A Tool for creating online Anatomical Atlases

Summer Scholarship Report
Faculty of Science
University of Auckland
Summer 2003/2004

Daniel Rolf Wichmann
UPI: dwic008
UID: 9790045

Supervisor: Burkhard Wuensche

Abstract:

Many researchers in medical and biological sciences produce 3D data sets of anatomical structures but are unable to communicate and explore their data sets interactively over the internet because they lack the necessary technological skills. As a result opportunities for feedback, collaboration, and promotion and development of research are missed. In this report we present a tool for creating online anatomical atlases. The tool will allow users to publish their own biomedical data sets on the World Wide Web and enables other users to interactively explore the data sets over the internet. The users are able to annotate their data sets by selecting and labeling regions of interest. Possible applications are the publication of medical data sets of healthy and diseased human organs (brain, heart, lung, etc.), CT and MRI data sets of animals (e.g. the frog's anatomy) and 3D microscopy data sets of anatomical structures (e.g. slice images of an animal's eyeball).

1. Introduction:

Most of the online anatomical atlases already in existence are quite inflexible and static in nature. They generally focus on one particular data set, which limits the application domain. *"The Whole Brain Atlas Navigator"* [1], developed at the Harvard Medical School, is an example of an anatomical atlas which is restricted in its domain, in this case the human brain. In addition to this, it is also quite restricted in the functionality it offers; in particular it only allows viewing of the data set along one axis. It is not very suitable as a teaching tool as it doesn't accommodate for labels, highlighting and describing regions of interest to a novice user.

The application presented in this report will allow users to create anatomical atlases using biomedical data sets. These data sets can be obtained from a variety of sources including computed tomography (CT) and magnetic resonance imaging (MRI). This allows for a lot flexibility and a wide application domain.

The application is written in Java [2] using a client server architecture. For a good introduction on the client server architecture, see [3]. The application is web based and runs as an applet in a web browser obtaining the data needed from the server(s). The application can also be downloaded and run as a stand-alone application (provided the Java Virtual Machine is installed).

The application relies on 3 servers (see Figure 1). The login server deals with the user accounts and logins. In other words, when a user tries to login, the request is delegated to the server which checks the user database for the user name and password and then informs the client whether the login was successful or not. The data set server is responsible for dealing with requests for data sets, eg. When a user requests a particular data set, the server fetches the data set and sends it to the client. The third server deals with the labels; the server gets requests for labels for a given data set and slice, it then checks the labels database if there exist any labels for the current data set and slice and sends the labels back to the client.

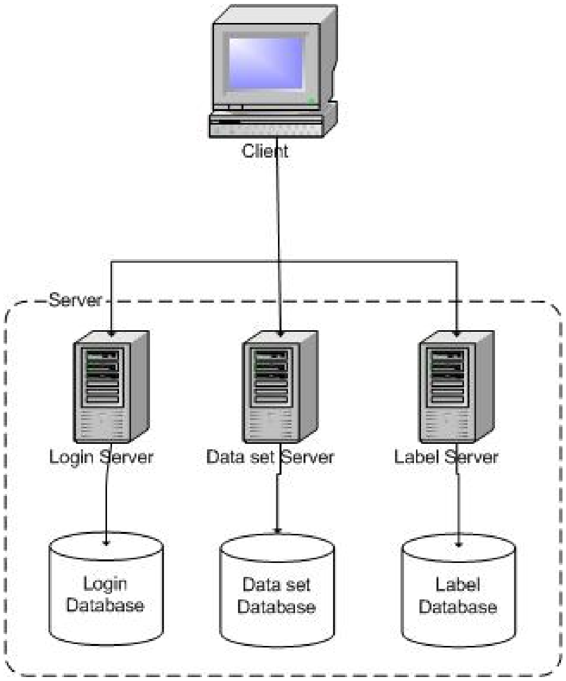


Figure 1

2. Goal:

The goal of this project is to develop a tool to enable a user to create an online anatomical atlas from a biomedical data set.

