patient, I want to book an appointment for a specific date and time. |
| US04 | Patient | As a patient, I want to cancel an appointment if I can't attend. |
| US05 | Doctor | As a doctor, I want to view my schedule to see my upcoming appointments. |
| US06 | Doctor | As a doctor, I want to see patient details for upcoming appointments. |
| US07 | Administrator | As an administrator, I want to manage users and monitor the system’s status. |
| US08 | User | As a user, I want to contact the administrator in case of any issues. |

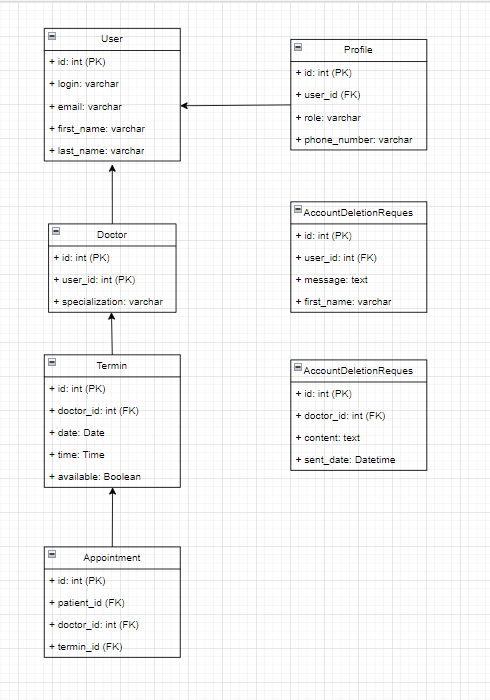
1. **Use case diagram**



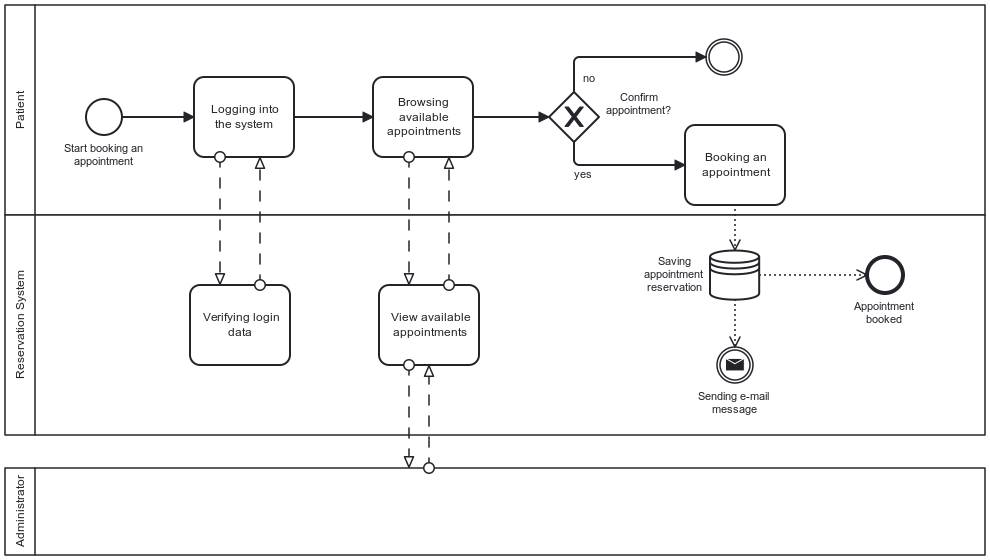
1. **Activity diagram**



1. **Class diagram**

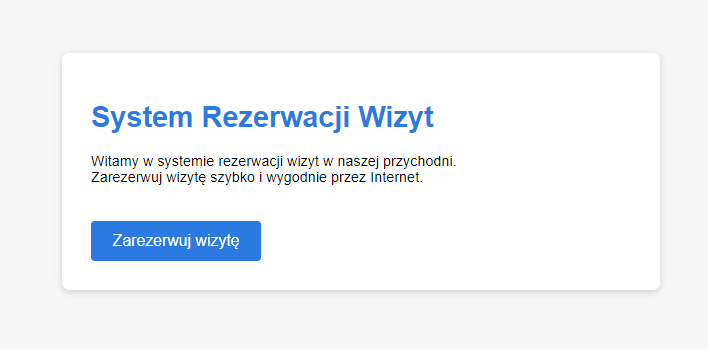


1. **BPMN diagram**

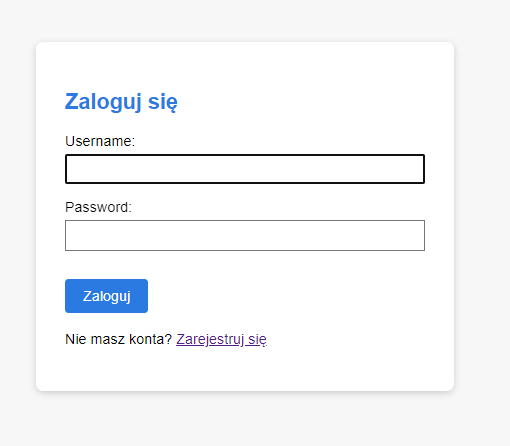


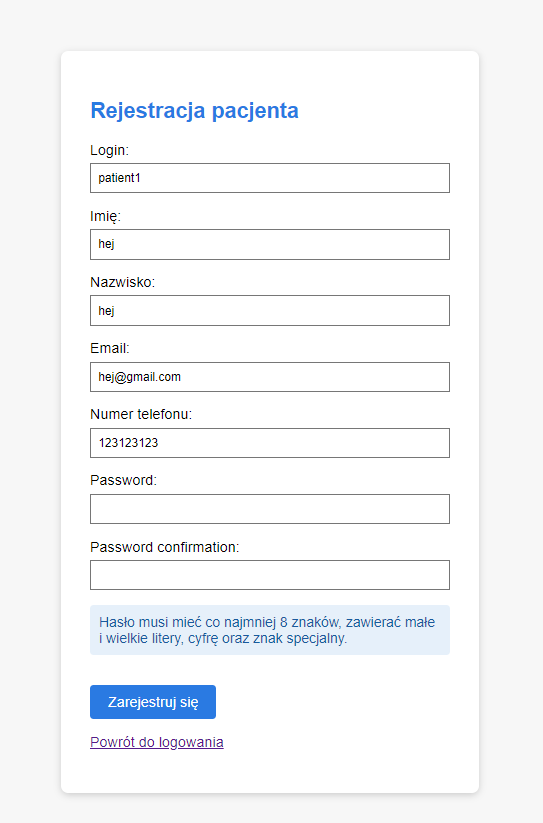
1. **Patient GUI**

Application home page

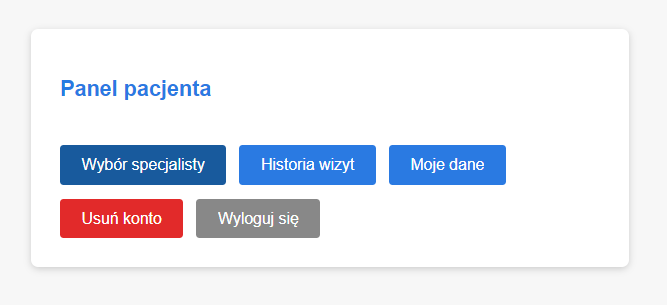


Login page for registered users. The application checks whether the user exists based on the entered login. If there is no account, click "Sign up".





