Diario di lavoro

|  |  |
| --- | --- |
| Luogo | Trevano |
| Data | 23.01.2019 |

|  |
| --- |
| Lavori svolti |
| Oggi è stata una giornata meno proficua del solito. Essendo infatti il primo giorno lavorativo composto da solamente 4 ore, mi son dovuto gestire diversamente con il diario. Inoltre, il problema che avevo notato la scorsa volta si è rivelato più impegnativo del previsto. Ero convinto di poter trovare una soluzione alternativa e cambiare il sistema di Spring Security, ma ne è risultato che tutti usino un sistema uguale se non molto simile, che comunque non prevede l’utilizzo di più campi.  Non trovando i risultati sperati, ho quindi pensato a modellare la domanda, in modo da ottenere risultati simili a quelli sperati senza dover utilizzare più parametri. Ho infatti cominciato a ricercare il modo di utilizzare, all’interno di Spring Security, un filtro.  Dopo aver esplorato un po’ il web tuttavia mi son reso conto che questa strada non mi avrebbe permesso comunque di svolgere il lavoro come desiderato. Il link primario controllato è:  <https://www.baeldung.com/spring-security-custom-filter> .  Nel frattempo, ho provveduto a creare il Dao dell’utente, ovvero il mezzo tramite il quale andrò in futuro a interrogare il database tramite metodi specifici e chiusi (potendo così gestire appieno SQL Injections, errori di valori nulli e al contempo non dare la possibilità a nessuno di eseguire query non necessarie).  La pagina che ho cercato di replicare, senza tuttavia arrivare ad una soluzione, è questo: <https://www.concretepage.com/spring/spring-security/spring-mvc-security-jdbc-authentication-example-with-custom-userdetailsservice-and-database-tables-using-java-configuration>  Alla fine, per evitare una perdita di efficienza lavorativa, ho deciso di ripiegare a ciò che in fondo avrei potuto pensare di implementare fin dall’inizio: un campo unico denominato id all’interno dell’utente, che farà da ridondanza unendo i campi nome, cognome e data di nascita, separandoli tramite l’utilizzo di un “.”.  Ho quindi effettuato il cambiamento a livello di database, per poi tuttavia essere costretto a rimandare il completamento e verifica di funzionamento alla prossima giornata lavorativa. |

|  |
| --- |
| Problemi riscontrati e soluzioni adottate |
| Il problema principale è stato proprio la mancanza di risultati riguardo all’utilizzo di Spring Security specifico e adatto alla mia situazione. Alla fine ho risolto proprio rinunciando alla possibilità e creando una ridondanza all’interno del database, che però mi permetterà di alleggerire alcune operazioni e la tabella sottostante “subscription”, quindi direi che in fondo anche rinunciando a qualcosa non c’è stata una vera e propria perdita di qualità ed efficienza del prodotto. |

|  |
| --- |
| Punto della situazione rispetto alla pianificazione |
| La scorsa volta non avevo pensato al fatto che questa sarebbe stata la mia prima giornata di solamente 4 ore lavorative. Pertanto, l’implementazione del login non è ancora stata completata, sebbene abbia già creato tutta la logica quantomeno del login tralasciando la sua gestione tramite Spring Security, e abbia già predisposto i mezzi necessari al funzionamento effettivo della registrazione. |

|  |
| --- |
| Programma di massima per la prossima giornata di lavoro |
| Terminare la logica di implementazione della prima parte, modificare lo schema E-R una volta completato il tutto in modo da avere una versione da poter definire quasi definitiva di quest’ultimo e strutturare l’invio effettivo delle email. |