Week 1

Summary of the Application

The selected application is fueled by an initiative to help a particular diaspora community enter marriage-hood with peer cultural members. This is a matchmaking site that will increase the social network to find soul mates for the sole purpose of leading them to marriage.

The main functionalities of the application are allowing a member to create a profile as a new member, search other members’ profile for potential match, send requests of interests, view responses, view requests(incoming) and either accepts them or reject them(or simply ignore them).

All these transactions are overviewed by an administrator who can override them for the sake of keeping the site civil and functional. Hence, the admin can warn users who violate the site policies and even go as far as suspending them.

Use case diagram



List of prioritized use cases

1. Create Profile
2. Search for matches
3. Send Requests
4. View Responses
5. Accept Requests
6. Reject Requests
7. Ignore Requests
8. Monitor users
9. Warn members
10. Suspend members

**Create Profile – Brief Format**

The member clicks on the link to create a profile and is prompted to enter their credentials to login. Once authenticated, a form is displayed to enter names, location, brief description, interests and preferences for a mate. Finally, there is an option to upload their picture. The user can hit the submit button to post the profile.

The underlined phrase represents a use case “includes” or “uses”.

**Search for match– Casual Format**

1. Main scenario success

* The member clicks on a link to search for a potential mate, they are then requested to enter the credentials to login. Once successfully authenticated, they can get a search box where they enter the search criteria based on the targeted profile. After submitting the search criteria, the search results are displayed back to the user in a form of a list of profiles. The user can drill down each profile for the details.

1. Alternate scenarios(extensions)

* The credentials are invalid, the user is requested to enter new ones and also presented a link for forgotten username/password and a link to register as a new user.
* The user is suspended, the user is asked to contact the administrator through the “contact us” information.
* The search criteria do no yield any results, the user is presented back the search box to re-enter new criteria.

**Send Request – Fully Dressed**

Use Case: Send Request

* Primary actor: site member
* Stakeholders and interests:

-Member: wants to connect, to hook up to another member to express interest.

-Site: wants to provide a civil, efficient, safe and secure platform for mates to meet.

* Preconditions

-The site is up and running, with the menu options include the search one that leads to send request.

* Success Guarantee

-The request is sent and available to be viewed by the targeted member.

* Main Success Scenario:
  + 1. The member click on the link to search for profiles
    2. They are then requested to login
    3. The member enters their credentials
    4. The member is shown a box to search for profiles
    5. The member enters the search criteria
    6. The search results are presented back to the member with a link to send a request to each
    7. The user clicks on the send request link for the chosen member
    8. The user enters some expressing words to show interests and submits.
    9. The user is redirected back to the main menu.
* Extensions:

1. Invalid credentials:

- The member is requested to enter new ones

-The member also presented a link for forgotten username/password and a link to register as a new user.

1. The user is suspended, the user is asked to contact the administrator through the “contact us” information.
2. Search results empty:

- The member is presented back the search box to re-enter new criteria.

* Special requirements:

