

THE UNIVERSITY OF MELBOURNE
DEPARTMENT OF COMPUTER SCIENCE AND SOFTWARE ENGINEERING
433-371 Interactive Systems Design
Project Specification — 2003

The object of this project is for you to gain some experience in designing and implementing a graphical user interface (GUI). This will be a group project, done in teams of three or four—for details see below. The project is due at 12 noon on Tuesday, 21 October 2003, and is worth 30% of your mark for 371. Additional information about the project, which cannot be provided here, will be made available from the project page of the 371 website, which constitutes part of the project specification.

1 The Task

Your team's task is to design and implement a Java/Swing application to help people put their digital photos on the Web. This is deliberately a somewhat open-ended project, in which you can exercise your creativity. However, the minimum requirements for a good mark would entail that your application:

- Provide some mechanism by which users can select which images are to be included in the photo album, where the images are stored as files in the filesystem.
- Allow users to associate an optional text description and date with each image.
- Allow users to perform simple manipulations of the images to make them more suitable for presentation. These should include:
 - cropping out an upright rectangular area of interest;
 - resizing to a particular size, while preserving aspect ratio;
 - rotation (by multiples of 90 degrees only) to allow for images scanned or photographed sideways;
 - adjustment of brightness, to allow for images that are too dark or too bright.

The manipulations should affect only the image to be displayed on the web and *not* change the original image.

- Automatically create a coherent set of HTML webpages to display the resulting images along with associated descriptions and dates. The webpages should be to some extent customizable, say to allow users to provide their own page titles and headings.

All these capabilities should be provided within a framework of good interface usability, both for the

application and the resulting webpages. For the webpages, accessibility will be another consideration.

To receive a high mark you would need to go beyond these minimum requirements in terms of functionality or usability, preferably both.

2 Arrangements

As mentioned, this project is to be done in teams of three or four. You should organize your own team, and a representative of the team should send mail to <cdmcc+371reg@cs.mu.oz.au> to register the usernames of all your team members. The body of your mail message should consist only of a list of the usernames of the team members, one per line. (This is so the registration can be largely automated.) You should do this between Wednesday, 3 September and Friday, 12 September. Earlier registrations will be ignored; later registrations will incur a one-mark penalty. Chris will confirm by mail the registration of your team, tell you the Unix group created for your team, and the directory in which you're to put your team's CVS repository. Note that you will need to develop your application's sources under CVS control as a CVS module called `WebPhoto`, and you should include a suitable makefile to do all necessary compilations, etc., to build your executable application. Required makefile conventions will be posted on the project webpage.

Your application should be accessible via a command `webphoto`. This will most likely be a shell script which invokes Java appropriately, setting up any needed command-line arguments or environment variables, but how you implement it is up to you (for example, it could be a Perl script instead), so long as the behavior is as required.

Your project should also include a justification of your design, explaining your choice of functions, layout and GUI components. This justification should be formatted in appropriate HTML and be equivalent in length to about 2 to 4 printed A4 pages, and should be in a file called `why.html`.

3 Submission and Assessment

Details of submission procedures will be advised later. However, keep in mind that the assessment of your project will be based largely on a live demonstration to be conducted after submission during the last weeks of semester. Arrangements for timetabling the demonstrations will be advised later.

4 Notes

- The work submitted by a team must be done entirely by the members of that team. Violations will be treated as academic misconduct, and pursued under applicable University disciplinary procedures. If you plan to make use of non-standard libraries, you will need to get permission from Les Kitchen or Chris McCarthy.
- It is expected that all members of a team will make equal contributions to the team's submission, and normally all team members will receive the same mark. This will be assessed during the demonstration and by inspection of the code and CVS logs.
- It is essential that you maintain a CVS log of the development of your project. Part of the mark for the project (5 marks) will be based on the project's having a reasonable history of development, as evidenced by the CVS log entries. **Projects without a reasonable history of development will receive a failing mark**, no matter what the quality of the rest of the work. In practice, this means that if you do the right thing, and check-in reasonably during the course of development (e.g., for completion of each significant section of code, and subsequent bug fixes), you will get an easy 5 marks.
- Your application will be expected to run under Java on the Department's Sun x86 Solaris machines, namely mungee, murang, murree, or queeg. You are welcome to do most of your development on your own computer using Java SDK under say Linux, MacOSX, or Windows (possibly with CygWin). However, you will need to check-in your files to your CS CVS repository from time to time during development, so as to establish a proper history of development. You will also need to verify that your application does in fact work as expected on the CS machines (in case of minor differences across platforms).
- Because of the constraints of the marking schedule, it will not normally be possible to grant any extensions. Unless there are extraordinary circumstances, **late submissions** will receive a **zero mark**.

If the performance of your team is significantly impaired by documented illness or misadventure, you should notify Chris McCarthy as soon as you become aware of the circumstances, so that they can be taken into account in your assessment.

- Fully implementing all the required functionality in a self-contained application

would be way too ambitious, even for such excellent programmers as yourselves. You will need to exploit existing functionality, both in Java libraries and in other external programs. For example, your application need only create the webpages: the user can use a separate browser for viewing them. Image processing can be done by various Java packages, such as `java.awt.image`, or handed off from your Java code to utilities such as `djpeg/cjpeg/jpegtran` for decompressing/compressing/transforming JPEG images, and those provided by Netpbm for various other operations, like `pnmcut`, `pnm-scale`, `pnmflip`, `pnmgamma`. All these programs are installed on CSSE machines and are otherwise widely available. See the `exec` method of `java.lang.Runtime`.

- You can assume that all original and displayed images will be in JPEG (JFIF) format. Some sample images will be made available via the 371 project webpage.
- Image operations, particularly for large images, are very costly. It will require some cleverness on your part to devise ways of providing users with adequate feedback, without incurring unacceptable computational costs, particularly on a shared machine.
- You might get some ideas about what functions to provide and how to set up the user interface by looking at other applications that provide similar functionality. However, you should not feel that you have to provide exactly the same functionality as such applications, nor should you feel that you have to use the same style of interface.
- The project specification may require changes and clarification between now and the due date. Announcements will be made in lectures, via the subject webpage or via `cs.371`, as appropriate. It is your responsibility to be aware of any such announcements.

Les Kitchen
29 August 2003

\$Id: spec2003.tex,v 1.4 2003/08/29 03:14:38 ljk Exp \$