3. Description of the datasets:

Since we are providing a tool to let users create their own anatomical atlases using their own data sets, which can be in stored in many different formats, we must define a standard format of the data sets in order to make them compatible with the program. The data sets consist of two files:

1. A file storing the Meta data of the data set (file extension is .vol).

   This file stores the attributes belonging to the data set which consists of:

   - Name of the file storing the slice data

    - x, y and z size of the data set (ie. The number of pixels)
    - The real distances (in mm) between the pixels along the x, y and z-axes.
- The other file stores the slice data itself (file extension is .dat).
  Each slice is an eight bit grayscale image and the gray values are stored row by row, left to right. The x-axis is defined to be the axis going along the rows and the y-axis is defined to be the axis going down the rows (see Figure 2).

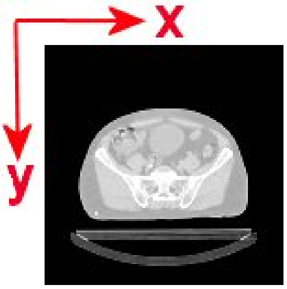


Figure 2 – definition of the x- and y-axis

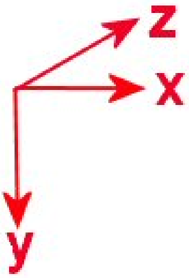


Figure 3 – the orientation of the co-ordinate system as used in this project

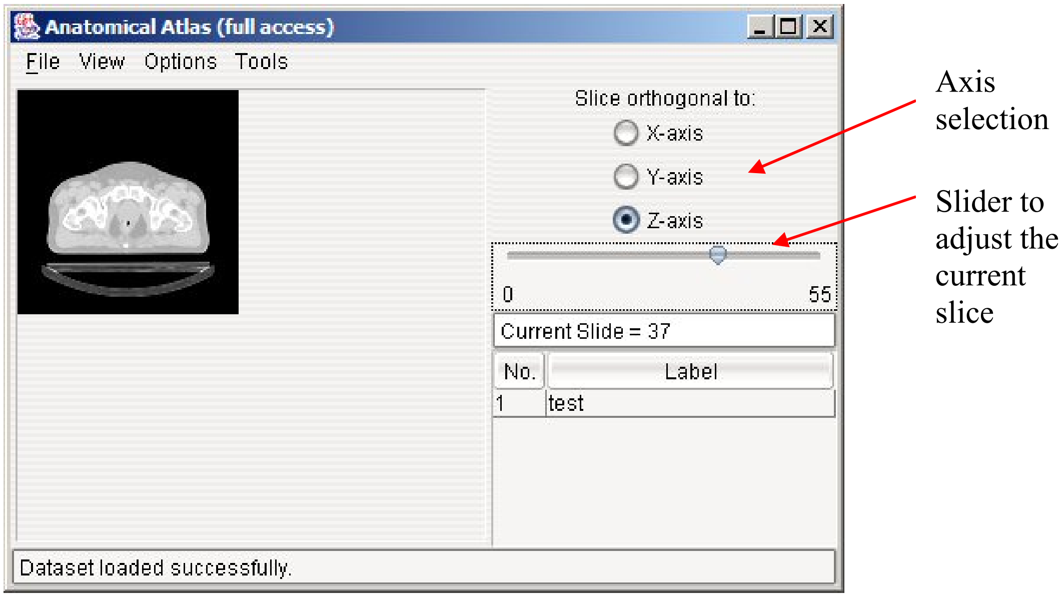
The first slice is defined to have a z-value of 0 and the z values increase as the slices are read in; on Figure 2 the z axis points into this page (see Figure 3).

4. Experimental Results / Program Functionalities:

This section describes the various features available in this project.

4.1 Navigating through the data set:

The single most important feature in this project is the navigation of the data set. Without it we would be looking at a static image of a single slice. To navigate through the slices the user can adjust the slider (see Figure 4).



