Project 1a

Name: Jiabin Dong NetId: jd836

1. To run the tryFirst.war, place it under tomcat/webapps, start tomcat and type http://localhost:8080/tryFirst/hello. You can also run it through Eclipse: import the tryFirst folder into Eclipse, start tomcat, type http://localhost:8080/tryFirst and click hello link.

2. There are three code files:
    a. HelloUserSession.java, which is the main servlet processing file.
    b. styles.css, used for setting web page style including word font, background or so.
    c. index.html, which creates an index file for user to access the session page. By simply clicking the hello link, clients are able to get a session behavior page.

3. In HelloUserSession.java, there are a SessionData structure, 4 helper functions, doGet and doPost functions for processing session behavior and cookie management.
    a. SessionData class defines the format of session data, containing <sessionID, version, message, expiration-timestamp>. Session data are stored in a HashMap, whose key is sessionID and value is SessionData.
    b. updateId helper function is used for generating global unique sessionID. To make sure the Id is globally unique, I use UUID to generate the Id number and store it and a string.
    c. removeOutOfDate helper function can remove timed-out sessions from the session data table.
    d. Given the name of cookie, getCookie helper function can get the relevant cookie from request.
    e. In doGet and doPost, after receive a request, I remove the timed-out sessions from session data table. Then I check whether the cookie with name “CS5300PROJ1SESSION” exists. If not, I create new sessionId using updateId(), and set version number to 1, default session message and expiration time. After that I put the new session data structure into session table, use synchronized method to avoid multiple accesses to the table simultaneously. I also create a new cookie with name = “CS5300PROJ1SESSION”, max age = 5 seconds and put them into response. If the cookie with name “CS5300PROJ1SESSION” exists, I firstly update there session data(increase the version number) from session table and modify the cookie value. Then check which type of request it is. “replace” and “refresh” requests are similar in terms of update expiration time, I implement this by add the 5 seconds to the current time, as well as max age of cookie. In “replace”, I also put the message requested to display into session table. However, if “logout” is requested, I set the expiration time of session data to zero as well as max age of cookie and remove this session data from session table, also synchronized access is considered here. Then a “logout successfully” page is created to show the logout state.
    f. The last function is for display the session behavior page using html language.