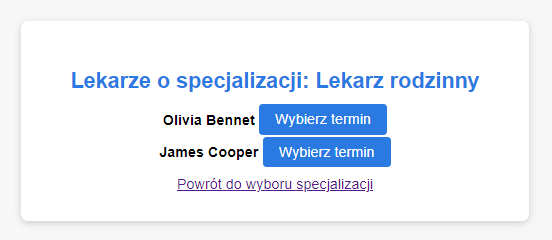
Account registration form with validation for login, email, phone number, and password.



The **patient panel** provides quick access to key features. Users can book a new appointment, check their visit history, view their personal details, delete their account, or log out of the application.

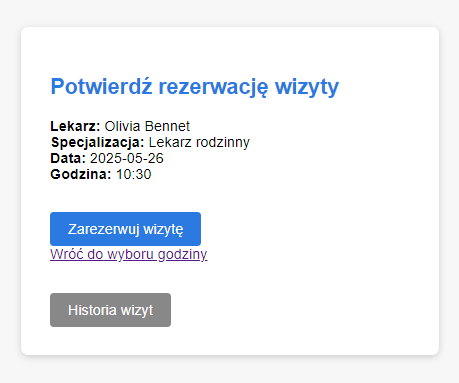


After clicking the appointment option, the user selects a **specialist category** (e.g., cardiologist, dermatologist). The application then displays a list of available specialists within that category. The user can select a specialist and view their **available time slots** to book an appointment.

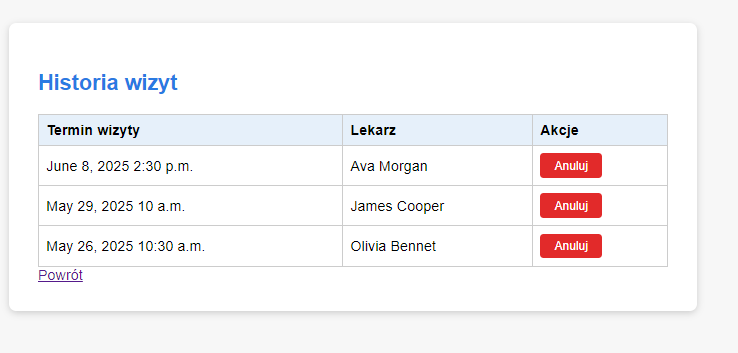
The user chooses the desired **date** and **time** for the appointment from the available options.



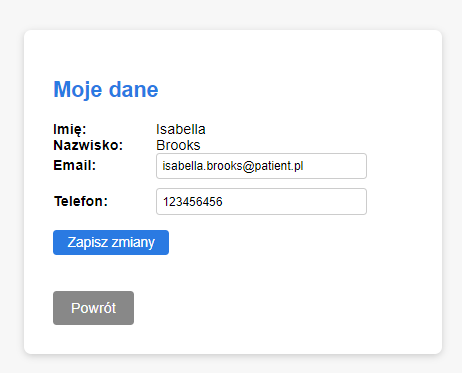
After selecting the day and time, the user confirms the appointment reservation by clicking the **Confirm** button.



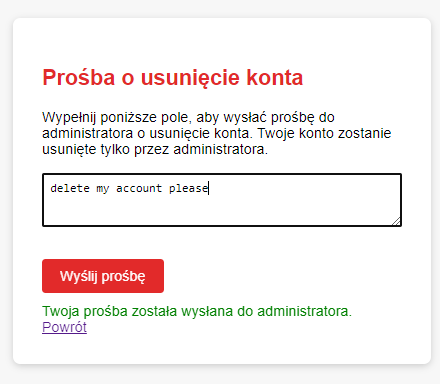
After booking an appointment, we are redirected to the appointment history view, where it is also possible to cancel appointments.



In the "My Data" tab, the user's personal information is displayed.  
 It is possible to update the email address and phone number.

