Initial Comments About the ESME (Enterprise Social Messaging Experiment) Project

v1 r15
14-November-2008

by Bill Fernandez

Bill Fernandez Design
5713 Guadalupe Trail NW
Albuquerque, NM 87107
(505) 346-3080
bill@billfernandez.com

Revision	Date	Status	Revision Description
v1 r14	4-NOV-08	done	First draft.
v1 r15	14-NOV-08	done	Added addendum.

INTRODUCTION

My friend David Pollak asked if I could take a look at ESME from my perspective as a User Interface Architect, and offer what comments or suggestions seemed appropriate. Having some time available I agreed.

After studying two example ESME UIs (at http://esme.us, and http://esme.mrinalwadhwa.org/), the ESME blog, Twitter, and consulting with Marcia Conner (a micro-messaging expert), it became clear that simply making a few comments about the current ESME UI would not be very useful.

So I have gone so far as to develop and suggest an overall UI architecture that tries to address (or provide a logical place to address) all of the issues I was able to identify in my research.

WHAT'S IN THIS DOCUMENT

This document starts with a summary of the background and rationale for ESME as I have come to understand it (corrections and clarifications are welcome).

It continues with a description of some conceptual approaches I've adopted for dealing with issues such as groups, conversations and affiliates (defined later).

And the bulk of this document maps out a conceptual approach to structuring a web-based UI for ESME.

Note that this document addresses only the front-end, end-user UI — not any back-end, installation or system administration facilities (which would also be needed as part of a comprehensive system design).

Note that this document maps out the structure of a UI based upon a certain conceptual approach. It is not a complete UI design or specification, and it is certainly not a visual design.

If the team likes the approach mapped out in this document I can certainly flesh out the details into a full-fledged UI specification.

Also note that the UI concept is uneven in depth: In some areas I've sketched out the elements of the UI in detail, and in other areas I've just drawn a box and said that an appropriate UI should be placed in that area. In some cases this is because more detail was need to explain the concept, in others because I got on a roll. In either case, remember that my objective is merely to suggest a collection of ideas for consideration by the ESME team.

USER EXPERIENCE PRIORITIES

In user interface work it's important to have a clear sense of what aspects of the user experience are most important. Here are the main thoughts I had in mind.

The Usual Suspects

• It must have the features and capabilities that will be needed by the anticipated user base.

• As much as possible it must be learnable, usable, intuitive, explorable, predictable, rewarding to use, etc.

• ESME should be a "good citizen" on the desktop, allowing the user to decide what degree of prominence to give it in his/her daily life.

Scalability
Given the intended enterprise-wide scope of the intended user base, we must take into consideration the user experience dimensions along which

the sytem must scale.

Screen size — The UI should work well on the smallest anticipated screen size (I suggest we assume 1024 x 768 pixels), but also work gracefully and take advantage of larger screen sizes.

Effort — We should anticipate a range of users from those who are technology adverse and who want to avoid learning anything new, to those who have the ability and desire to embrace any promising new piece of technology.

Interaction Style — We should anticipate a range from those users who only point and click, to those users who prefer to type (command-line style) as much as possible.

Devotion — We should anticipate a range of users from those who will use ESME as little as they can get away with, to those who act addicted to it. for the former we will want the UI to be simple and unobtrusive, for the latter we will want the UI to put all of ESME's power at their fingertips.

INTERNATIONALIZATION

Except to acknowledge that not all users speak the same language, and that you might wish to choose to see only messages in a language you understand, there is no attempt at this time to address internationalization issues.

The Rationale for ESME

INTRODUCTION

I always like to start a project by trying to understand what it’s intended to accomplish, for whom, why, and how it came to be.

Here is my summary of the understanding I have come to. It is a combination of what I have learned through questioning and investigation, and have filled in blanks with assumptions that made sense to me.

Therefore I would not be at all surprised if those more familiar with Twitter and ESME disagree with aspects of what I’ve written, and I’d welcome correction and clarification.

BACKGROUND AND RATIONALE

The Birth of Twitter

Twitter began in March 2006 as a way for people to stay in touch with each other by posting brief status messages. Since becoming open to the public it has developed a worldwide community of users who post messages of many different types, including:

- This is what I‘m doing (“I’m re-installing my operating system.”)
- This is how I feel (“It feels great to be in love again.”)
- Thoughts (“I wonder who I‘d be if I wasn’t me.”)
- Humor (“How do you drive a Twitter-user crazy? I‘ll tell you tomorrow.”)
- Questions (“What does ESME stand for?”)
- Answers (“ESME: Enterprise Social Messaging Experiment.”)
- Coordination (“Let‘s meet at noon at the sound entrance.”)
- Announcements (“My wife just gave birth...”)
- Alerts (“A plane just flew into the World Trade Center!”)
- Recommendations (“You’ve got to try the Greek pizza at Alfredo‘s.”)
- Etc.

Twitter lets you designate a list of users you want to “follow”. You can then switch between viewing the global message stream, and seeing just the stream of messages from the users you follow. This latter gives you a smaller stream of messages that combines messages from your circle of friends and associates with those of “pundits” whose postings interest you.

Twitter lets you direct a message to a named user, which both posts it publicly and insures that the named user is alerted to your message. On the other hand, there is no way to direct a message to a named group of users (the workaround is to list all intended recipients in your message).

Twitter lets you send a private message to a named individual, and lets you view lists of private messages you’ve sent and received. but it does not have a way to track individual “conversations”, or to have a private discussion among a group of users.

Twitter lets you view certain information about each user, including some brief biographical and contact data, and their lists of those they follow and those who follow them.

At present there is no official way to designate the topic of a message or to view all messages on a given topic.

The Extended Uses of Twitter

Because of its immediacy, simplicity and universality, and despite the sparseness of its features, Twitter has been used to form project groups, to coordinate group efforts, to share knowledge, to accelerate response to events, to disseminate impactful news, and to create a sense of camaraderie and social cohesion among geographically and culturally diverse individuals who in most cases never meet face-to-face. These uses go far beyond, and have impact far greater than, the founding use of keeping in touch through constant, terse status updates.

Twitter and Enterprise

Many of the results that Twitter users have achieved would be of great benefit to enterprises: increasing efficiency, reducing response times, sharing knowledge, building social cohesion, etc. But Twitter is not an enterprise-suitable platform; primarily because it does not give an enterprise any control over the server operation, the message ownership and retention, the users who are authorized to use the system, or the scope of message distribution.

The Rational for ESME

It is proposed that micro-messaging can be of great benefit in enterprise, but that public services such as Twitter are unsuitable for enterprise use. At the time ESME started there were no enterprise-oriented alternatives to Twitter (although several projects have since emerged), but there were a number of individuals from around the world who were both passionate and dedicated enough to initiate a project designed to produce a micro-messaging system suitable for enterprise deployment.

ESME is an acronym of Enterprise Social Messaging Experiment:

- Experiment because it is uncertain whether contemporary enterprises are ready to integrate social computing into their corporate cultures.
- Social Messaging because it is the essence of micro-messaging to build a sense of community through inter-personal messaging.
- Enterprise because this project is designed for enterprise deployment.

SOME KEY ASPECTS OF THE ESME PROJECT

Participants

About twenty people from around the globe...

Goal

The goal of the ESME project is to produce an open-source system that any enterprise can deploy to give them a robust and effective micro-messaging infrastructure.

Server Infrastructure

- Java
- Scala
- Lift
- etc.

End-User Infrastructure

- Modern web browser
- JavaScript turned on

Anticipated Benefits

When properly installed and used, a micro-messaging system can produce both tangible and intangible benefits in these areas:

- Social cohesion across a geographically distributed organization.
- Organizational responsiveness to changing

conditions.

- Knowledge sharing.
- Troubleshooting, problem resolution.
- Team coordination.
- Information security.
- Historical record-keeping.
- and the requisite “and more...”

How is ESME the Same as Twitter?

- Primary client platform is the web browser.
- Messages are brief. Reading consists cognitively of “scanning headlines”.
- Messages are broadcast to all users in their scope.
- Users may “follow” the posts of other users.
- Users may view certain information about other users (details TBD).

How is ESME Expected to be Less that Twitter?

- The user pool is restricted to those authorized by the organization.
- Most deployments may limit access to be only from within the intranet.
- No SMS integration(?)

How is ESME Expected to be More than Twitter?

- Puts the enterprise in control of its server(s).
- Puts the enterprise in control of its data.
- Puts the enterprise in control of its user base.
- Puts the enterprise in control of its network accessibility.
- Keeps messages private to the organization.
- Adds support for public and private user groups (of some kind).
- Provides a better solution for private individual and group conversations.
- Adds message tagging.
- Provides better support for discovery of “interesting” messages (through tag clouds, etc.)
- Provides better message-filtering capabilities.
- Provides better search capability.
- Adds audit trail support.
- Adds retention policy support.
- Allows users to create automated responses to messages (called “actions”. analogous to email “rules”.).
- User base can be defined and managed through corporate directory and authentication infrastructure (e.g. LDAP).
- Better support for embedded references (URLs, etc.) in messages.
- Supports integration with “services” (user interactions with other servers mediated by ESME).
- Live updates (users don‘t have to manually refresh their web pages).
- Supports federation between ESME servers.
- Perhaps supports venn-diagram-like relationships between “affiliates” (this is a Bill Fernandez concept that is not yet fully formed).

Business Model

ESME is an open source software project. All contributions are owned by their creators and open-source-licensed to the ESME project. (Currently using the Apache 2.x license?)

ESME Rationale

Beyond the open-source distribution, any person or company is free to provide fee-based services and extensions as long as they respect all terms of the open-source license.

Conceptual Model

CONCEPTUAL MODEL

After gathering and considering as much initial information as I could about the ESME project, and about Twitter with which it is most often compared and contrasted, I developed a family of concepts that I hope captures, clarifies and positions all of the ESME end-user needs in a way that can be readily designed to. Here they are.

ORGANIZATION

A primary goal of ESME is to empower organizations to own, operate and control their own micro-messaging system. Therefore a fundamental concept is that an ESME instance has an "organization" behind it.

Although ESME is generally described as being "enterprise" oriented, I've chosen the term "organization" because I believe ESME can scale from non-profit to for-profit, volunteer to paid staff, small business to multi-national corporation; and that "organization" is the best single word that captures the common essence across this range.

USERS

The primary function of ESME is to enable people to communicate with each other through micro-messaging. Therefore we have the concept of "users". We anticipate that the vast majority of users will be human beings, but we acknowledge that there will be times when it might be useful let humans interact through micro-messaging with software servers, and that it might be useful to let these servers masquerade as (human) users. (See: Services.)

USER POOLS

Whereas Twitter is primarily a global, public "party line" of communication, ESME is designed to meet the needs of (primarily business) organizations.

It is expected that these organizations will have a pool of "internal users" comprised of all members of the organization, and that share a public party-line of communications.

However I also believe that for ESME to achieve its potential it will have to support private pools of users who can communicate privately (but still using the party-line paradigm) among themselves. Such user pools might include project teams, or the set of all "C" level executives, or an internal liaison paired with an external set of users at an outside vendor, contractor or partner.

The common element is that ESME needs to be able to support "pools" of users within which messaging is public, but between which messaging is private (or hidden).

Any user may be a member of multiple user pools, but he/she may only post to one user pool at a time.

See: Groups, Affiliates, Message Pools.

MESSAGES

ESME is a micro-messaging system, so "messages" are at its core. Messages are short text strings (Twitter limits them to 140 characters), with certain metadata attached, and with certain conditions limiting who can see them, etc.

Primarily messages are intended to be read by humans, but there are special terms that can be embedded into messages that give directions to the ESME system, and there are also messages intended to be received, parsed and responded to by software servers (see: Services).

MESSAGE POSTING

Twitter asks users to answer “What are you doing?” with an “Update”.

The ESME sites I saw asks users to answer “What are you working on?” with “Send Message” (the button label).

I suggest since micro-messages can and will be used for such a wide variety of uses, that we should not ask a question. We should model these as postings (as in “users post messages that others can read”), and that we simply have a message box where you type your message and a “Post” button to post it.

MESSAGE POOLS

Essentially, there is one message pool for each user pool. The important concept about message pools is that unlike other messaging and discussion systems all messages posted by a given user pool are placed within a common pool of messages against which searches and sorts can be performed.

The result of this is that there is no fixed linking structure relating messages, and there is no one single “right” or “best” way to display a set of messages.

Messages are associated by their metadata, not by any sort of fixed linking structure. Thus sets of messages are gathered for display, statistics, etc. by finding all messages that meet certain metadata requirements.

In general ESME tends to display sets of messages as chronological lists of messages.

MESSAGE STREAMS

Most of the time, what users want to see, and what ESME produces, is chronological lists of messages.

These lists are constructed by finding all messages within a given pool that share a set of attributes (metadata) that are meaningful to human users.

I use the term “message stream” to refer to these time-ordered sets of messages.

