Diario di lavoro

|  |  |
| --- | --- |
| Luogo | Trevano |
| Data | 05.02.2019 |

|  |
| --- |
| Lavori svolti |
| Oggi finalmente mi son potuto dedicare al verificare il funzionamento della registrazione. Questo ha richiesto più tempo del previsto in quanto ho provato a strutturare il tutto tramite metodi più immediati e un uso più completo delle prepared statement.  Ho inoltre rivisto ancora una volta lo schema E-R, aggiungendo un ulteriore campo all’utente: il link. Questo permette di utilizzare un link, che altro non è che una stringa casuale di 10 caratteri assegnata al momento della registrazione dopo aver confermato la sua unicità all’interno del database, per abilitare l’acccount. In questo modo, l’utente non si ritrova la sua chiave primaria all’interno di qualcosa di vulnerabile come un url in metodo get. Questo link, essendo comunque in grado di selezionare l’utente, viene cancellato al momento della sua conferma. In questo modo, lo spazio occupato, nonché il numero di possibilità presenti (le quali comportano una maggiore difficoltà, con grandi numeri, a trovare una stringa univoca), diminuisce.  La struttura finale è la seguente:  Metodo di gestione registrazione:  Metodo di creazione utente:    Metodo che gestisce la richiesta sql tramite PreparedStatement:    Questo metodo ancora mi ritorna false, nonostante l’operazione va a buon fine. Devo ancora trovarne il motivo, ma per ora lo trovo un problema secondario.  Metodi di abilitazione account (il primo nel controller, il secondo nel Dao):      Ho poi ristrutturato allo stesso modo anche il Dao dal punto di vista delle richieste di select, con le opportune modifiche, quali:      A questo punto ho potuto finalmente gestire i possibili errori che possono essere riscontrati.  Per il login, gli unici errori possibili sono un account non registrato oppure uno non abilitato. Non avendo veramente un modo comodo di differenziare i due stadi, ho deciso di far notare semplicemente la possibilità che la registrazione sia presente ma che l’account non sia abilitato. In questo modo, non è possibile effettuare dei controlli non permessi dello stato di registrazione di altri utenti conoscendone i dati personali. Per lo stesso motivo, non viene fatta la differenza tra errori per id e password.  Per quanto riguarda la registrazione invece, c’è solamente la possibilità che l’utente sia già registrato da gestire.  Sono poi da gestire la validità di quanto viene inserito nei campi input, di entrambe le fasi:  nome e cognome sono poi controllati tramite JS con regular expression, per permettere l’inserimento di caratteri che non vengano inseriti numeri e caratteri speciali (vengono però tenuti da conto lettere con accenti, come cediglie o simili). La regular expression in questione è stata presa da il seguente sito: <https://stackoverflow.com/questions/150033/regular-expression-to-match-non-english-characters> (soluzione nei commenti della risposta).  Il campo email invece usa una regular expression tutta sua, che è stata per semplicità presa da un sito (https://emailregex.com/).    Per finire ho effettivamente implementato la parte riguardante alle email, sebbene non abbia ancora potuto effettivamente testarla in quanto la porta smtp viene bloccata dal proxy. Questo problema, assieme alla mancanza di un effettivo spazio per il database esterno, nascono principalmente dal fatto che avrei dovuto richiedere i due elementi molto prima, ma che alla fine rimandavo. Ho provveduto solo oggi, nel mentre vedrò di verificare le email fuori dall’orario lavorativo e il database, così come di conseguenza il sito, solo in locale (cosa effettivamente fatta, sebbene non penso serva a molto creare un test-case dedicato, tenendomi quest’ultimo per quando potrò testare tutto online). |

|  |
| --- |
| Problemi riscontrati e soluzioni adottate |
| Nessun problema in particolare. |

|  |
| --- |
| Punto della situazione rispetto alla pianificazione |
| Reduce delle 8 ore perse, ho ancora un po’ da recuperare, ma se il codice per l’email funziona come deve direi che sono praticamente in pari. |

|  |
| --- |
| Programma di massima per la prossima giornata di lavoro |
| Testare email, creare home, implementare multilingua. |