Current Slide = 37

No.	Label
1	test

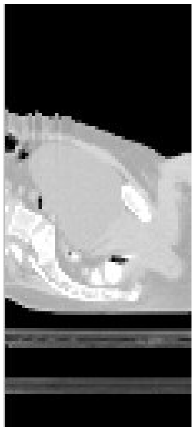
Dataset loaded successfully.

Axis selection

Slider to adjust the current slice

Figure 4 – the main window of the Anatomical Atlas

The slices viewed are orthogonal to the z axis when the program is first started. These are the slices that were read in when loading the data set. The user however can also slice the data set orthogonal to the other two axes (see Figures 5, 6, 7). When the data set is loaded, all the gray values of the slices are stored in a 3 dimensional array in the format [x] [y] [z]. By looking at a particular slice, we are taking a 2 dimensional subset of this array. The axis that is going to be orthogonal to the slice will be held constant while all the other values are extracted from the array. The value to be used as the constant is the current value of the slider. For example if we were slicing orthogonal to the y axis at y=10 then all the values in the array that have a y value of 10 are extracted in order to produce the 2 dimension slice, ie. All values satisfying [x] [10] [z] where 0 <= x < xSize and 0 <= z < zSize.



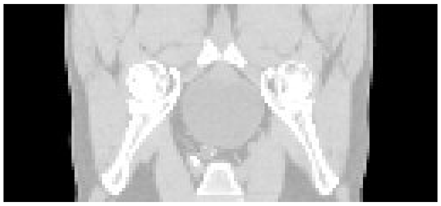


Figure 6 - slice orthogonal to the y-axis

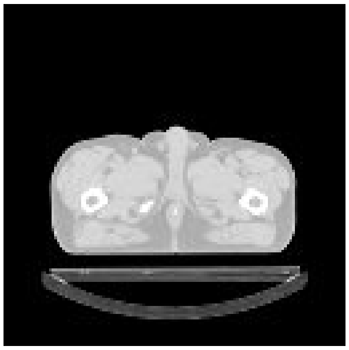


Figure 7 - slice orthogonal to the z-axis

Figure 5 – slice orthogonal to the x-axis

4.2 Viewing Options:

Some of the data sets that may be used can be of a relatively small size, in the case of the sample data set used during development it was only 128x128x56 pixels large, which results in very small slices when viewed on screen. A zoom functionality was added to the program to overcome this problem. A user can zoom in on the data set slice by clicking on “Zoom In” under “View” in the menu bar. Every time the “Zoom In” menu item is pressed the enlargement factor increases by 1. The enlargement factor determines the size of the slice image, for example, an enlargement factor of two means the width and height of the slice image displayed is twice as large as the original image slice, making the area of the enlarged image four times bigger than the original.

Another zoom feature that has been implemented is the “zoom box”, which allows a user to select a region of interest and zoom into it (see Figures 8 and 9). The zoom box can be drawn by pressing and holding down the left mouse button and releasing it once the area of interest has been selected. The zoom box can be applied multiple times, each time zooming in a bit further. By pressing the right mouse button we can zoom out again, ie. we go back to the previous zoom box view or if there isn’t a previous zoom box we zoom out completely.

