

## Programmazione di Sistemi in Rete (A.A. 2017/18)

Esercitazione n. 14 del 28 novembre 2017

Prof. Eugenio Zimeo

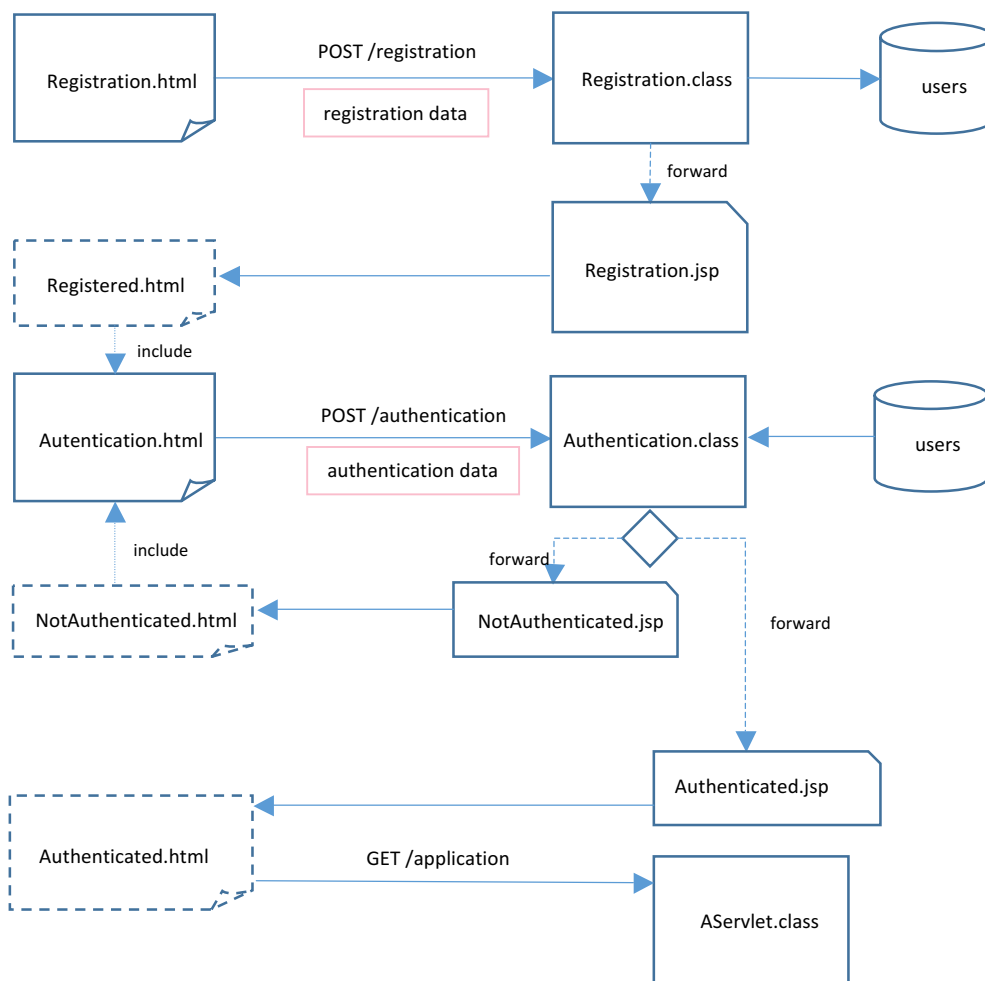
### Esercizio 14.1

Scrivere un'applicazione Web MVC che preveda la registrazione degli utenti e la successiva autenticazione (si può partire dall'esercizio 12.3). La fase di registrazione deve consentire l'immissione di nome, cognome, username, password, indirizzo email ed eventuali altri dati dell'utente. Una volta realizzata la registrazione, l'applicazione visualizza un form per l'autenticazione.

La fase di autenticazione deve consentire l'immissione di username e password. Se le credenziali inserite in fase di autenticazione sono corrette, l'applicazione visualizza un messaggio di benvenuto, contenente anche alcuni dettagli relativi all'utente che si è autenticato (es. Nome, Cognome, E-mail); nel caso le credenziali siano errate o l'utente non è accreditato sul sistema, viene visualizzato un messaggio di errore.

Impiegare il modello MVC (una servlet per la gestione dell'input e una JSP per la generazione dell'output) sia per la registrazione sia per l'autenticazione. Si realizzi una versione di JSP (Authenticated.jsp) con azioni (useBean, getProperty) e bean, ed una con espressioni (EL) e bean.

Per la gestione di un semplice database di utenti, si usi una collezione memorizzata nel contesto dell'applicazione, utilizzando una classe Java (UserMgr).



Si gestisca lo stato mediante Java Bean di tipo `User` con le seguenti proprietà: `firstName`, `lastName`, `sex`, `email`, `username`, `password`. Per caricare il modello (funzioni e stato) nella memoria con scope

application prima che qualsiasi servlet o JSP venga invocata, si può ricorrere all'uso di un listener di contesto i cui metodi del ciclo di vita sono invocati quando l'applicazione è avviata e quando viene spenta. A valle dell'autenticazione, conservare le informazioni che caratterizzano l'utente autenticato nella memoria di sessione.

