Assignment #5 — FacePamphlet Requirements

The three figures in this handout summarize the requirements that you need to meet when you are implementing the three regions of the FacePamphlet application. Figure 1 shows the structure of the center graphics panel you implement for Milestone #5, and Figures 2 and 3 summarize both the requirements and action information for the two side panels.

Figure 1. Requirements for the center graphics panel

- **This GLabel** displays the full name of the view user. The font and location information is provided in the **FPConstants** interface.

- **The GImage** displayed here must be scaled so that it fits in the available space. The horizontal space is given by the left and right margins. The limits on vertical space are specified in the **FPConstants** interface. If the image is too large in either the x or the y dimension, it should be scaled **linearly** so that it fits. The image, however, should never be expanded.

- **This GLabel** displays the view user’s status, which is really just the string after the first name of the user and the word is. If no status exists, this line should simply be absent.

- **This GLabel** appears only if an error has been reported by the server. In this case, I have issued a friend request for piech instead of cpiech, and the server has notified my of the error. This line should disappear when the graphics area is next updated..
Figure 2. Requirements and actions for the west side panel

- This button is the “home button” indicating the full name of the current user, i.e., the user who is logged into the server. Clicking this button should restore the view to that of the current user.

- This scrollable list shows the ids of your current friends. Double-clicking on a name should have the same effect as selecting a name and then clicking Visit.

- This button changes the view user, which is the user whose information is listed in the center panel and the wall. The code for the button should bring up the information for the name currently selected in the friends list. If no name is selected, it should restore the view to that of the current user.

- This text field contains the id of the user you are inviting to be your friend. Hitting the ENTER key in this field should have the same effect as clicking the Request button.

- This button issues a friend request to the user whose id appears in the text field above the button. Clicking the button should make issue a friend request on the repository and then clear the contents of the request field.

- This scrollable list shows the ids of the users who have asked to be your friends. This information must be updated by calling the appropriate method in the FPRepository class. Double-clicking should have the same effect as the Accept button.

- This button accepts as a friend the user whose id is currently selected in the requests list. If no name is selected, clicking Accept should have no effect.

- This button rejects the friend request, but is otherwise the same as Accept.

- This text field contains the string you want to set as your status.

- This button changes your status in the repository and clears the status text field.

- This text field contains the file name or URL of the image you want to use.

- This button updates your image in the repository and clears the image text field. If creating the image causes an error, you should report that in the message area.
Figure 3. Requirements and actions for the east side panel

- The label on this panel must change so that it includes the first name of the view user.

- This scrollable text area is used to display the contents of the view user’s wall. The text area is not editable, so the user cannot type into it, although the program can call `setText` to change its contents. Note that you don’t have to manage the structure of the contents yourself in terms of adding signature lines and the like. What you do as the programmer on the client side is call `sendMessage` to update the contents of the wall, then call `getProperty` to retrieve the entire text, and finally call `setText` to update the display.

- This scrollable text area is the editable region in which you type messages you are sending to other user’s walls. This area generates no events.

- This button writes the text in the message area on the wall for the view user. Clicking this button should also clear the contents of the message area.