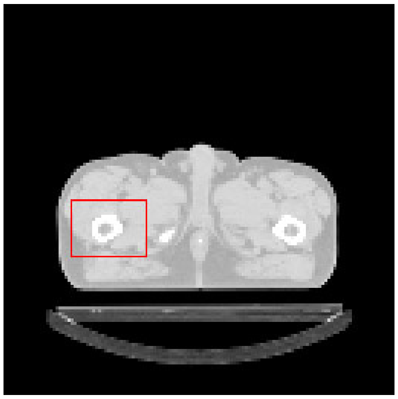


Figure 8 – the region of interest surround by the zoom box

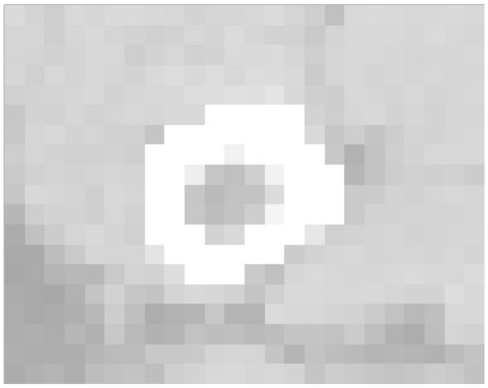
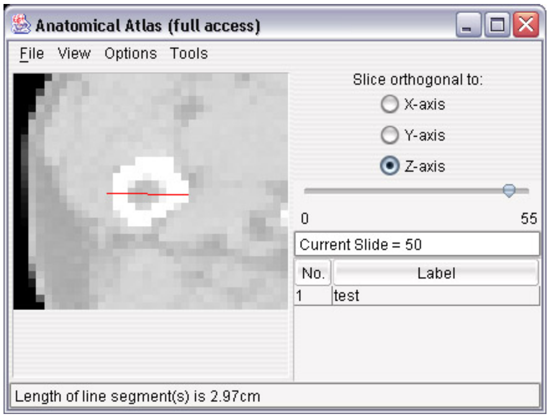


Figure 9 – after zooming in on the region of interest

4.3 Measuring Tools:

Looking at a data set slice on the screen makes it quite difficult when it comes to estimating distances and getting a sense for the scale. To help with this task, two measuring tools have been implemented, a line distance tool and an area tool.

- Line Distance Tool
  The line distance tool allows the user to measure the distance along a single line segment or multiple line segments joined together (see Figures 10 and 11).
  To enable (and disable) the line distance tool, the user must click on the "Line Distance Tool" menu item under "Tools" in the menu bar. The user can then select a start location for the line segment by pressing the left mouse button on the data set slice image. Once a start location has been selected, the user can see the current length of the line in the status bar. The length of the line is updated when the mouse pointer moves. If the user presses the left mouse button again, then the current position will be used as the end of current line segment and as the start of the next line segment. By pressing the right mouse button, the line segments are reset and a new start location can be selected.



