1 Introduction

The on-line version of the Dermofit tool is based on the Java JSP technology. Java servlets, running on a Tomcat server, generate the content, which is displayed in a user's web browser. A user can interact with the content thanks to JavaScript, and results of these actions are sent back to the server for processing.

The Java servlets reuse the code from the main, standalone application. Indeed, the JSP code can be thought of as a visualization layer.

There are two versions of the on-line Dermofit application. The first one, DermofitOnline, enables conducting experiments which were generated using the configuration tool. The test and reference lesions, as well as the flow of experiment, are read from a configuration file. The DermofitSmart does not use any of this information, but instead takes advantage of the programs written by Matteo and Lucia. The reference lesions are chosen based on features of the test lesions, and diagnosis history for a given test lesion.

2 DermofitOnline

This on-line application consists of only four JSP pages: dermofit.jsp, detailsPage.jsp, questionPage.jsp, feedbackPage.jsp. A user is redirected between the pages in the order above, when each stage of an experiment is completed.

2.1 dermofit.jsp

This servlet displays a welcome message, an instructional video, and sets up the essential objects and a session.
The name of the experimental configuration is passed to this servlet as an URL parameter. The file path for the relevant configuration file is found, and an instance of the ExperimentalSetup is read using the code as in the standalone application. An instance of the ExpLogger is created, so that events can be logged throughout an experiment, and saved to a file at the end. Finally, an instance of ExpController is created, which governs the flow of an experiment. The three objects are saved in a session.

The instructional video had been posted on YouTube, and here only the link was inserted.

On clicking a button, the browser is redirected to the details page.

2.2 detailsPage.jsp

The servlet display a form for statistical data, parses the input, checks its competeness, and re-directs the user to the question page.

The form is bound with an instance of the ParticipantDetails class. When a user submits their data, the page is reloaded. If it is incomplete, the prompts for filling in the missing information are displayed. Otherwise the complete stats information is passed to the ExpLogger, stored as a session variable.

2.3 questionPage.jsp

The servlet displays the test lesion, alongside diagnosis options, and enables a user to select the best one. The servlet handles user’s actions (a diagnosis, request for more options, backtrack) and reacts to them. When the experiment is finished, a user is redirected to the feedback page.

The servlet responds to the following user’s action, which are communicated a URL variable:

- a user chooses a reference lesion matching the test lesion best
- a user decides to backtrack, regarding the available reference lesions as inappropriate
- a user asks for more reference lesions
- no user action, which means a new question needs to be displayed, or the experiment finished when all the questions are finished

In response to the above events, the servlet calls methods of the ExpController instance. The calls return instances of the DisplayCommand class, which are interpreted by the servlet and relevant HTML is displayed.
2.3.1 Example of interactions within the system when a diagnosis is made

1. A user requests a new question to be displayed, by making an HTTP request to `questionPage.jsp`.

2. The JSP servlet generates HTML content, which is displayed in a user’s browser. The question screen consists of a test lesion, and few available reference lesions, which become diagnosis options.

3. A user selects a matching lesion. A JavaScript (`index.js`), within the user’s browser, highlights the selected lesion.

4. A user selects a matching lesion, and presses the `Match` button. The JavaScript, on the user’s side, processes the request. It passes the index of the selected lesion, together with the action type (here match) as parameters in the new HTTP call to `questionPage.jsp`, on the server side.

5. `questionPage.jsp` checks the argument values passed in the URL, especially checks what action had been taken.

6. In response to the user’s action, `questionPage.jsp` calls a method of `ExpController.java`. In this case, it will be the method which handles a match event.

7. Action handling by `ExpController.java` generates an instance of a `QuestionCommand.java`.

8. The display directives in `QuestionCommand.java` are then executed by `questionPage.jsp`. The HTML content which displays the new screen is generated.

The JavaScript script (found in scripts/index.js) on the top of the servlet enables a user to select a lesion by highlighting an image. The result of the selection is passed to the servlet in the URL call.

2.4 `feedbackPage.jsp`

The servlet displays a goodbye message to the user, quoting the accuracy of diagnosis in the finished experiment.

3 DermofitSmart

This JSP application has exact same structure as DermofitOnline, with the main servlets being `dermofit2.jsp`, `detailsPage2.jsp`, `questionPage2.jsp`, `feedbackPage.jsp2`. The code is largely unchanged, and the above description still applies.
The biggest difference is using different experiment controllers. ExpControllerMatteo is used where ExpController would be used in DermofitOnline. ExpControllerMatteo, apart from controlling the experiment flow, accesses Matteo’s and Lucia’s code, which automatically suggests appropriate reference lesions.

The LesionSelector class, specifically its method retrieveDispLesions, serves as an interface with the controller. As arguments this method takes the test lesion ID, the IDs of intermediate diagnosis lesions, and required number of reference lesions. The method returns the reference lesions, based on the features extracted from the lesion images.

4 Accessing the demos

4.1 Dermofit Smart

A demo experiment:
http://demos.inf.ed.ac.uk:8848/DermofitSmart/dermofit2.jsp

The results can be accessed at:
http://demos.inf.ed.ac.uk:8848/DermofitSmart/results/smartDermofit.htm