-The site must respond within 5 seconds

-The site must be available for all major browsers

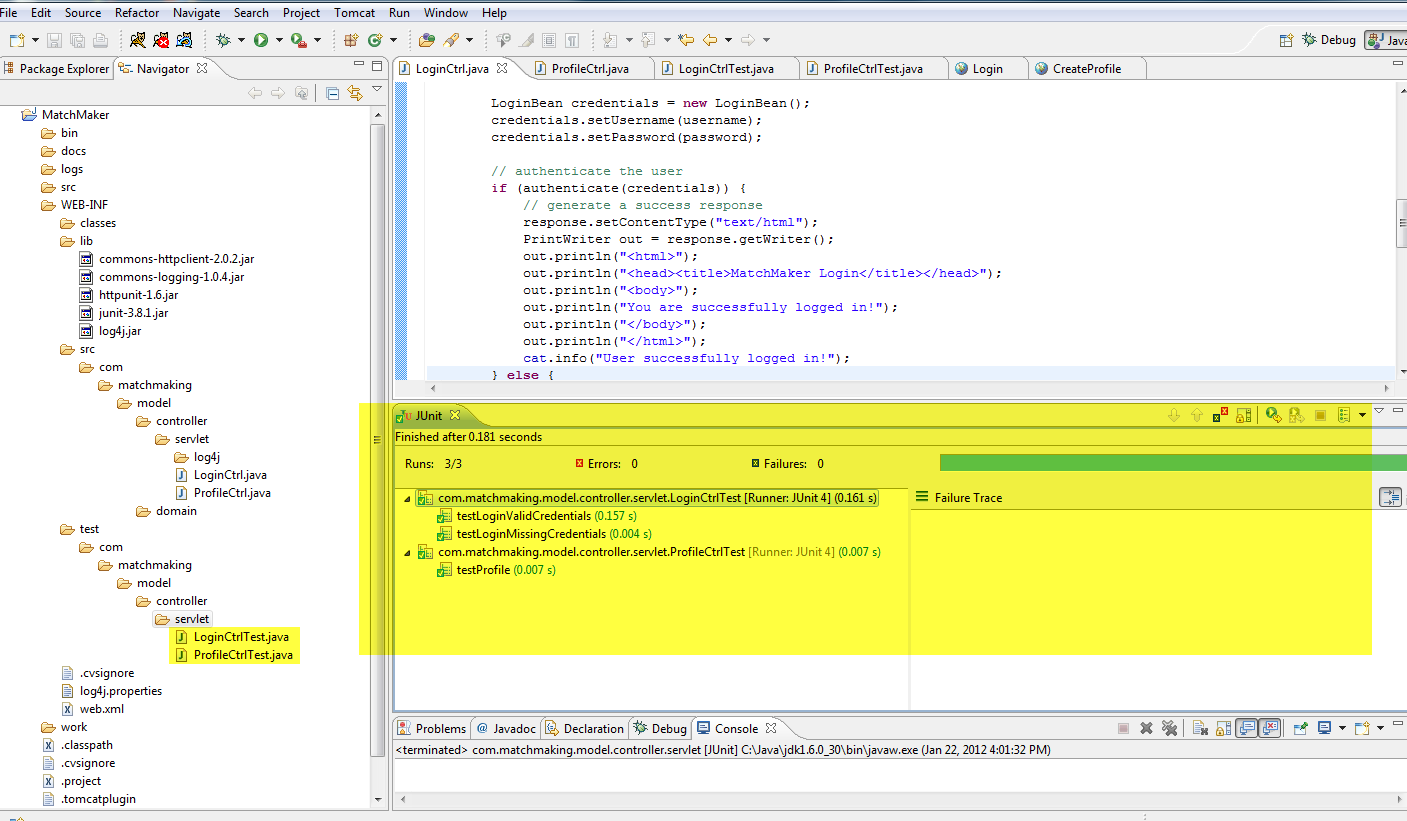
* Technology and data variation list: N/A
* Frequency of Occurrence: approximately 1/second once the site is mature.
* Open Issue: N/A

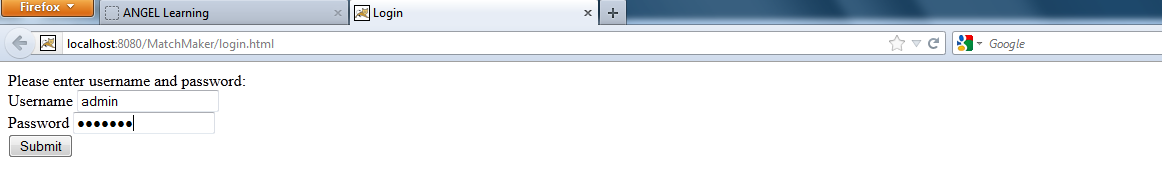
Week 2

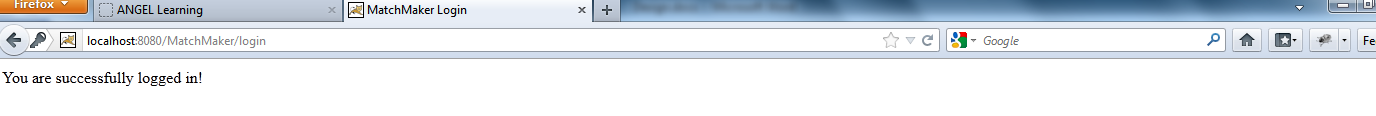
I have implemented login.html, createProfile.html for the presentation layer(view part), LoginCtrl, ProfileCtrl as servlet classes to process the request and response. These classes are part of the controller part of the presentation layer. The latter classes tap into LoginBean and ProfileBean of the domain layer to save and retrieve data as POJOs.