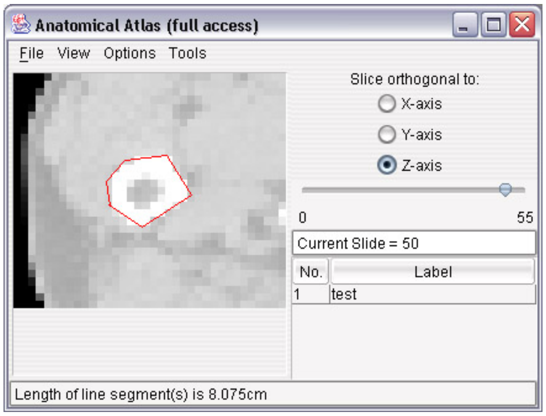
Slice orthogonal to: ● Z-axis

Current Slide = 50

No.	Label
1	test

Length of line segment(s) is 2.97cm

Figure 10 – The length of a single line segment



Slice orthogonal to: ● Z-axis

Current Slide = 50

No.	Label
1	test

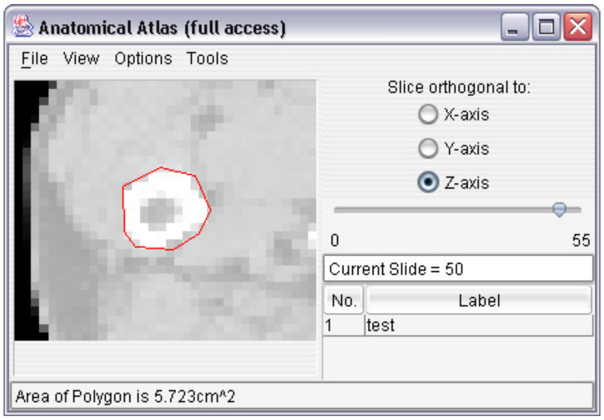
Length of line segment(s) is 8.075cm

Figure 11 – The length of the several joined up line segments

Displaying the length of the line segment(s) in pixels is of course not very helpful, so the distance between the selected points should correspond to the real distance. For example, from Figure 10, saying that the distance across the bone is 83 pixels doesn’t really tell us anything as opposed to saying that the distance across the bone was 2.9 cm. In order to find the real distance of a given line segment we must first find the corresponding points on the data slice without any zooming and then calculate the distance between the ‘x’ and ‘y’ components on the image. Depending on which axis the image slice is orthogonal to, we convert the ‘x’ and ‘y’ distances to the real distances using the “xDist”, “yDist”, “zDist” parameters specified in the meta data file (*.vol) of the data set. Now that we have the real distances along the ‘x’ and ‘y’ axis, we can use Pythagoras to find the straight line distance between the start and end point of the line segment.

- Polygon Area Tool

  The polygon area tool is another useful feature that lets the user select a region of interest and calculate the corresponding area. For example, it could be used to calculate the area of a cross-section of a tumor or a bone (see Figure 12).



Slice orthogonal to:
○ X-axis
○ Y-axis
● Z-axis

Current Slide = 50

No.	Label
1	test

Area of Polygon is 5.723cm^2

Figure 12 – The area of the cross-section of a bone

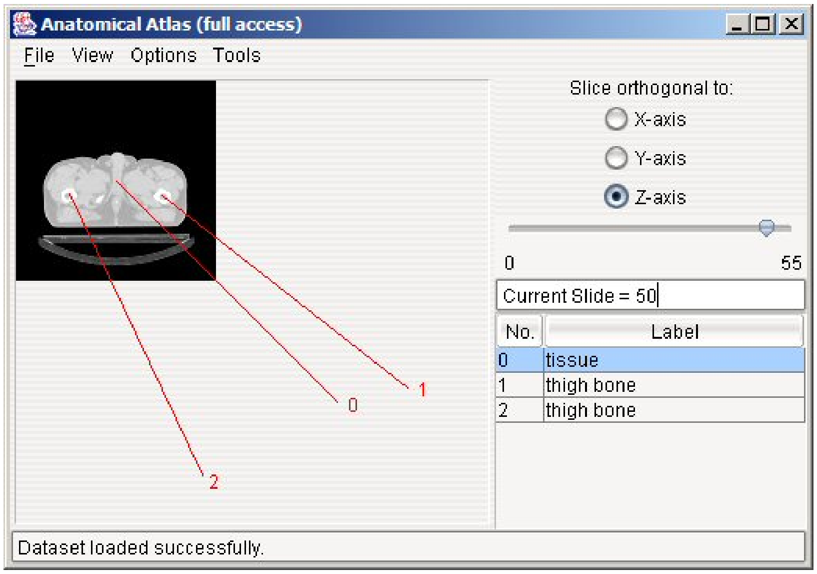
The user can enable (and disable) the polygon area tool by clicking on "Polygon Area Tool" under "Tools" in the menu bar. Once the polygon area tool is enabled, the user can select the region of interest by pressing the left mouse button to mark the points of the polygon. The points must be selected in a clockwise or counter clockwise fashion. By pressing the right mouse button the polygon can be reset. The area of the polygon is calculated in the following way:

$$A = \frac{1}{2}\left|\sum_{i=1}^{n-2}(P_i - P_0)\times(P_{i+1} - P_0)\right|$$

where $P_{0,}$ $P_{1,\ .....,}$ P_{n-1} are the vectors to the points of the polygon.
Similarly to the line distance tool, the points must first be converted to the real coordinates before they can be used in the area calculation.

4.4 Labels:

Allowing users to annotate their data sets by selecting and labeling regions of interest is a very useful feature as it allows other users, who may not be familiar with the data set, to recognize certain regions and structures. For example, labels could be used to point out diseased tissue which a user without much knowledge in this area wouldn't recognize. Figure 12 shows an example data set slice with labels.



Slice orthogonal to:

○ X-axis
○ Y-axis
● Z-axis

0 55

Current Slide = 50

No.	Label
0	tissue
1	thigh bone
2	thigh bone

Dataset loaded successfully.