ESME has some message streams built-in, such as: The stream of all messages I have posted, plus all those posted by users I “follow” (this would be the primary personal stream of each user). Or the stream of all messages posted by internal users (this would be the “public” stream within the organization).

GROUPS

I believe organizations will want to collect users in to groups of two kinds: (1) Those whose communications are accessible to a larger pool in which the group exists (e.g. the organization’s internal-users pool), and (2) Those whose communications should only be accessible to members of the group (and hence which have their own private message pools).

I suggest we call these “private groups” and “public groups”.

Group Creation

I think it should be possible for any user to create both private and public groups (although we may want to refine this as we think things out).

Group Management

I suggest that the creator of a group be it's first manager. This user should be able to add additional co-managers, all of which would have equal powers. If the original creator leaves the group, the remaining manager(s) still have full administrative privileges. Thus there is no notion of "ownership" of a group, only of "control" thereof.

Group Accessibility

All public groups should be discoverable, through searching, etc. by all users within the group's larger user pool.

Private groups should have properties that can be set to make them visible or invisible to users who are not group members.

More...

The foregoing is not meant to be a complete functional specification for groups — just to give enough detail to sketch out the major concepts.

AFFILIATES

I believe that organizations will quickly come to desire the ability to extend the benefits of micro-messaging to their interactions with outside users.

For example: Vendors, key customers, partners in joint activities, consultants, contractors, etc.

The General Idea

The general idea is that ESME should support a notion I am calling "affiliates". Basically an affiliate is a private group that has members from both inside and outside the organization. This concept is flexible enough to support a range from two users who act as liaisons between two companies, to two project teams working on a joint project.

An important aspect this concept is that these "affilate" groups should be labeled clearly as being distinct from the fully internal groups.

Individual Contractors

Rather than having to set up an "affiliate" private group to allow an external contractor to work on an internal project, it would probably be better to simply allow private groups to include external users. But then a great deal of care will have to be taken, both with the user access control technology and with the social engineering aspects to insure that these outside users do not gain access to confidential internal communications.

PRIVACY AND TRUST

Perhaps the most important way in which ESME differs from Twitter is that its user base is constrained to contain only users that the organization has vetted and authorized.

This establishes a very important assumption within the minds of its users: that they can trust the other users on the system to be appropriate recipients of information that is confidential to the organization, and that therefore they can communicate with a large degree of freedom on most topics.

SERVICES

Although most "users" of esme are human, ESME supports the integration of software servers in such a way that human users can communicate with them through micro-messaging.

I have chosen to refer to all variations on this theme as “services”.

Some Examples

- Ask the price of a given stock.
- Get a weather forecast for a given region and date range.
- Check the status of a network device. Change some of its settings. Reboot it.
- Ask for the contact info of a given customer in the corporate sales CRM.
- Update your status for a given task within a given project in the corporate project management system.

The above examples are intended to illustrate the wide range of possibilities; from those that might serve fairly naive users (such as checking stock prices) to those that are highly technical (like rebooting a network server).

Command Line Interaction

Services make ESME message streams act like a command line interface, not unlike the UNIX command line interface, where human users issue control and interrogation commands to ESME-integrated servers.

The simple cases are not unlike what are often called “bots” in the Twitter world which are usually very simplistic. For example:

User: @stock AAPL

Stock: APPL high $105, low $100, close 102

In the above “conversation” a user directs a message to a service masquerading as a (human) user named “stock”. The service replies to the user with the requested information.

More sophisticated services will need a more sophisticated command syntax. We can look to Unix to see examples of what ultimately we will have to support, with the ability to set options and supply value parameters.

Command Line Syntax

I think it will be important to adopt an adequatly flexible and universal, standard syntax for services, so that users only need to learn one set of rules for formulating command strings (although of course each service will have its own set of commands, parameters and values).

It would also be good if there was a common way to automatically generate GUIs for formulating service commands, so that users could invoke services from a point and click (rather than command line) user interface.

Reporting Services

There is also another category of service. This is the kind that only transmits information into a given user pool. This kind of service would be used for news feeds, event announcements, status messages, etc.

These should all be tagged in a consistent way, and should only appear in users’ message streams on demand.

Privacy and Anti-Pollution

We might want to treat user invocations of and interactions with services as private conversations; both to keep from polluting the shared message pool, and to keep each user’s activities private.

CONVERSATIONS

The core paradigm of Twitter-like micro-messaging is that all messages are publicly available to all users.

We have extended this approach to include the concept of supporting multiple, exclusive pools of users (and letting users be members of multiple pools if appropriate). This supports the organizational needs to support certain groups of users having free but private and confidential communication within them.

Now it's time to introduce some final concepts in this area.

Directed Public Messages

Twitter allows you to insure that a given user sees your message. This is done by using an embedded command, which is Twitter is to begin a message with and "@" character followed immediately by the user name of the intended recipient. For example:

> @joe Hey Joe, when are you going to return the chair you borrowed from my office?

In Twitter, there are two ways to accomplish this. One is simply to type in the command at the beginning of a messsage. In this way you can "send a new message" to someone. The other is to "reply" to an existing message. This is done by clicking a "Reply" button, which prefills the beginning of a new message with the "@" sign and the name of the sender of the message to which you are replying.

In short, Twitter uses the same basic mechanism (@username) to support two intents: "send to" and "reply to".

We need to support these two functions in a similarly simple and straightforward way.

Private Messages

Twitter also allows an extension to the above command, which is:

> "d" + space + "@" + username

For example:

> d @joe

This causes the message to be sent privately to the specified user.

ESME needs to support the notion of private messaging, but I think we need to do better than Twitter (see next section).

Conversations

Whether these directed messages are sent publicly or privately, ESME needs to support the notion of "conversations" or threads of connected messages.

To use an analogy, Twittter behaves in the same way that most mobile phones handle SMS text messages: All messages, to/from all users, pertaining to all topics, are displayed in one common chronological message stream. I think ESME needs to handle sets of related messages the way the Apple iPhone handles SMS conversations, and to group each set of related messages into its own, separate converational thread. (This is also how instant messaging programs behave.)

User Interface Preliminaries

THE ONE-WINDOW ASSUMPTION

First, I'm assuming that the entire ESME end-user UI will be displayed within a single window, and that it will replace the login page when a login has been successful.

I'm assuming that you can have multiple ESME windows open, but that each will have a duplicate ESME UI (rather than each having a fragment of the UI.

Note that I suggest the user guide be displayed in a separate window, and I would expect that when we build a system administrator UI it will be in other window(s) as well.

SCREEN REAL ESTATE USAGE

I believe that across an enterprise-wide user base the vast majority of users will use ESME:

- little and rarely, or

- often but as an ancillary service alongside several other applications.

Having a window that monopolizes the desktop would not be acceptable to these these categories of users, thus arguing for a small-window design.

On the other hand, there will be some users who will spend a large portion of their time in communicating across the organization through ESME, and will make heavier use of its extended features.

To support these users we should it possible for them to productively devote more of their screen space to give quick access to ESME features.

UI Design Strategies

Given the above requirements, the primary strategies I have used in this design are to:

(1) Make the minimum interactive UI as small and simple as possible,

(2) Let additional features and capabilities be shown/hidden as the user desires,

(3) Make the ESME window resizable (and make the UI within scale gracefully to use the space), so that users can adjust it to fit comfortably into their work environment,

(4) Let the ESME window toggle between two modes: expanded and shrunken.

Expanded vs Shrunken Mode

In shrunken mode the ESME window should be very small — only big enough to display activity indicators which alert the user to the existence of new messages, and perhaps hinting at their urgency.

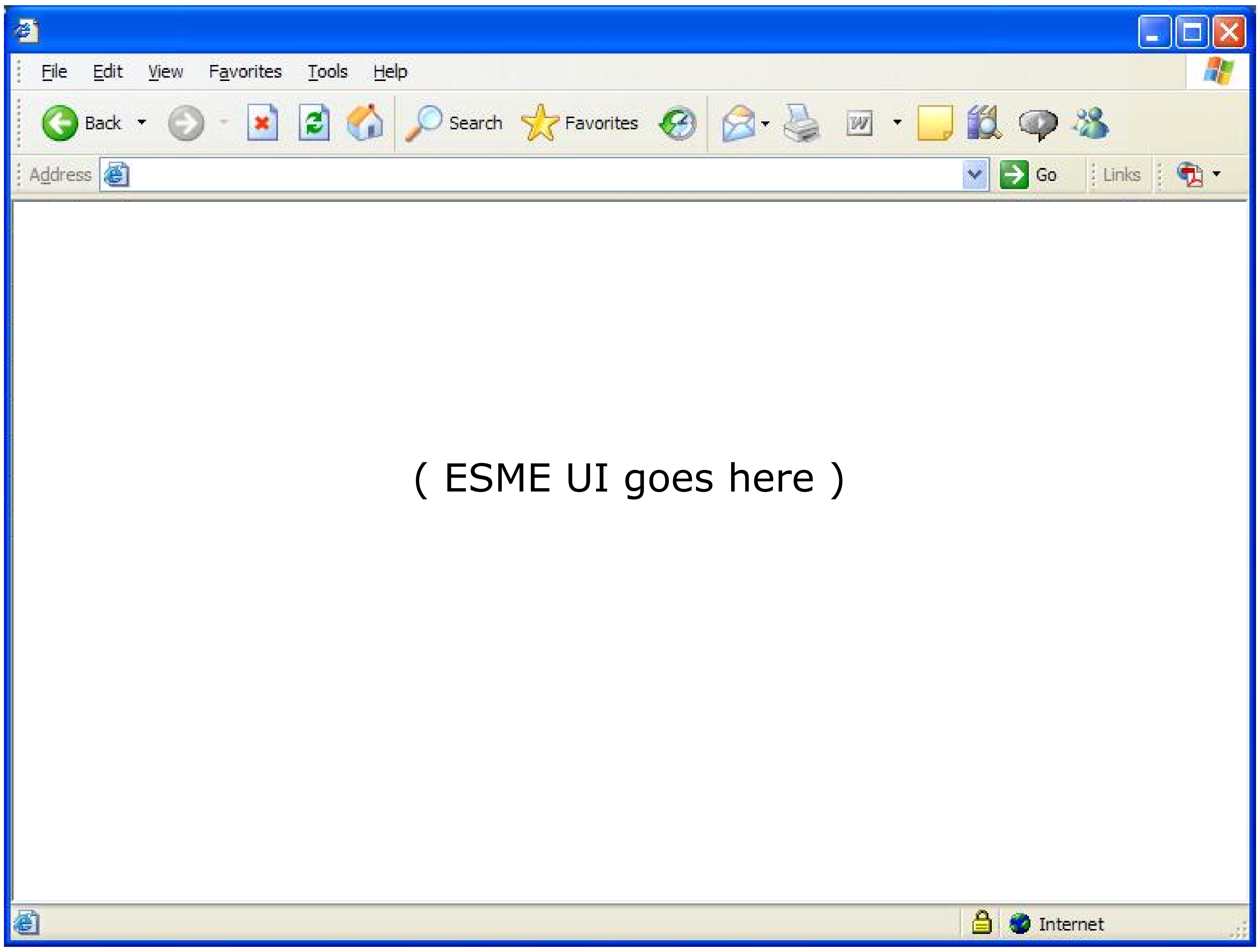
In expanded mode, the ESME window should display the full UI in whatever state the user last left it.

I shall describe all of these ideas in the succeeding sections of this document.

THE WEB BROWSER WINDOW FRAME

In reading through this document please bear in mind that all illustrations show only what appears in the interior of a web browser window.

In most cases it is irrelevant whether a user accesses ESME through Firefox, Internet Explorer, Safari or any other web browser, whether ESME tries to hide the menubar, toolbar, address bar, status bar, etc., and whether the UI is in shrunken or expanded mode. The important question is what should appear inside each web browser window, so to save effort the inside is all I've illustrated.



(ESME UI goes here)

Login Page

THE LOGIN PAGE

All users must be required to log in before gaining access to the ESME service. The login page is where they do this.

Required Elements

The login page should contain all of the following elements:

• The name of the organization. This tells the user the scope of the user base, and also the policy-making body under whose terms he/she may use the service.

• The name or description of the service. This let’s the user identify the nature of the service he/she is accessing, and is one of the key pieces of information needed to choose between the many services and tools he/she may utilize over time.

(These first to can easily be accomplished with a title or header such as “Log into Acme Corp’s ESME messaging system”)

• A login form.