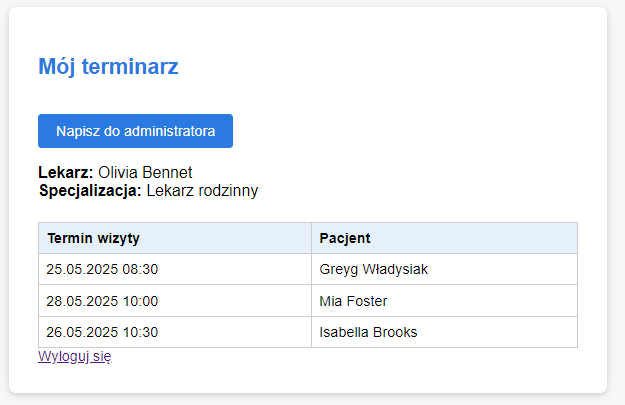


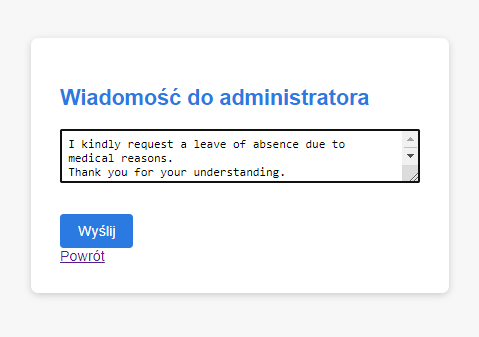
In the "Delete Account" tab, the user can send a message to the administrator to request account deletion for consideration.



1. **Doctor GUI**

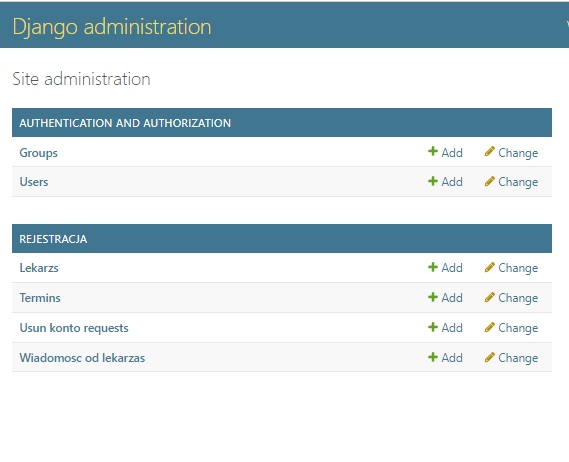
After logging in, the doctor sees their schedule and a button to send messages to the administrator.