Figure 13 – A data set slice with labels displayed

The user can choose whether to display the labels or not by clicking on "View Labels" under the "View" menu item in the menu bar. The labels are numbered and point to the corresponding location on the slice image (see Figure 13). The actual text of the label can be read from the label table on the right ride of the program. The reason for storing the label texts in the table is that crowding and overlapping of labels on the slice image is avoided.

Users with full (privileged) access can also add new labels by enabling (and disabling) the "Add Label" tool from the "Tools" menu in the menu bar. After enabling the label adding tool, the user can add a new label by clicking on the region of interest on the slice image. A dialog box will appear (see Figure 14) where the user can now enter the text of the label.

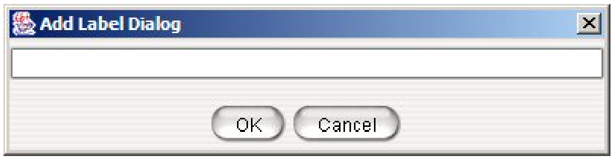


Figure 14 – the dialog frame for adding labels

As was mentioned earlier, the labels are all stored in a database. The labels are not stored for a particular slice but are points in a 3 dimensional space instead. This means that a label is visible no matter in which direction the slices are taken as long as the coordinates of the label location lie on the current image slice.

5. Future Work:

There are many additions that could be made to this application in order to improve its functionality. Some examples include:

- *"Magic wand"*
  The magic wand tool could be used in order to automatically find a region of interest and then calculate its area and perimeter length. For example, the user clicks on a bone and the program will automatically detect the region that belongs to the bone. This tool could also be extended to 3 dimensions, ie. The magic wand could find the volume and surface area of the structure selected.

- *"Growing Regions"*
  Another useful feature to include would be "growing regions". Once an area of interest has been selected, the area grows and shrinks as the user navigates through the slices. For example, if we had the cross-section of an egg shaped object, the area would grow as we move towards the centre and grow smaller again beyond it.

6. Conclusion:

There is great potential for online anatomical atlases like the one presented in this report. By residing on the internet, the atlas is available to literally millions of people across the globe. By allowing researchers to publish their own data sets, and allowing others to explore their data sets interactively, opportunities for feedback, collaboration, and promotion and development of research arise. It is also very useful as a teaching tool. The fact that the online anatomical atlas presented is quite flexible, allows users from a wide range of fields to make use of it, for instance, an MRI data set of a frog's anatomy could be used just as well as a 3D microscopy data set of an animal's eyeball. Being so flexible and online, it could potentially grow to a very large atlas indeed.

References:

[1] *"The Whole Brain Atlas Navigator"*, http://www.med.harvard.edu/AANLIB/cases/java/case.html

[2] *"Java Technology"*, http://java.sun.com/

[3] *"Introduction to Client/Server Networking"*, http://compnetworking.about.com/library/weekly/aa050201a.htm

Further Information:

[i] *"A digital brain atlas for surgical planning, model driven segmentation and teaching"*, http://splweb.bwh.harvard.edu:8000/pages/papers/atlas/text.html

Appendix:

Sample data set vol file:

The following box shows the contents of a sample data set .vol file:
*“Mystery.vol”*contents:

```
filename=Mystery.dat
xsize=128
ysize=128
zsize=56
xDist=2.7
yDist=2.7
zDist=2.7
```

Setting up the databases

Both the label server and the login server rely on a database to store their data. For the servers to be able to access the databases, they must first be registered as a data source. Using Windows, the steps to do so are as follows:

1. Open the control panel and select “Administrative Tools”
2. Double click on “Data Sources (ODBC)”
3. Click on “Add”
4. Select the “Microsoft Access Driver (*.mdb)” and click on finish
5. A dialog frame will pop up. As data source name use “Labels” (or “Users for the user database).
6. Click on “Select…” and browse for the database file (“labels.mdb” for the label database and “users.mdb” for the user database)
7. Click “OK” several times.