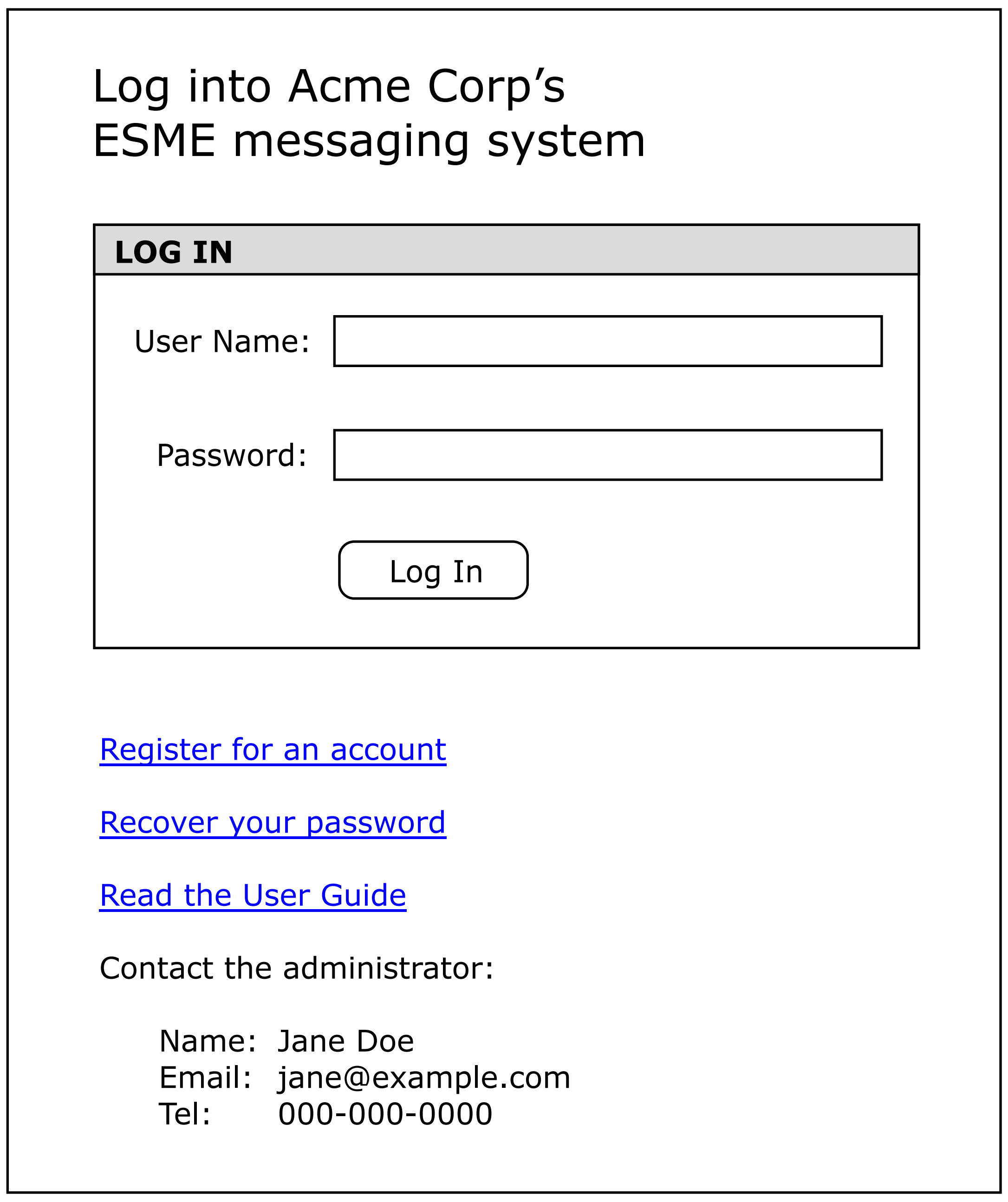
• Instructions for contacting the person(s) designated to provide information and support to the user base.

• Instructions for how to become authorized to use the service (e.g. a “Register for an account” link).

• Instructions for how to recover your password, or re-authorize a lapsed account, or whatever other error conditions may arise.

• Clear and helpful error handling for when a user supplies invalid login information. E.g. “your account has expired, call 000-000-0000 to renew your account”, “we do not recognize your user name”, etc.

An Example Login Page



Log into Acme Corp’s
ESME messaging system

LOG IN

User Name:

Password:

Log In

Register for an account

Recover your password

Read the User Guide

Contact the administrator:

Name: Jane Doe
Email: jane@example.com
Tel: 000-000-0000

Optional Elements

The login page should also be able to display information such as:

• A usage-policy statement (e.g. “Use of this system is restricted to employees with class E clearance or above.”).

• Instructions for accessing the user guide.

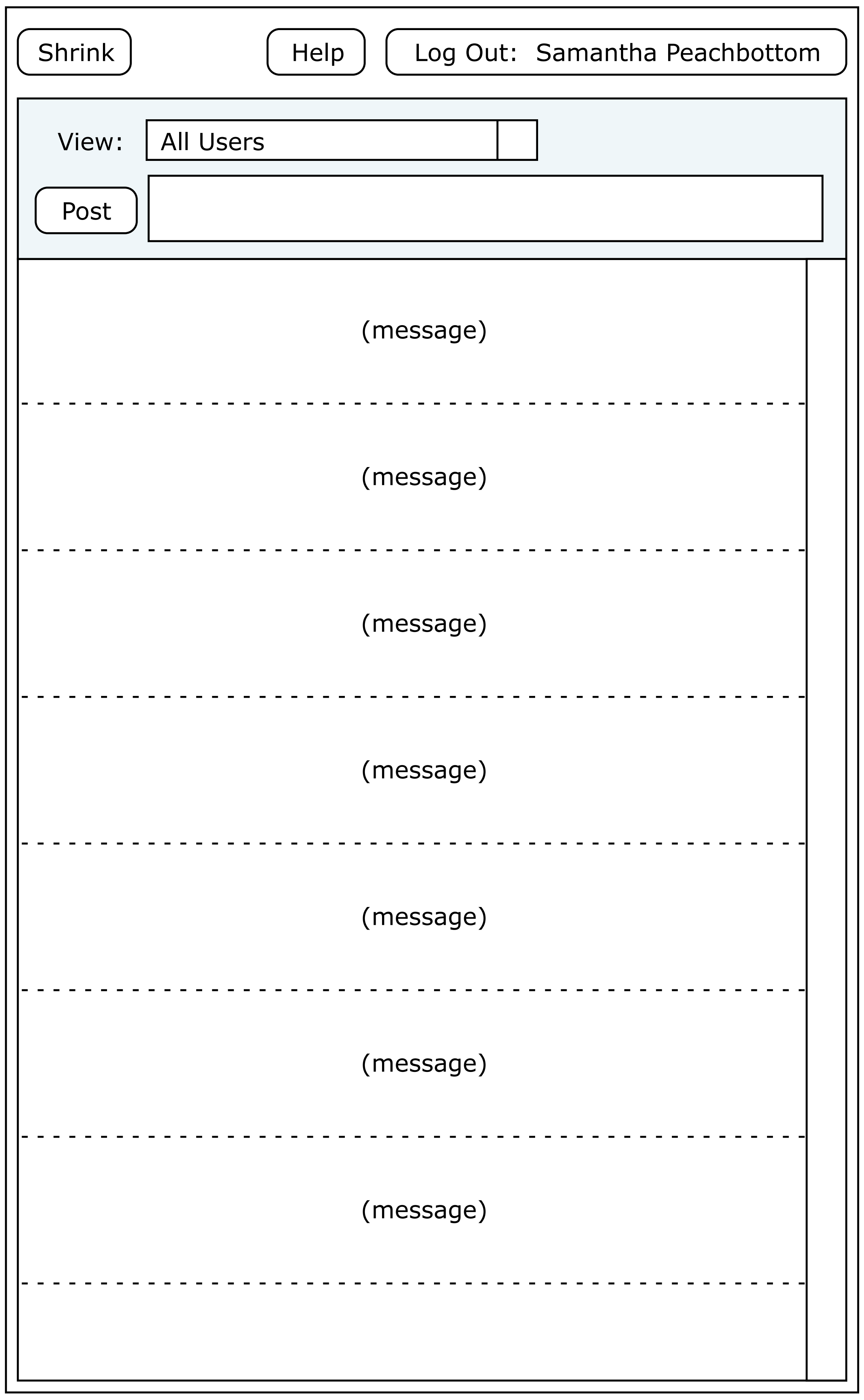
Spelling Recommendations

I strongly recommend that we use the two-word verb forms “log in” or “log into” when labeling an action (e.g. the button that logs you in should be labeled “Log In”), and that we use the one-word noun form “login” when labeling a place or a thing (e.g. the “Login” page is where you go to log into the system).

The Core End-User User Interface

THE CORE END-USER UI

The core end-user UI is designed to present the minimum UI necessary for using ESME once you have configured your preferences, etc. Here is an illustration showing it's elements and how they'd be laid out in the window:



The "Shrink" Button

Clicking this button shall cause the window to shrink down to it's shrunken state. (In that state the window displays a corresponding "Expand" button, so that the user can easily toggle between the two states.)

The "Help" Button

Clicking this button opens the user guide in a new window.

The "Log Out..." Button

This button does double-duty: It displays the name of the currently logged-in user, and clicking it logs that user out.

I suggest that upon logout we do **not** display a confirmation dialog. Rather, we should simply replace the ESME interaction UI with the login page, displayed in the same window.

The "View" Menu

This is a drop-down menu that lists all the message streams that are accessible to the user. I have not designed a structure for this menu, but you can imagine that any streams that appear in the left sidebar (described in a later section) will be selectable from this menu.

The "Post" Button

Clicking this button shall post any message in the text box to its right. It should be disabled when the message box is empty, and enabled when there is any text in the message box. After sending a message the message box should be empty and this button should be disabled.

The Message Box

The unlabeled text box to the right of the Post button is where you type the message you wish to send. You may include in this text string any coded commands and codes that ESME recognizes (e.g. "@" and "#").

The Message Stream

The main body of the window contains the list of messages that meet the specifications of the selected stream.

The Scroll Bar

At the right edge of the list of messages is a scrollbar.

MISCELLANEOUS CONSIDERATIONS

Window Title

What should be the title in the titlebar of the web browser's window in which the ESME UI appears?

I think this should be static text that is configurable by the administrator. The default should probably be something like "ESME Messaging System".

Can You Have Multiple ESME Windows Open?

I suggest that the answer should be yes. Each should be essentially equivalent.

What if You Close All Your ESME Windows?

If this is a condition that we can detect, then we should make this cause the user to be automatically (and invisibly) logged out of ESME.

If desired we can have a preference set by the system administrator that gives users a brief grace period in which they can re-open the window without having to log back in, but very security conscious organizations would want this to be set to zero seconds.

Window Resizing

On the next page we show some examples of the ESME window resized to various aspect ratios.

Notice how the layout responds:

- The message list resizes in width to take up the full width of the available space.

- The height of each message varies as needed to make sure the entire message is visible within the available width.

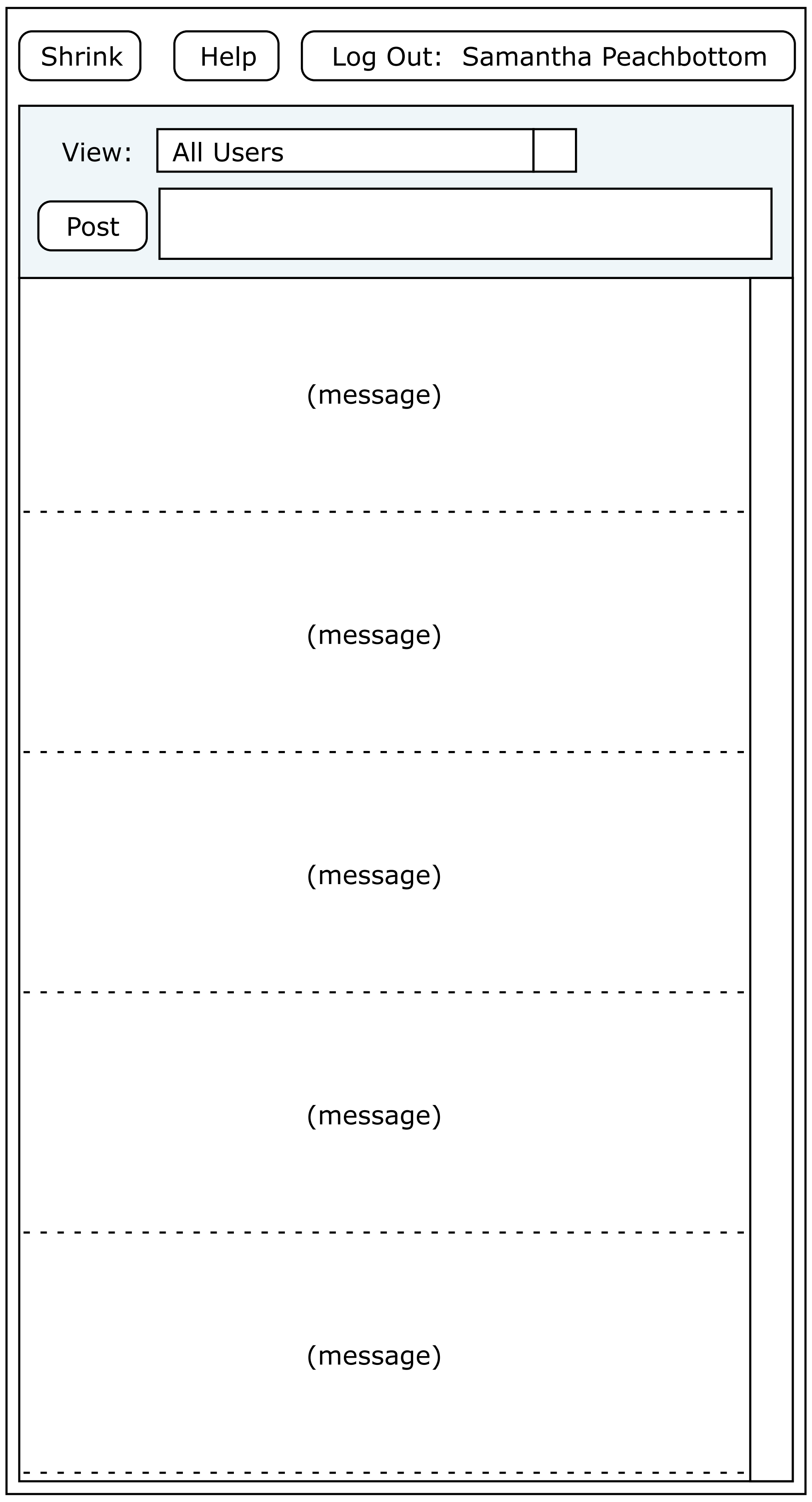
- The width of the message box grows or shrinks to take advantage of the available width.