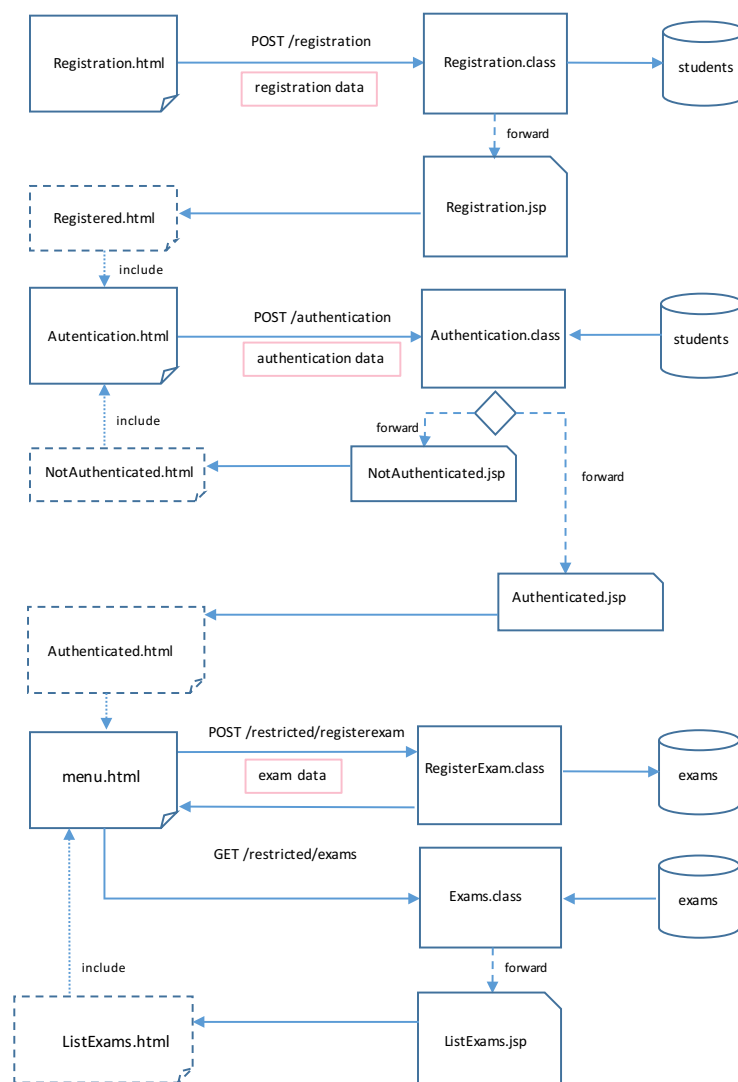
### Esercizio 14.2

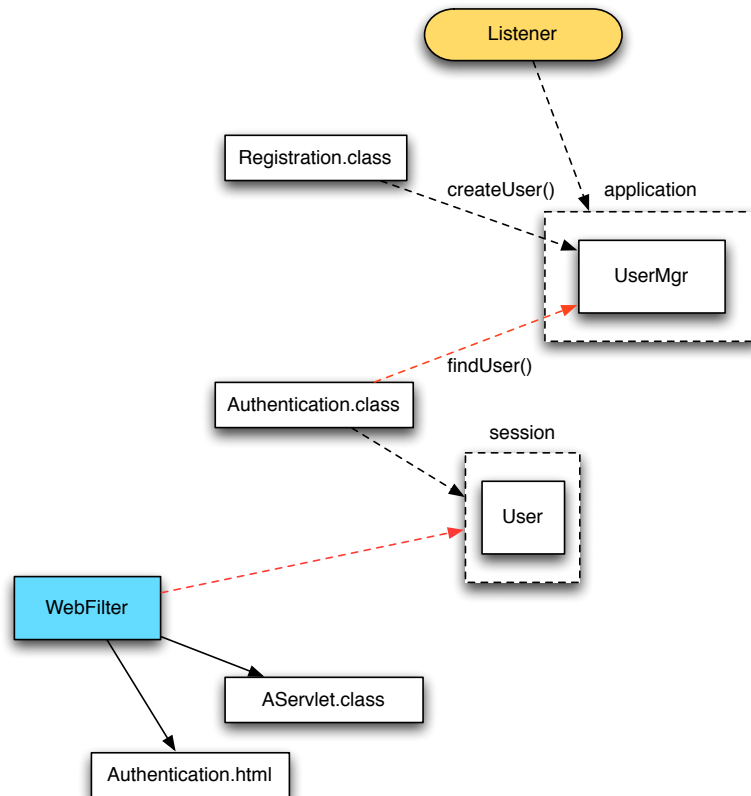
Modificare l'esercizio precedente in modo da evitare di scrivere il codice di verifica dell'avvenuta autenticazione direttamente nei controller. Si può ricorrere all'uso di un filtro (WebFilter) in grado di intercettare le richieste HTTP rivolte ai controller e di verificare che la richiesta sia legittima. In caso positivo, la richiesta è inoltrata al controller (es. AServlet); in caso negativo è re-diretta ad una JSP che visualizza un messaggio e chiede l'autenticazione.

### Esercizio 14.3

Scrivere un'applicazione Web per gestire gli esami sostenuti dagli studenti di un Corso di Laurea. L'applicazione deve consentire la registrazione e la successiva autenticazione come da esercizio 14.1. Una volta realizzata l'autenticazione con successo, l'utente è autorizzato ad accedere a risorse protette. Come esempio di risorsa protetta, si realizzi una funzionalità che consenta l'accesso all'elenco degli esami sostenuti dall'utente in sessione.

Si adotti il pattern MVC per l'implementazione dell'applicazione e si scrivano le seguenti componenti:





Si utilizzi per le risorse protette il seguente path nell'URL: /restricted/\*  
Si memorizzino i dati relativi agli esami sostenuti con scope application.  
Si utilizzi il filtro per l'accesso a risorse protette.