8. Repeat for the other database.

The dataset server currently does not rely on a database; it gets the datasets from its hard drive and then sends them to the client.

Setting up and starting the servers

All that is needed to start the servers is:

- Run "`java LoginServer`" from the command prompt in the folder of the login server
- Run "`java LabelServer`" from the command prompt in the folder of the label server
- Run "`java DatasetServer`" from the command prompt in the folder of the dataset server

By default, the login server uses port 7878, the label server uses port 7880 and the dataset server uses port 7879. The ports can be changed by modifying the "PORT" variable in the source code and recompiling it (ie. LoginServer.java, Dataset.java and LabelServer.java)

Running Instructions

The client application can be run in two ways, as an applet inside a web browser or alternatively users with full access can run the application as a stand alone program. To run it as an applet, start the servers and open "LoginScreen.html" in a java capable web browser (see Figure 15).

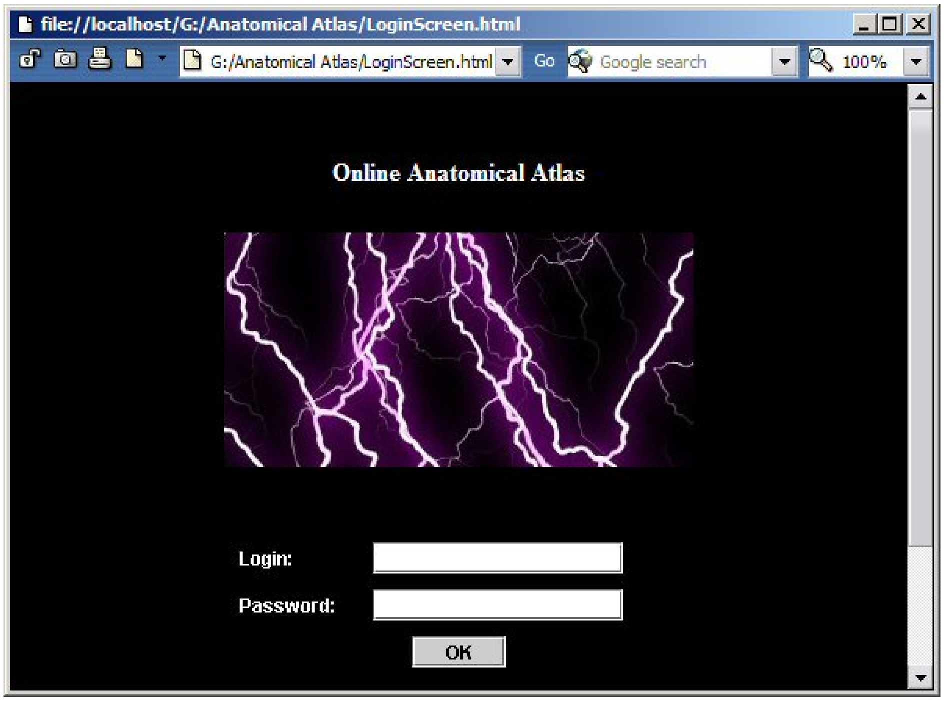

file://localhost/G:/Anatomical Atlas/LoginScreen.html
G:/Anatomical Atlas/LoginScreen.html
Go
Google search
100%
Online Anatomical Atlas
Login:
Password:
OK

Figure 15 – The login screen website

If the user is recognized then the Anatomical Atlas applet frame will pop up. Once the applet is open, a data set can be loaded by clicking on "Open over internet" and selecting one of the available data sets. Note that it is extremely slow to load the dataset this way.

Users with full access may have been given the chance to download the Anatomical Atlas program in which case they can just run it by typing:
"`java AtlasFrame`" in the command prompt in the folder where the program is residing.
Using the downloaded version has several additional features over the online version that couldn't be implemented in the online version due to the java applet security restrictions. The application version includes the use of a skin to improve the look of the program but more importantly it also allows the loading of data sets stored locally, which can be loaded much quicker than the data sets over the internet.