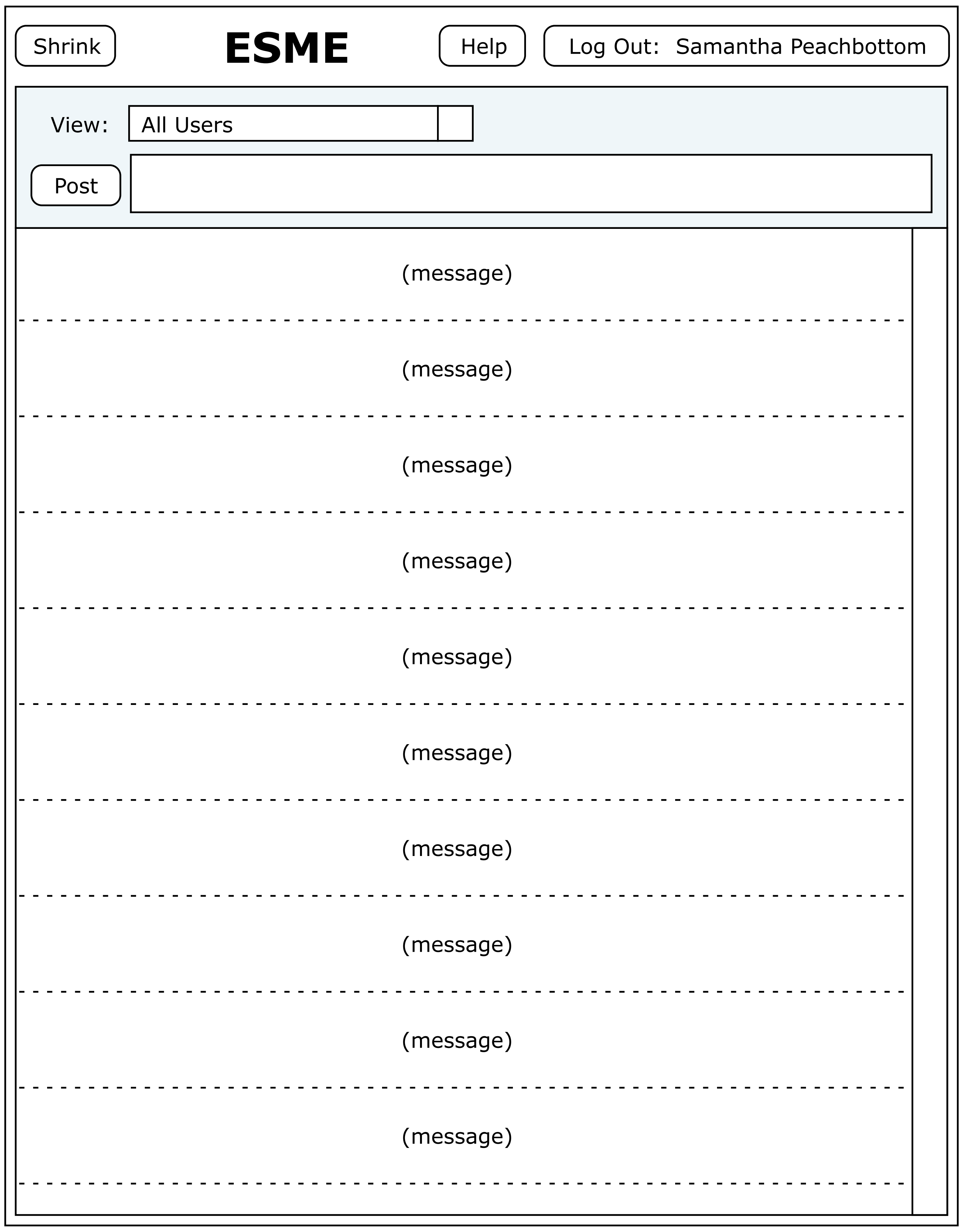
- At the very top, the "Shrink" button remains left-aligned, while the "Help" and "Log Out" buttons are grouped together and right-aligned.

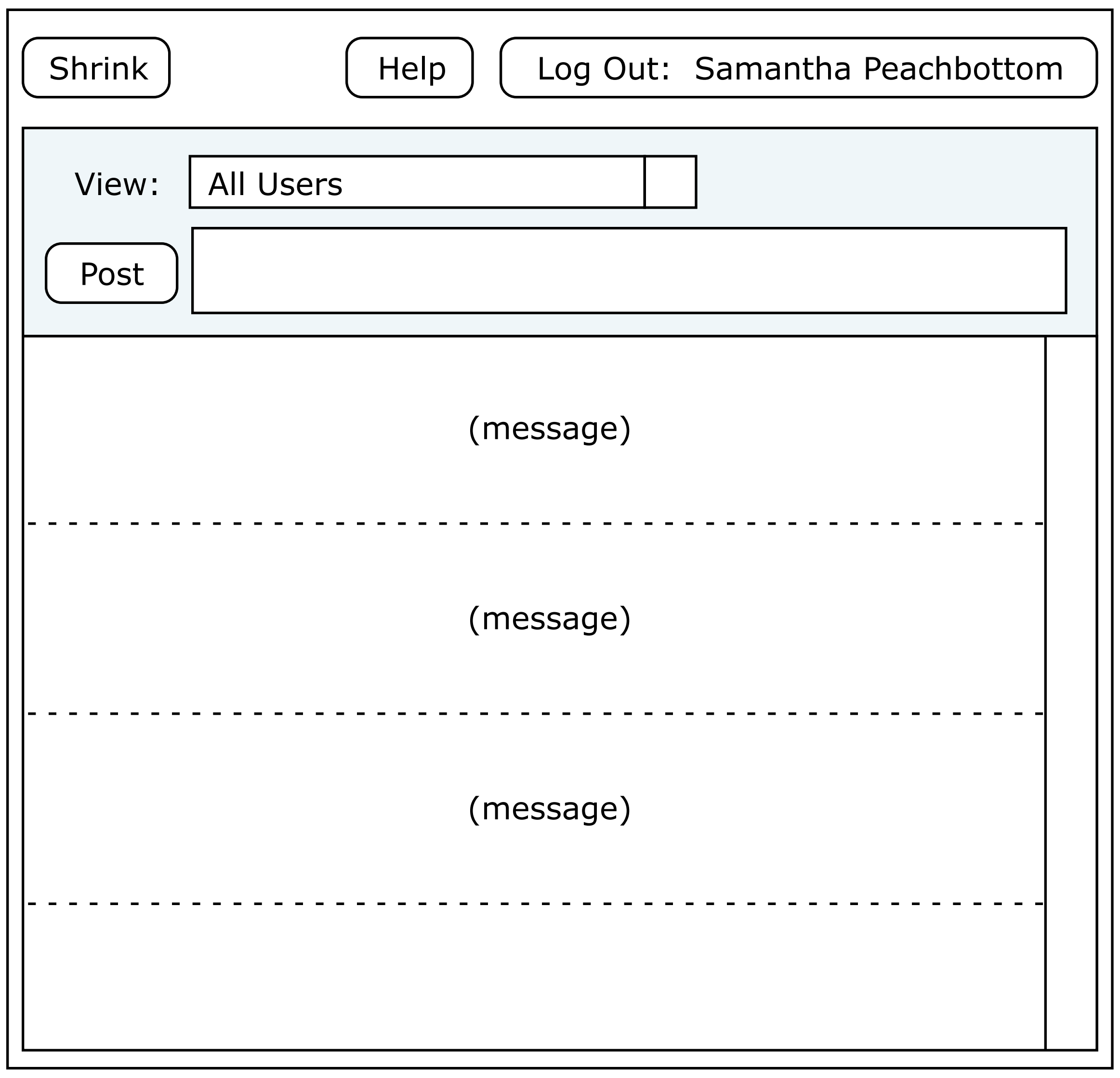
- When the window is wide enough to do so without clipping or overlap, we can display the ESME logo.

The Core End-User User Interface

Here are examples showing how the the core end-user UI adjusts its layout to respond to various window sizes.



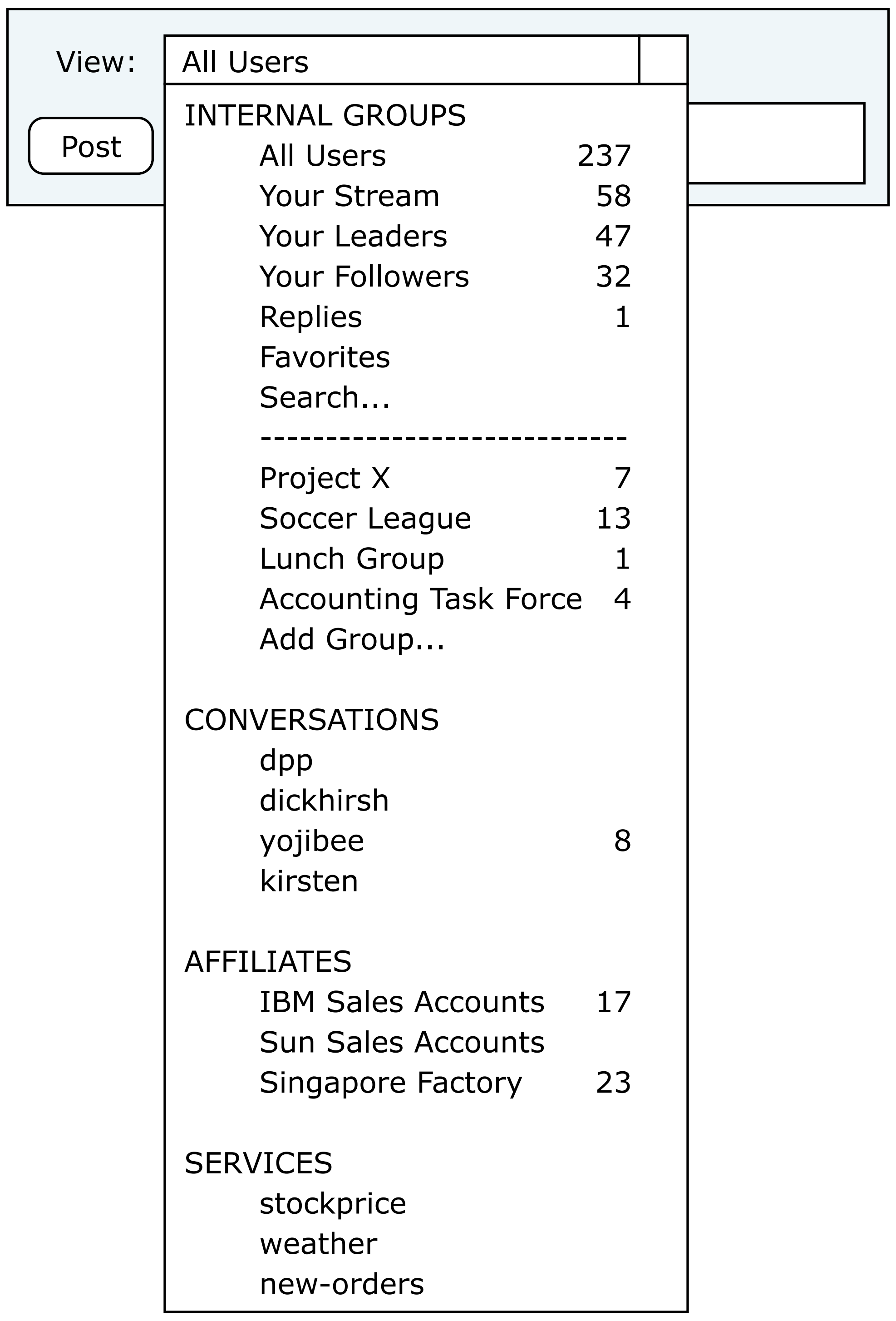




The “View” Menu

THE “VIEW” MENU

To a first approximation, the “View” menu should probably list all the streams that have been predefined for and by the user. The list should probably be organized in the same way as the left sidebar described later. And each item should probably have new-activity indicators or counters as described later for the left sidebar.