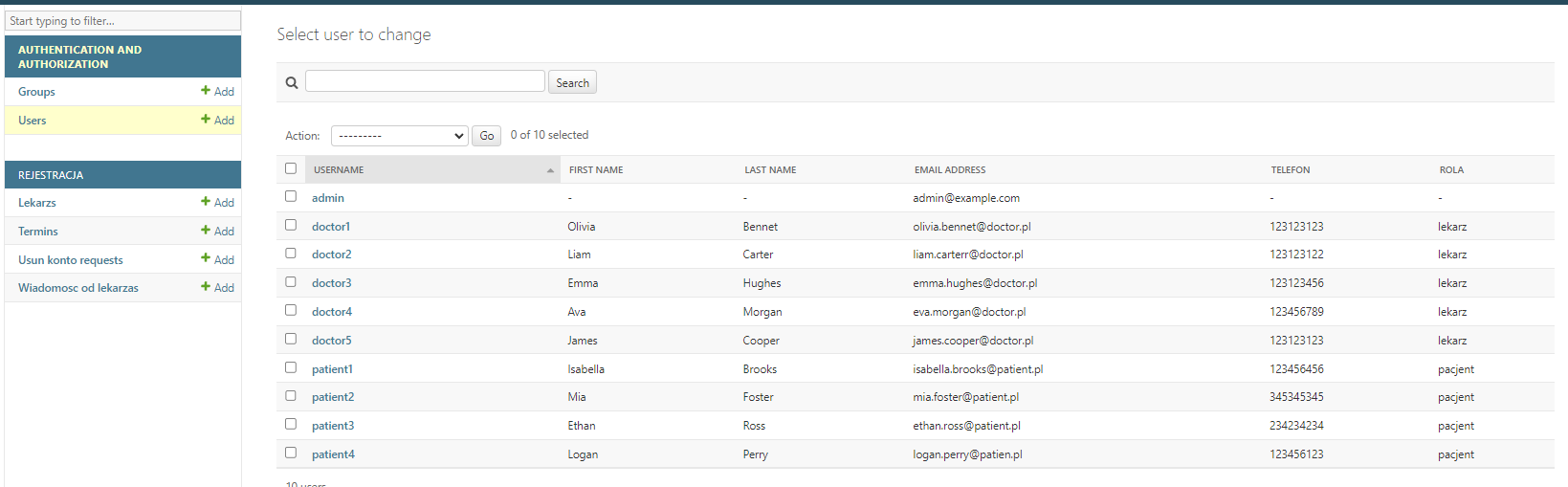


1. **Admin GUI**

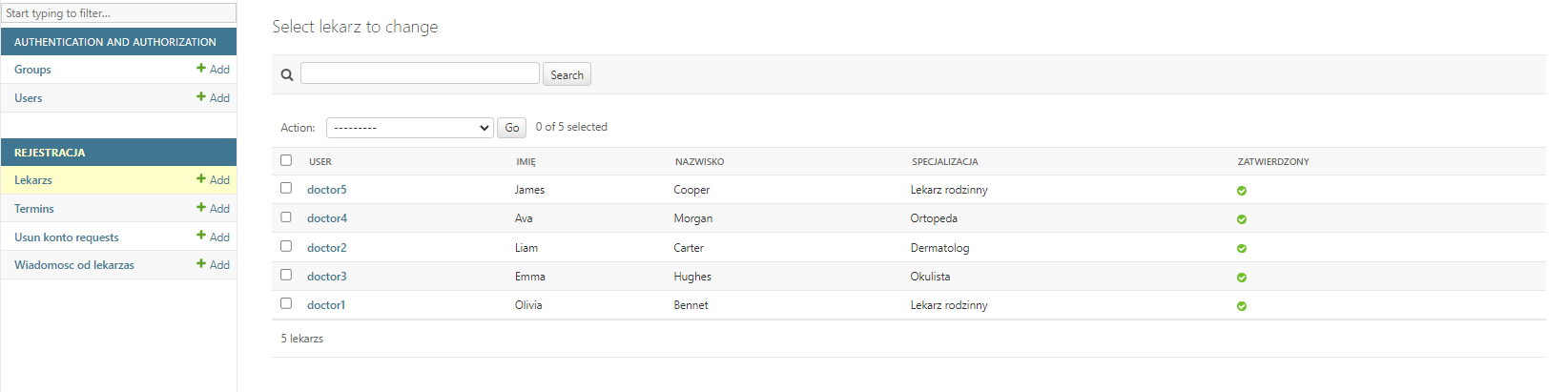
The administrator can view all users in the system, delete users, manage doctors’ schedules, handle patients’ requests to delete accounts, and check messages from doctors.