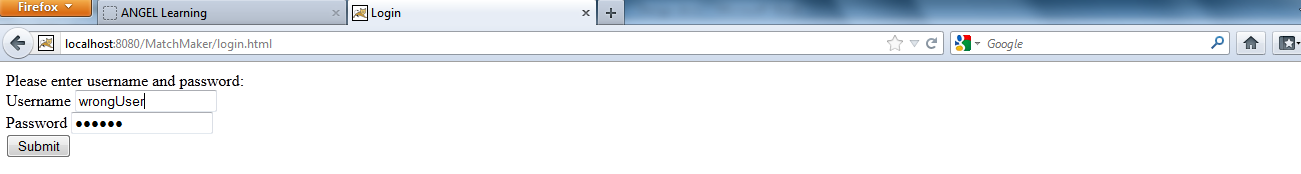
Testing is achieved by two classes, LoginCtrlTest to pass valid, invalid credentials, ProfileCtrlTest to validate the data submitted to the server.

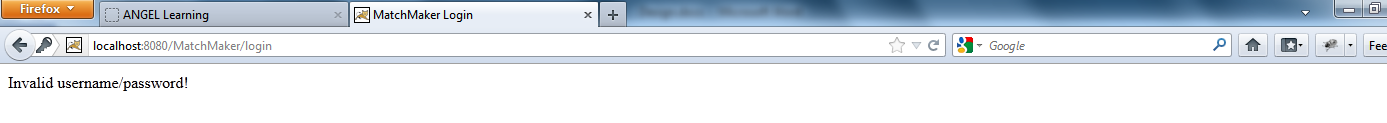
Log4J has been added to keep logging the data to the console and file for debugging purpose.