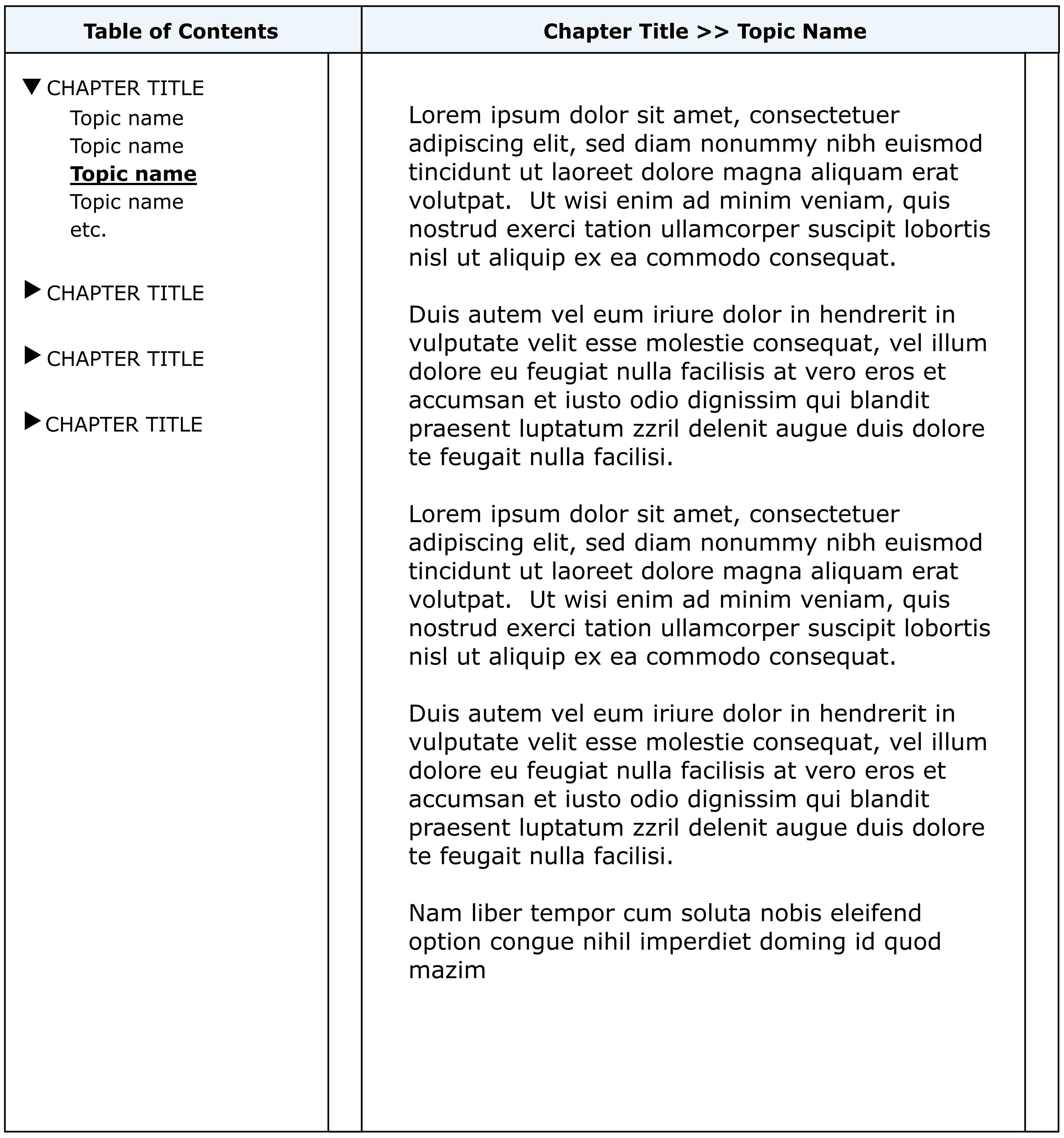
This menu allows users to:

- Quickly scan their streams to determine where there is new activity.
- Easily shift focus back and forth between message streams.

ONLINE USER GUIDE

As mentioned earlier, the core end-user UI has a “Help” button near the upper-right corner. Clicking this button should open a new web browser window, containing the online user guide.

I suggest that the user guide UI be structured basically as shown in the illustration below.



Lorem ipsum dolor sit amet, consectetuer adipiscing elit, sed diam nonummy nibh euismod tincidunt ut laoreet dolore magna aliquam erat volutpat. Ut wisi enim ad minim veniam, quis nostrud exerci tation ullamcorper suscipit lobortis nisl ut aliquip ex ea commodo consequat.

Duis autem vel eum iriure dolor in hendrerit in vulputate velit esse molestie consequat, vel illum dolore eu feugiat nulla facilisis at vero eros et accumsan et iusto odio dignissim qui blandit praesent luptatum zzril delenit augue duis dolore te feugait nulla facilisi.

Lorem ipsum dolor sit amet, consectetuer adipiscing elit, sed diam nonummy nibh euismod tincidunt ut laoreet dolore magna aliquam erat volutpat. Ut wisi enim ad minim veniam, quis nostrud exerci tation ullamcorper suscipit lobortis nisl ut aliquip ex ea commodo consequat.

Duis autem vel eum iriure dolor in hendrerit in vulputate velit esse molestie consequat, vel illum dolore eu feugiat nulla facilisis at vero eros et accumsan et iusto odio dignissim qui blandit praesent luptatum zzril delenit augue duis dolore te feugait nulla facilisi.

Nam liber tempor cum soluta nobis eleifend option congue nihil imperdiet doming id quod mazim

THE SHRUNKEN WINDOW

I think its very important that users be able to shrink the ESME end-user interaction window into a small size, which can be expanded back into a full-size window at the user's discretion.

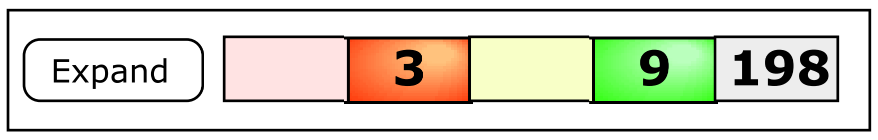
This allows the user to get ESME out of the way when he/she needs to focus on other tasks, then easily bring it back when his/her attention switches back to micro-messaging.

Obviously in it's shrunken size it will be too small to support a micro-messaging UI, and indeed why it is shrunken is so that the user can get the micro-messaging UI out of the way without having to close or quit the application.

So in its shrunken state its function should switch to one of informing or alerting the user to when there is new activity in which he/she may be interested.

My concept is that there should be red, orange, amber, green indicators that light up when there is new activity in a corresponding range of urgency. These indicators can also contain numeric readouts to show how many new messages there are in each category.

We can also have a grey indicator that never lights up but that does contain a numerical readout of how many non-urgent messages have been posted since the user last viewed the relevant message streams.



I have not yet thought about what kind of mechanism should be used to drive each of these indicators, but I'm sure when I have time to think about it I can come up with something meaningful.

The result should be that the user can from time to time glance at the indicators, and see if it seems urgent enough to force a shift in attention from other tasks to reading messages.

The Left Sidebar

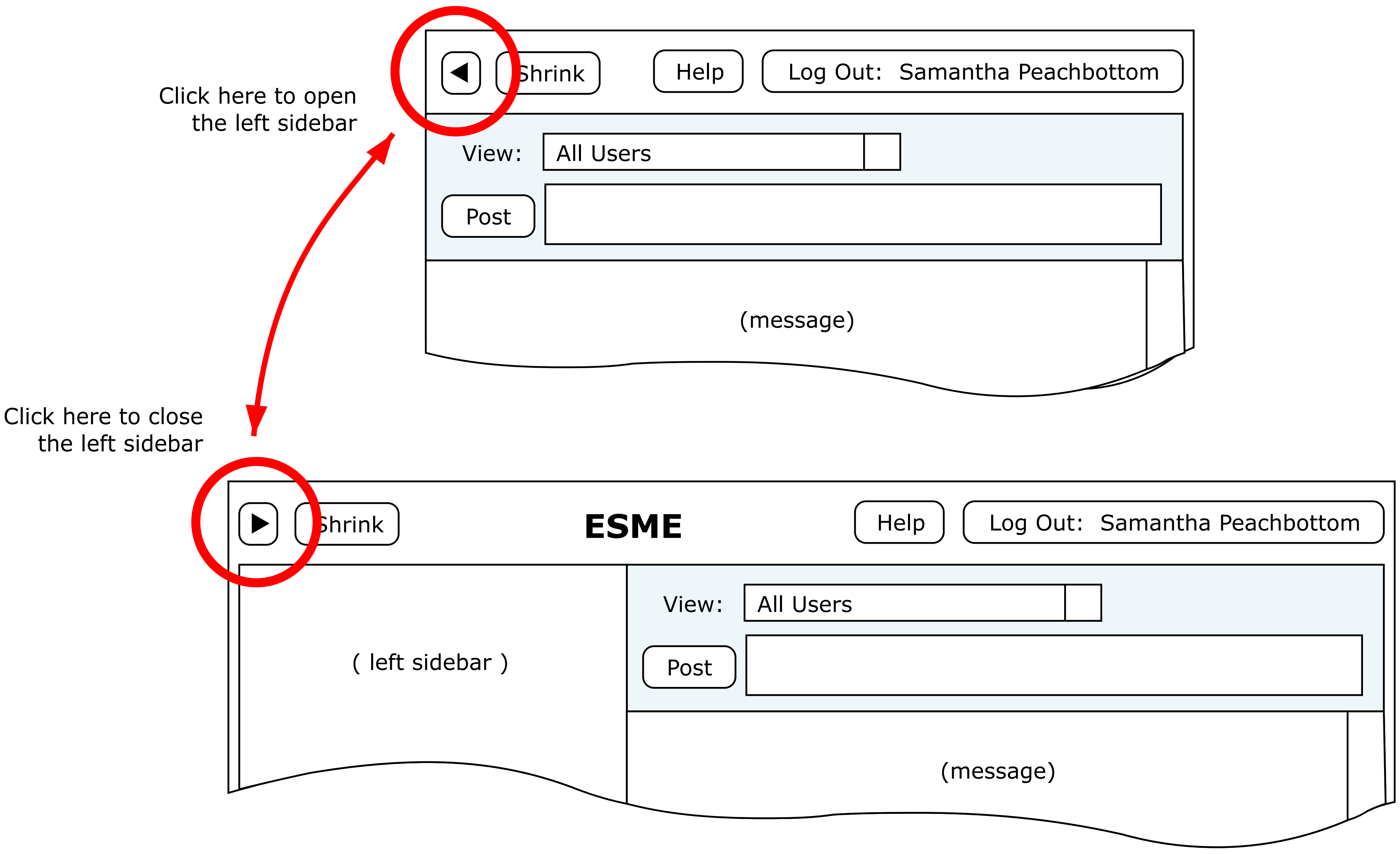
Left Sidebar

LEFT SIDEBAR

Once a user has set up his/her preferences, the core UI should be adequate for many users, much of the time.

But where do you set up your preferences (and what all does "preferences" include)?

I suggest that in the top button bar we add a small button to the left of the "Shrink" button, and that clicking this button will toggle the window between two states: wider, with a left sidebar displayed next to the message list, vs narrower, with just the core UI showing.



Click here to open the left sidebar

Click here to close the left sidebar

Window Repositioning On Opening Sidebar

If technically feasible, I suggest the following:

If there is enough room onscreen to the left, simply expand the window to the left to show the sidebar, leaving the upper-right corner unmoved.

If there is not enough room onscreen to the left, then open the sidebar and shift the window right to insure that the left edge of the window is fully onscreen, leaving the width of the message list unchanged.

If there is not enough screen width to both open the sidebar and keep the width of the message list unchanged, then shrink the width of the message list as much as needed so that when the sidebar is open the entire window will will be onscreen horizontally.

In no case change the vertical position or height of the window.

Window Repositioning On Closing Sidebar

Simply shrink the width of the window to hide the sidebar, leaving the top-right corner of the window unmoved.