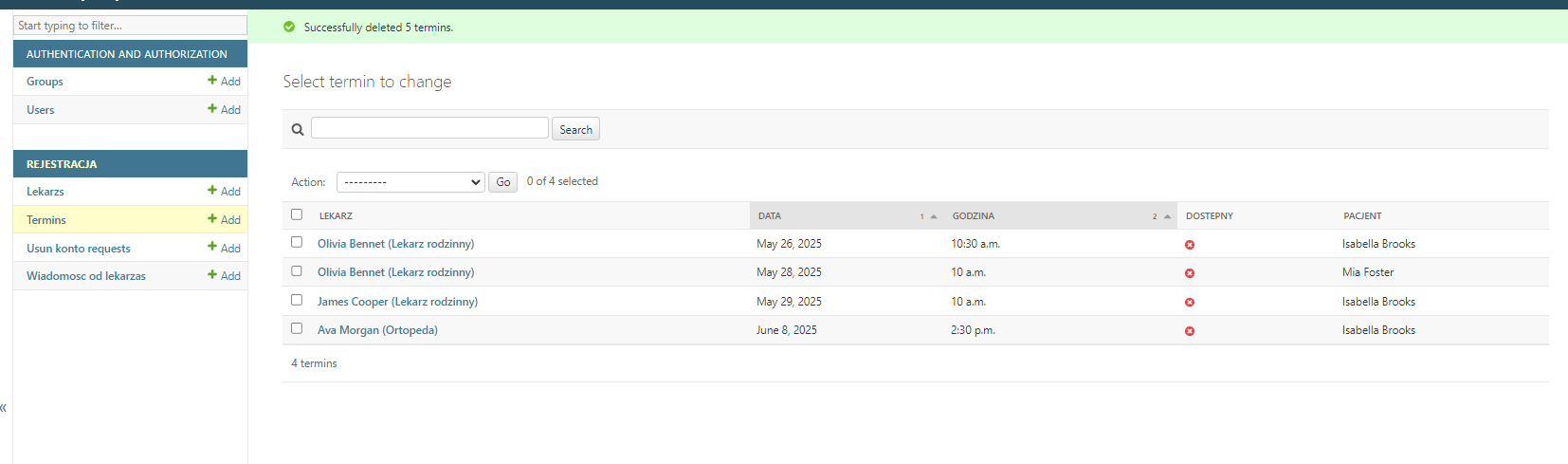
In the **Users** tab, they can view and delete users.



In the **Doctors** tab, they can see only doctors and their specializations.



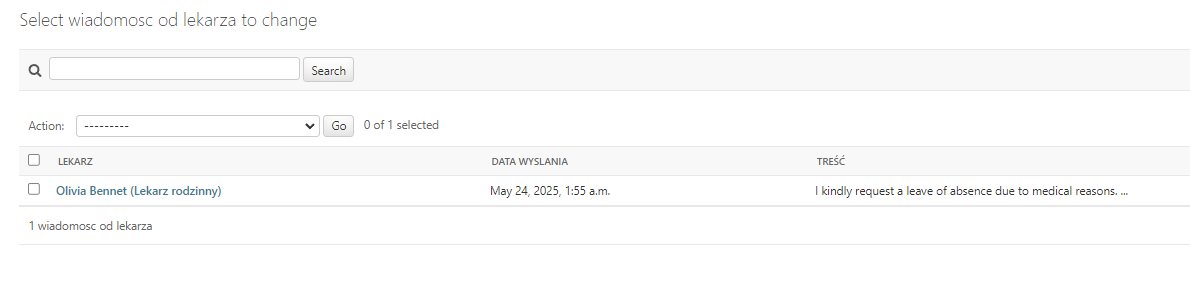
In the **Appointments** tab, the admin can see all appointments.



In the **Account Deletion Requests** tab, the admin sees user requests and can approve or reject them.



In the **Messages from Doctors** tab, the admin can read the content of the messages.



1. **Programming Language:**  
    The system was implemented in Python using the Django framework, which allows for rapid development of web applications following the MVC architecture (Model-View-Controller, referred to as MVT: Model-View-Template in Django).
2. **Architecture:**
3. **Backend:** Django (Python)
4. **Frontend:** HTML templates using Django’s built-in templating system, CSS **Database:** SQLite (default for Django development environment)
5. **Templating System:** Django Templates
6. **Routing System:** Django URL dispatcher
7. **Libraries and Tools:**
8. Django (core of the system)
9. SQLite (lightweight database, default in Django)
10. Python standard libraries
11. HTML/CSS (frontend)
12. Git (version control system)