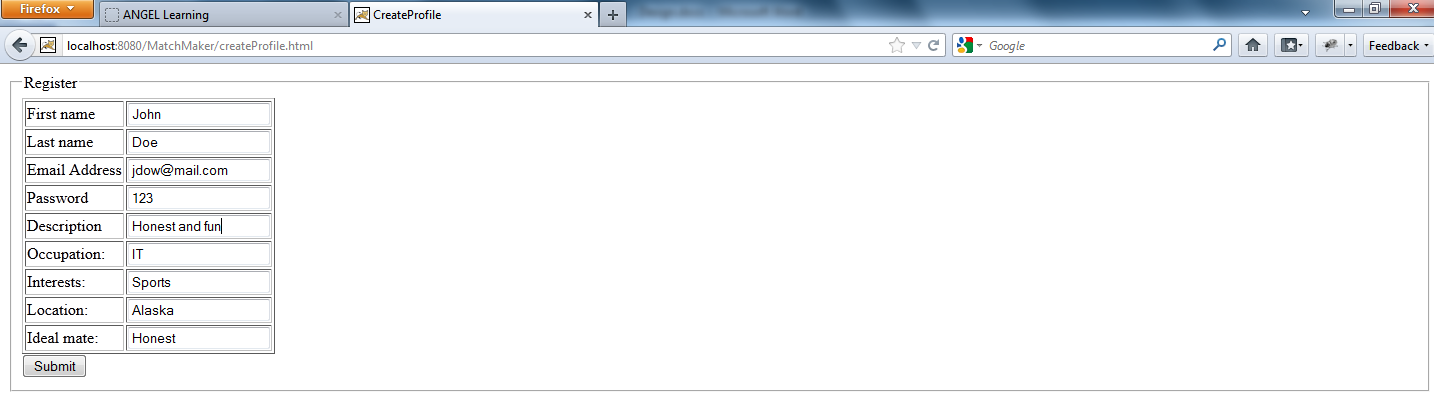


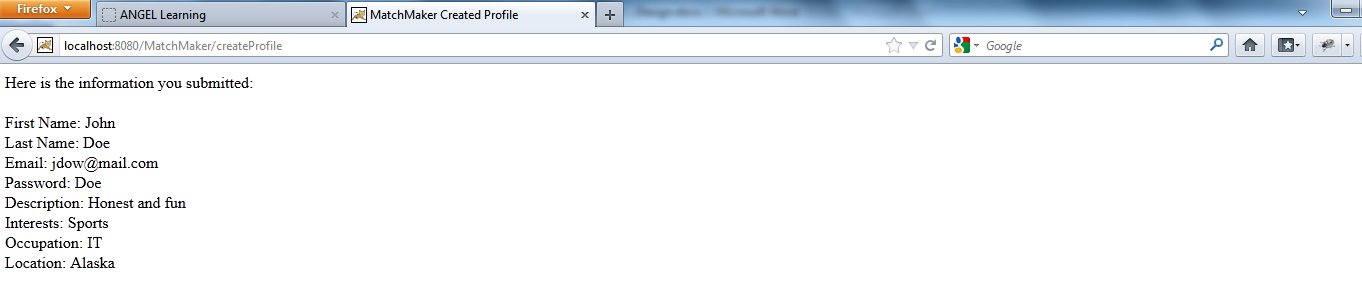


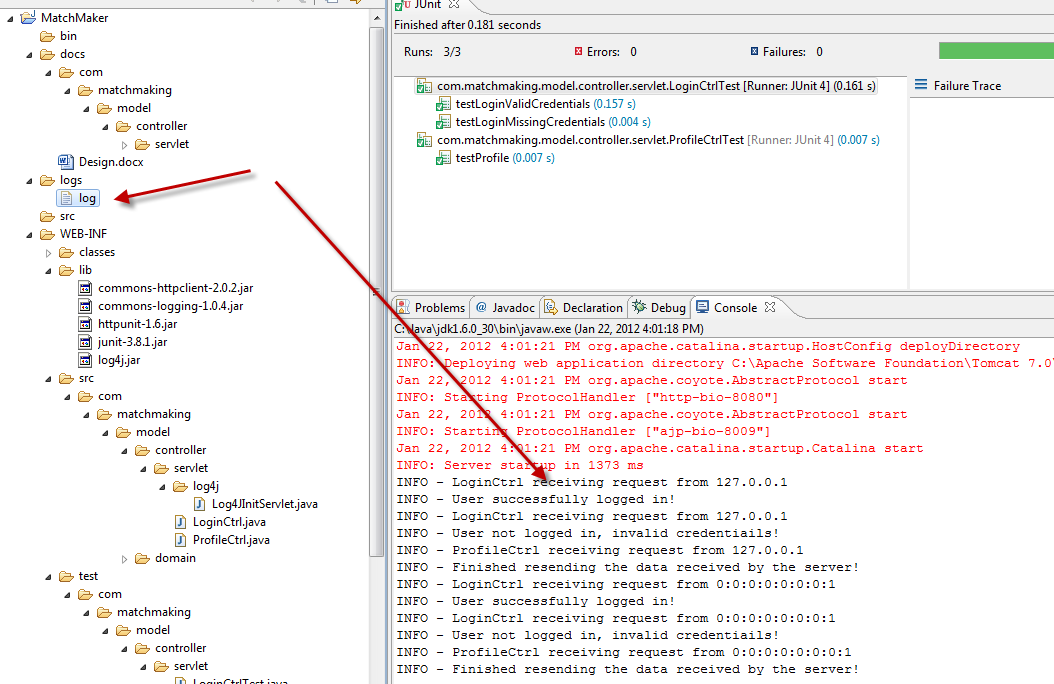












WEEK 3

* Added managers’s classes to process the business rule of authentication(LoginManager) and echoing the information received by the server(ProfileManager).
* Added “view” servlet classes to process the information to display when the user logs in successfully(HomeServlet), fails to log in(ErrorServlet) or to echo the data received at the server(EchoServlet). During the interaction with the controller servlets and these new “view” servlets, the pojos are passed in the session scope.
* Added test classes for LoginManager and ProfileManager
* Added toString() method to the pojos as suggested.

Week 4

* Created 3 jsp pages to replace the “view” servlet: home.jsp, error.jsp, echo.jsp
* Implemented a service layer to save and retrieve data from the database and pass them to the business layer. AuthenticationSvcJDBSImp.java and ProfileSvcJDBCImp.java are the main classes
* Added tests for all the layers. Ran into some serious debugging issues with JUnit not loading classes in the context, hence why some 2 tests are still failing. I will continue working on that and make sure they are fine with assignment 5.