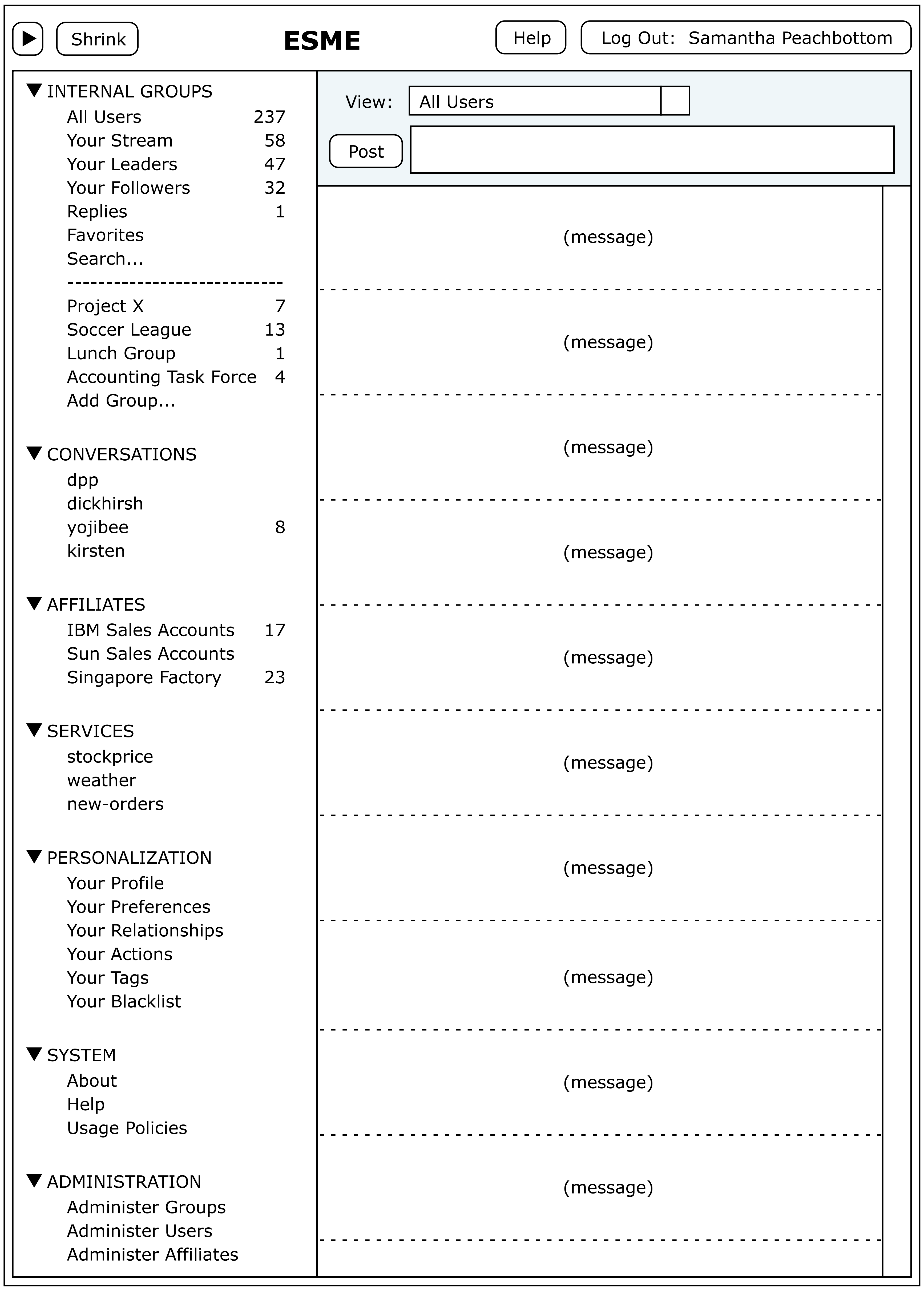
Do not change the vertical position or height of the window.

Left Sidebar Example Contents

LEFT SIDEBAR EXAMPLE CONTENTS

This is an example of what the left sidebar would look like and what might be in it.

In the following pages I describe what is intended by each item.



▼ INTERNAL GROUPS

All Users	237
Your Stream	58
Your Leaders	47
Your Followers	32
Replies	1
Favorites	
Search...	

Project X	7
Soccer League	13
Lunch Group	1
Accounting Task Force	4
Add Group...	

▼ CONVERSATIONS

dpp	
dickhirsh	
yojibee	8
kirsten	

▼ AFFILIATES

IBM Sales Accounts	17
Sun Sales Accounts	
Singapore Factory	23

▼ SERVICES
stockprice
weather
new-orders

▼ PERSONALIZATION
Your Profile
Your Preferences
Your Relationships
Your Actions
Your Tags
Your Blacklist

▼ SYSTEM
About
Help
Usage Policies

▼ ADMINISTRATION
Administer Groups
Administer Users
Administer Affiliates

Left Sidebar Function and Contents

LEFT SIDEBAR FUNCTION

The left sidebar presents a “table of contents” of things that can be displayed in the main body of the window (where the core end-user UI is displayed).

This table of contents should be divided into sections by function, and the sections should be collapsible / expandable.

There must always be one (and only one) item chosen in this list. When an item is chosen (usually by clicking on it) it will be visually highlighted to show that it is selected, and its corresponding UI will be displayed in the main body of the window.

LEFT SIDEBAR SECTIONS

Here are some of the sections I envision at this point.

Message Streams

Most of the choices are message streams (both built-in and user-defined). Clicking any of these choices causes the chosen message stream to be displayed in the message list and the “View” menu to be updated to correspond. Similarly, making a choice using the “View” menu will cause the corresponding item in the left sidebar to be highlighted (if necessary, the corresponding section should be expanded to make the selected item visible).

Groups

Among the items in the left sidebar should both be the list of groups of which you are a member, and an item invoking the UI for managing your groups.

Search

Other items in the left sidebar produce the UI for searching the message pool.

Personalization

Other items in the left sidebar invoke the UIs for personalizing the user’s profile, setting personal preferences, managing leaders and followers, etc.

System Info

The left sidebar can also be used as the place from which to invoke system-level UIs, such as the About info, Help info, and the organization’s Usage Policies or Terms of Use.

Administration

Other items in the left sidebar can be provided for those users who are “end users” but who have special privileges: for example to manage groups, or get management reports, etc.

NEW MESSAGE INDICATORS

You may have noticed the numbers to the right of several of the stream selectors. The idea is that we could show the number of new message that have been posted within each stream since you last viewed it.

I suggest that the new message count be reduced to zero whenever you view a given stream, and that we don’t try to be figure out whether you’ve actually read them all.

If it’s too hard to display actual numbers, how about just a small icon indicating that there are new messages?

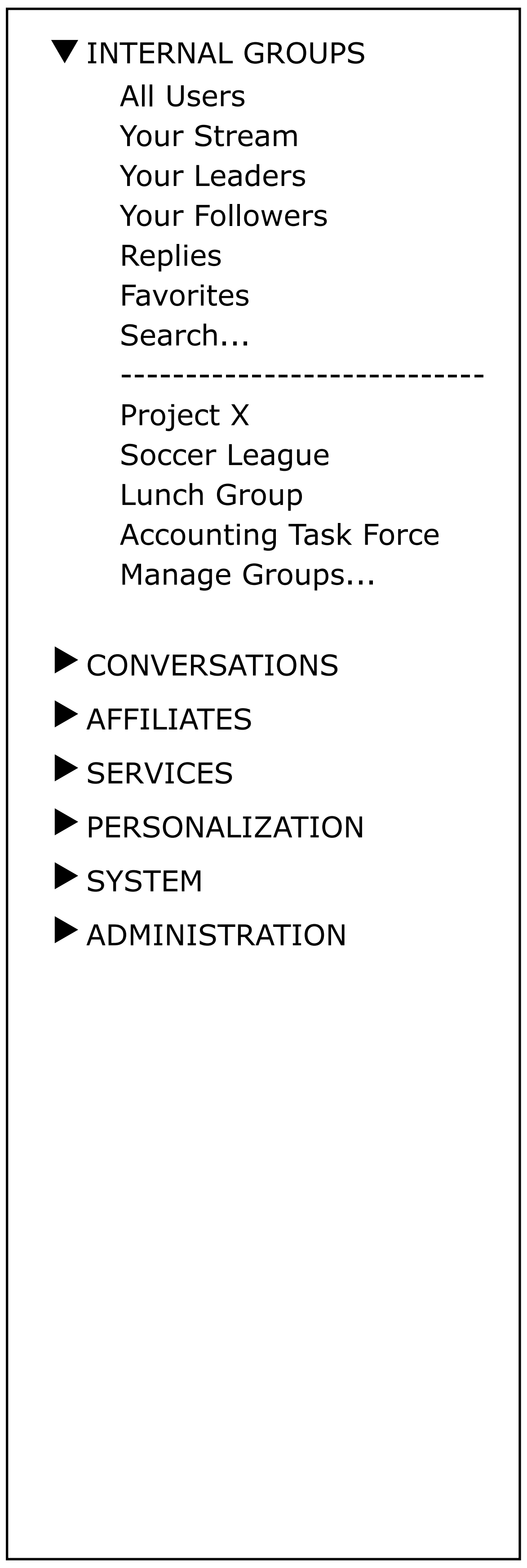
The “Internal Groups” Section

THE “INTERNAL GROUPS” SECTION

If we support affiliates (which are groups we communicate with that are outside our organization) then it’s critically important to know when we are free to write things that are confidential to the organization, versus things that might be confidential to each affiliate.

Therefore I have placed all the “inside the organization” message streams in an “Internal Groups” section (“Internal Streams” would also be a good name for this section).

I’ve also placed this section at the top since it’s the one that every ESME installation will have, and that every user will almost certainly use most.



Built-In Streams

The first part of this section lists the built-in streams that the user can’t change.

Search

I’ve put the search item here, but I’m not yet sure it’s the right place for it.

Groups

The last part of this section is where I envision listing all the groups of which you are a member.

I’ve also placed a “Manage Groups” item, intended to lead to the UI for joining and leaving groups.

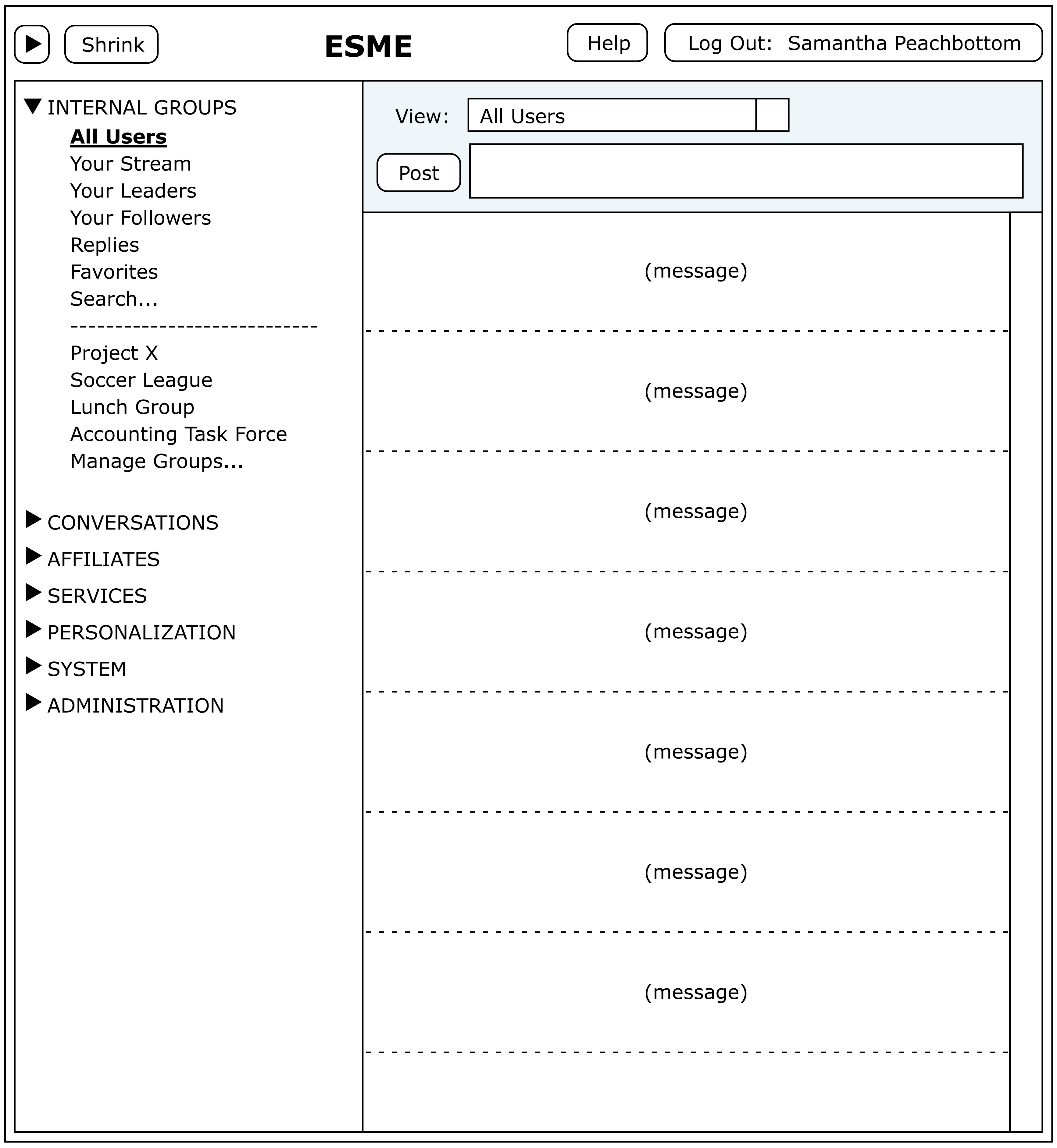
Stream — “All Users”

THE “ALL USERS” STREAM

This shows the latest posts out of the entire pool of internal posts. This is the stream that is most akin to Twitter’s public timestream.

