

1. Stwórz servlet, który zwróci zawartość pliku:
  - 1.1. Wczytaj plik users.txt ([first\_name],[last\_name]) do listy;
  - 1.2. Utwórz klasę UsersServlet i zaimplementuj metodę doGet(), aby wyświetliła użytkowników w postaci listy <ul>;
  - 1.3. Zmapuj servlet na adres /users.
2. Stwórz formularz dodawania nowego użytkownika:
  - 2.1. Stwórz statyczną stronę usersform.htm, w której znajdzie się struktura formularza;
  - 2.2. Utwórz klasę AddUserServlet i w metodzie doGet() dodaj użytkownika do listy, wykorzystywanej przez UsersServlet. Zastosuj wzorzec Singleton, tak, aby oba serlety współdzieliły jedną listę;
  - 2.3. Gdy użytkownika uda się dodać do listy, zwróć odpowiedź ze statusem 201.
  - 2.4. Zmapuj servlet na adres /adduser
  - 2.5. Zmodyfikuj kod metody UsersServlet.doGet(), aby korzystała z tej samej listy, co AddUserServlet.
3. Stwórz filtr uwierzytelniający:
  - 3.1. Stwórz bazę danych servers, a w niej tabele admins (admin\_id, login, password, session\_id) oraz session (session\_id, token);
  - 3.2. Stwórz formularz logowania w pliku login.htm, który wykona przekierowanie na servlet UserLogin;
  - 3.3. W metodzie doPost() pobierz z requesta parametry login i password i sprawdź, czy taki użytkownik występuje w tabeli admins;
  - 3.4. Jeśli tak, to wygeneruj za pomocą klasy Random dwie wartości typu long – session\_id i token. Przypisz je do nagłówków Token i Session\_id. Tymi samymi wartościami zaktualizuj bazę danych.
  - 3.5. Jeśli nie, to zwróć błąd 401;
  - 3.6. W metodzie doFilter() w klasie AuthenticationFilter porównaj wartość nagłówka Token i Session\_id i w przypadku niepoprawnej walidacji zwróć kod błędu 403;
  - 3.7. Dołącz filtr do servletu UsersServlet

## Servlety

### Ciasteczka

1. Utwórz tabele:
 

```
admins {login [varchar], password [varchar], session_id [bigint]}
sessions {session_id [bigint], token [bigint]}
```
2. Utwórz stronę login.htm z formularzem logowania: {login, password, submit\_button}
3. Utwórz servlet UserLogin i nadpisz metodę doPost(), aby pobrała z zapytania wartości login i password i dokonała uwierzytelnienia w bazie danych w tabeli admins (sprawdź, czy dany rekord istnieje)
  - 3.1. Jeśli uwierzytelnienie się powiedzie, to wygeneruj wartości session\_id oraz token i zapisz je do bazy, aktualizując rekord również rekord w tabeli admins
  - 3.2. W przeciwnym wypadku zwróć błąd z kodem UNAUTHORIZED
4. Dodaj do odpowiedzi ciasteczko z wartością token
5. Zmapuj UserLogin pod ścieżką /login
6. Dodaj filtr uwierzytelniający AuthenticationFilter. W metodzie doFilter pobierz ciasteczko token i sprawdź, czy odpowiada aktualnej sesji
  - 6.1. Jeśli tak, to idź dalej
  - 6.2. Jeśli nie, to zwróć błąd z kodem FORBIDDEN
7. Dołącz filtr do servletów AddUserServlet i UsersServlet
8. Dodaj stronę logout.htm z formularzem i przyciskiem

9. Utwórz LogoutServlet, który usunie ciasteczko token (ustawi jego czas życia na 0)
10. Zmapuj UserLogout pod adres /logout

## Przekierowania

1. W metodzie doPost() w klasie LoginServlet, zamiast zwracać kod błędu, wykonaj przekierowanie na stronę 401.htm
2. Stwórz stronę 401.htm i wyświetl na niej komunikat „Niepoprawne uwierzytelnienie”
3. W metodzie doFilter() w klasie AuthenticationFilter, zamiast zwracać kod błędu, wykonaj przekierowanie na stronę 403.htm
4. Stwórz stronę 403.htm i wyświetl na niej komunikat „Brak dostępu”
5. W pliku web.xml zmapuj strony 401.htm oraz 403.htm do odpowiadających im kodów błędów. Metody LoginServlet#doPost oraz AuthenticationFilter#doFilter przywróć do stanu z zadania 4

## Sesja

1. Zmodyfikuj metodę LoginServlet#doPost tak, aby zamiast ciasteczek, po poprawnym uwierzytelnieniu, utworzyła nową sesję z atrybutem user
2. Zamień globalną listę użytkowników na obiekt typu Map<String, List<User>>, który będzie przechowywał osobną listę użytkowników dla każdego z zalogowanych adminów
3. W metodzie UsersServlet#doPost zmodyfikuj kod tak, aby wyświetlana była lista użytkowników z mapy po kluczu user, który zostanie pobrany z atrybutu sesji
4. Servlet UserLogout w metodzie doGet() zamiast usuwać ciasteczko, będzie unieważniał sesję
5. Przetestuj działanie logując się jako admin A – dodaj kilku użytkowników i wyświetl ich listę. Następnie wyloguj się, zaloguj, jako admin B i zrób to samo.

## Zmienne globalne

Stwórz licznik odwiedzin witryny internetowej

1. Utwórz servlet WelcomeServlet, mapowany pod ścieżką /welcome
2. W metodzie init() zainicjalizuj zmienną hitCount
3. Wewnątrz metod doPost() i doGet() inkrementuj wartość hitCount
4. Zwróć w postaci kodu HTML informację o liczbie odwiedzin witryny

## Auto Refresh

1. Stwórz servlet StoperServlet, mapowany pod adresem /stoper
2. W metodzie doGet() wywołaj: response.setIntHeader(„Refresh”, 1) – w ten sposób przeglądarka zostanie poinformowana o konieczności odświeżenia strony po sekundzie

3. Wyświetl w postaci kodu HTML czas, który upłynął od wysłania pierwszego zapytania na adres /stoper