“All Internal Users” might also be a good name for this stream.

This stream does not include any messages to or from affiliates, each of which has its own private pool of messages.



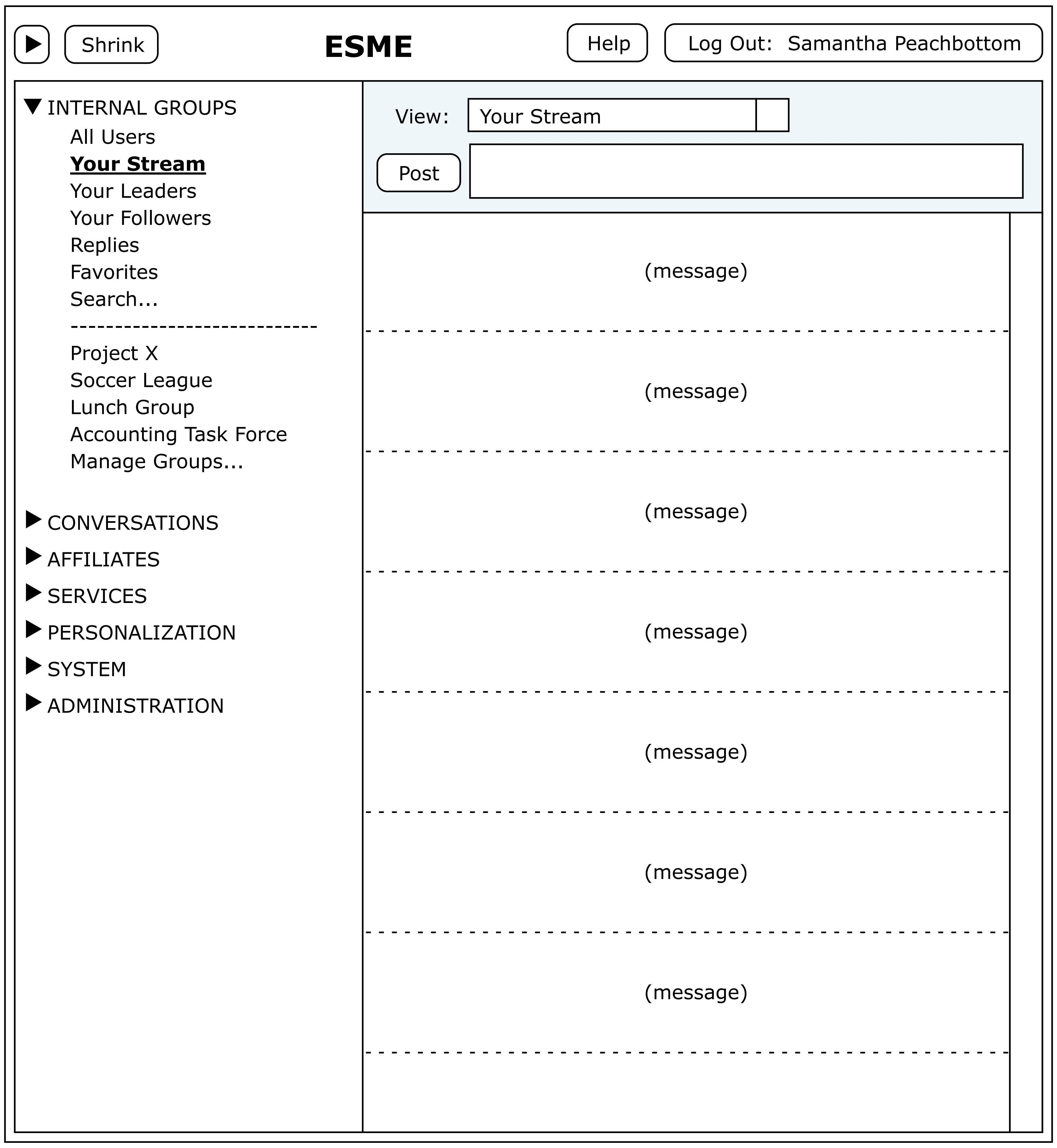
View: All Users

Stream — “Your Stream”

THE “YOUR STREAM” MESSAGE STREAM

This is the aggregate of all public messages sent by you and your leaders.

This is the stream most akin to Twitter’s “Home” display.



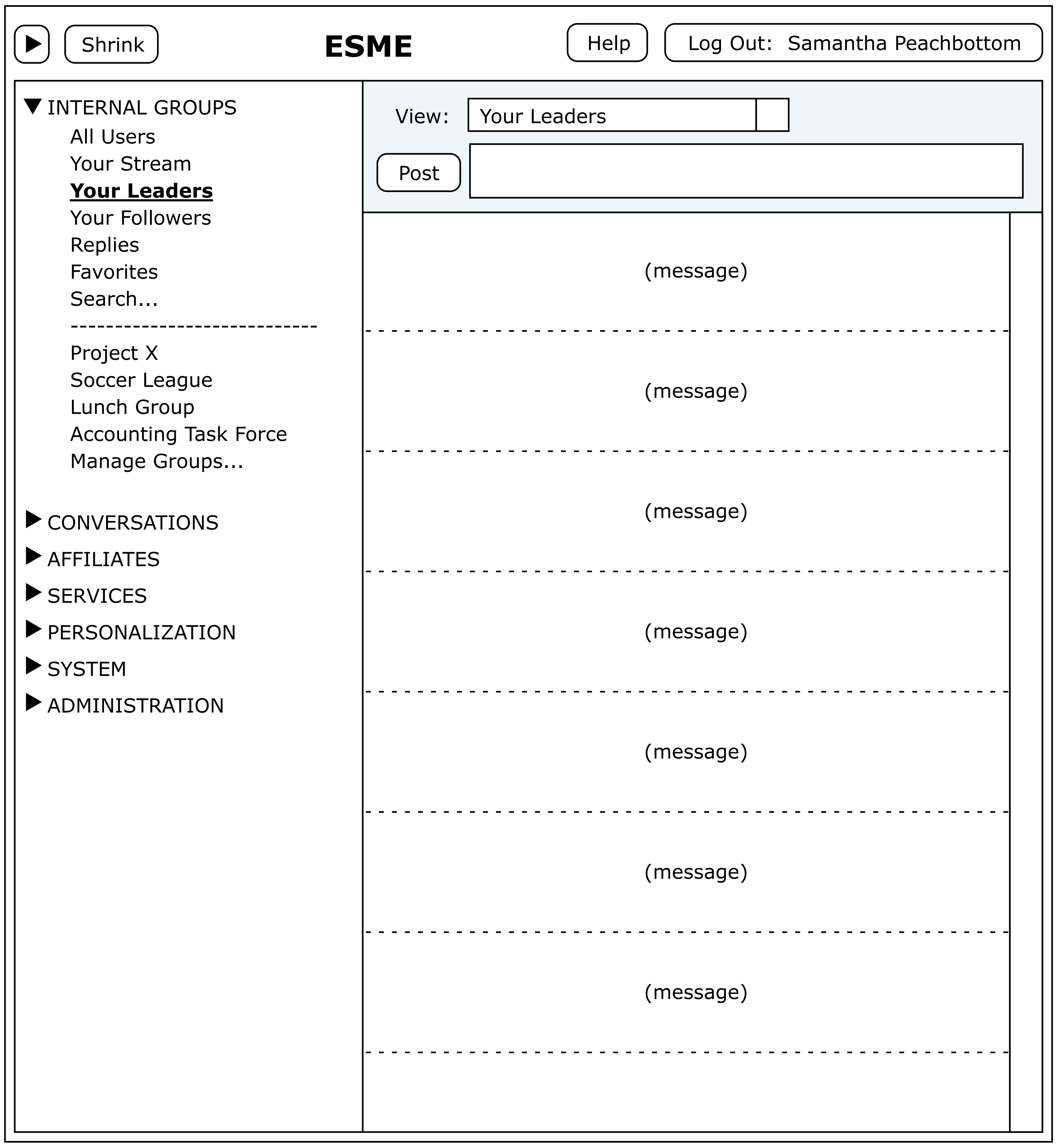
View: Your Stream

Stream — “Your Leaders”

THE “YOUR LEADERS” MESSAGE STREAM

This is the aggregate of all public messages sent by your leaders.

I’m not sure if this is a good or needed idea. But it shows it’s easy to extend the table of contents model whenever we think of something new.



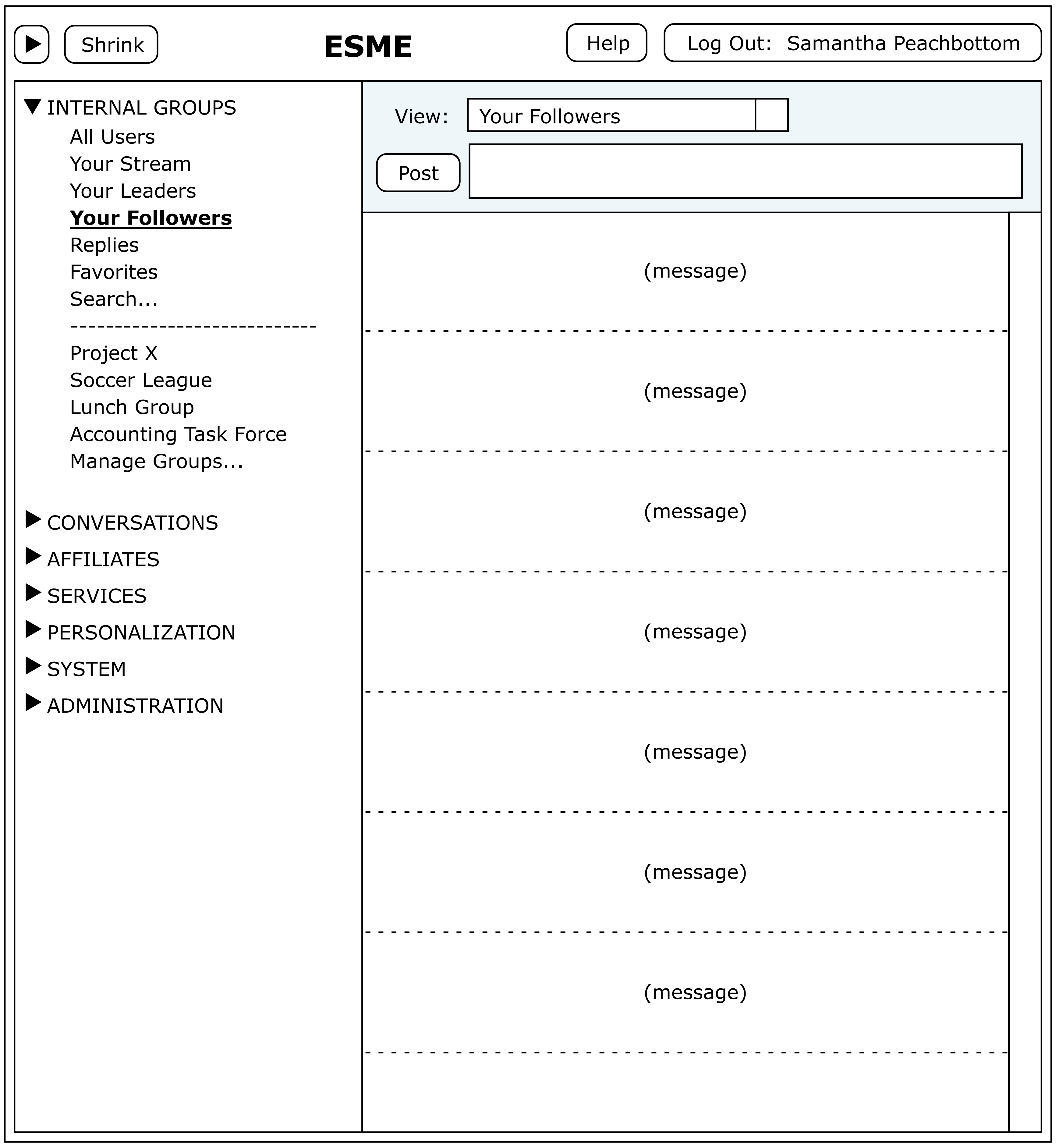
View: Your Leaders

Stream — “Your Followers”

THE “YOUR FOLLOWERS” MESSAGE STREAM

This is the aggregate of all public messages sent by your followers.

I’m not sure if this is a good or needed idea. But it shows it’s easy to extend the table of contents model whenever we think of something new.



View: Your Followers

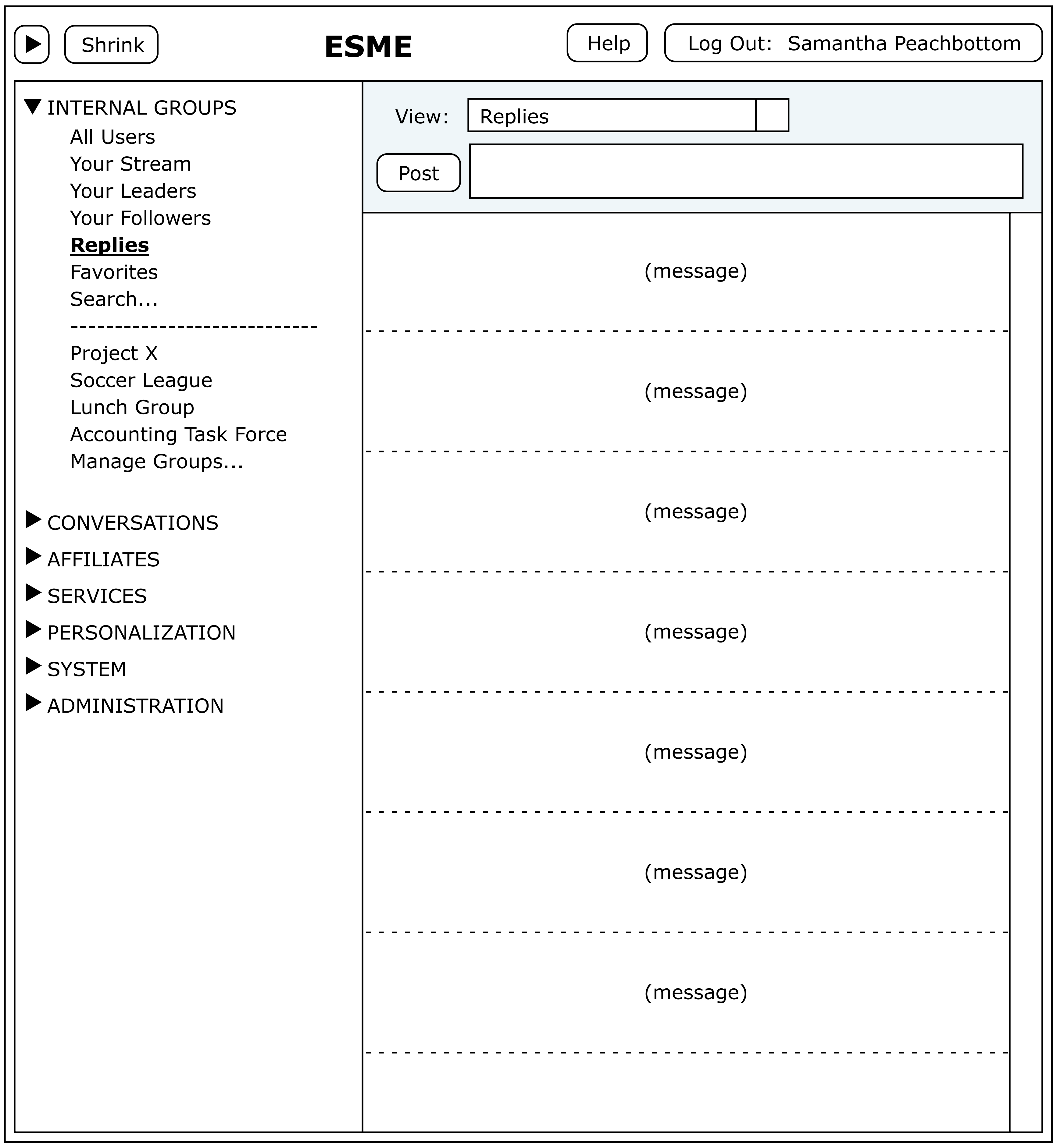
Stream — “Replies”

THE “REPLIES” STREAM

This is a list of all the replies that users have posted to your messages.

Like Twitter, this could simply be a chronological listing. But I don’t like that this puts all the replies out of context, and leaves it up to you to try to associate replies with your original messages.

A better approach might be to say that each time someone replies to one of your messages it starts a “conversation”, which then puts a new entry in the “Conversations” section. This would be very much like how Apple’s iPhone turns every exchange of SMS text messages into threaded, IM-like transcripts which makes it very easy to follow.



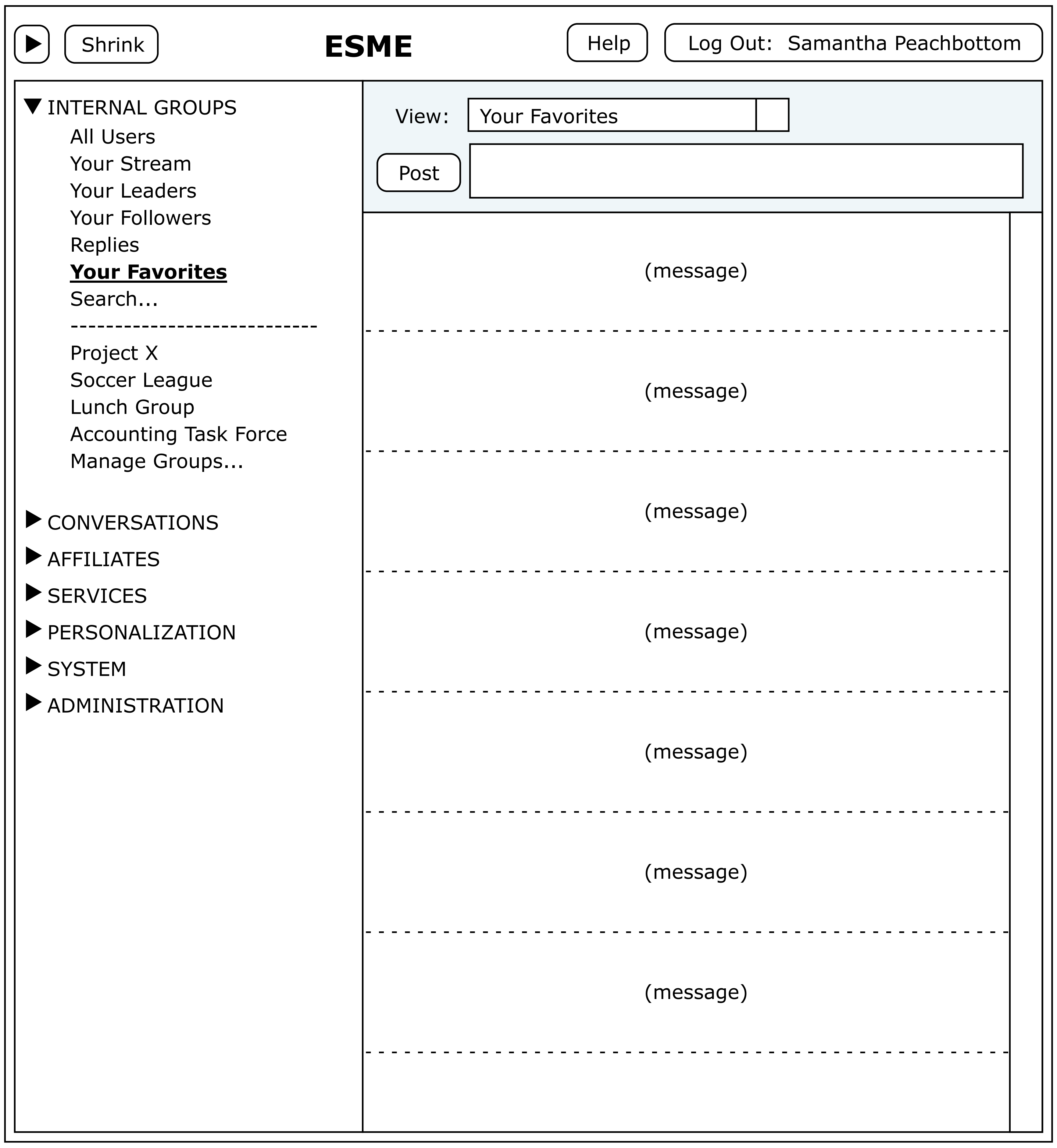
View: Replies

Stream — “Your Favorites”

THE “YOUR FAVORITES” STREAM

Just like Twitter, any message you want to rembember can be added to your list of favorites.

‘Tho there may be some issues with trying to mix internal and affiliate favorites, or public with private favorites... Hmm.



View: Your Favorites

(message)

(message)

(message)

(message)

(message)

(message)

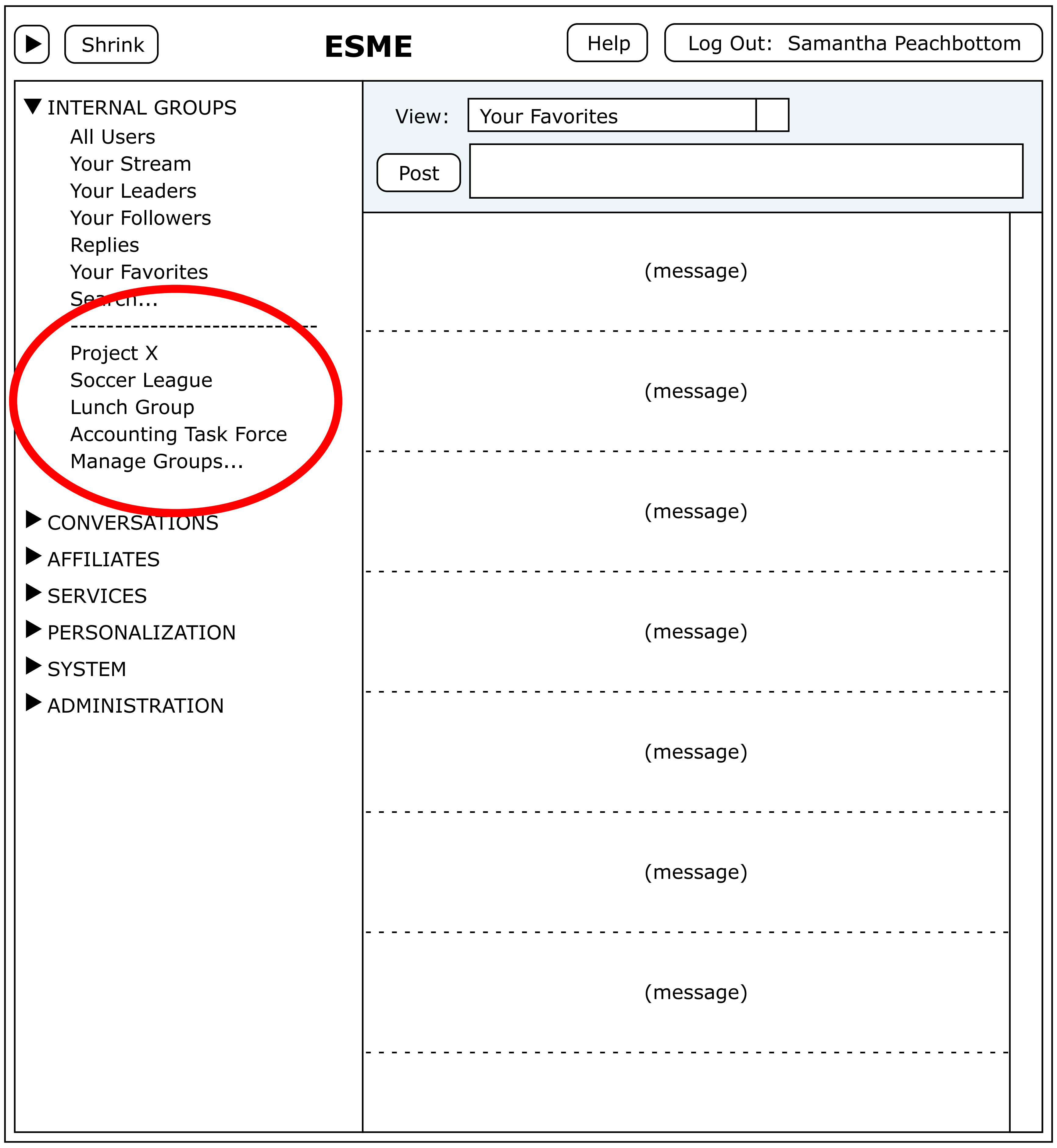
(message)

GROUP STREAMS

You may be a member of multiple groups, each of which has its own message stream.

We can add a list of these groups below the built-in streams in the "Internal Groups" section. These would behave as stream-selectors just like the fixed, built-in items above.

We will also need a way for users to manage what groups they are members of, and I haven't yet thought about how best to do that.



View: Your Favorites

Conversations

CONVERSATIONS

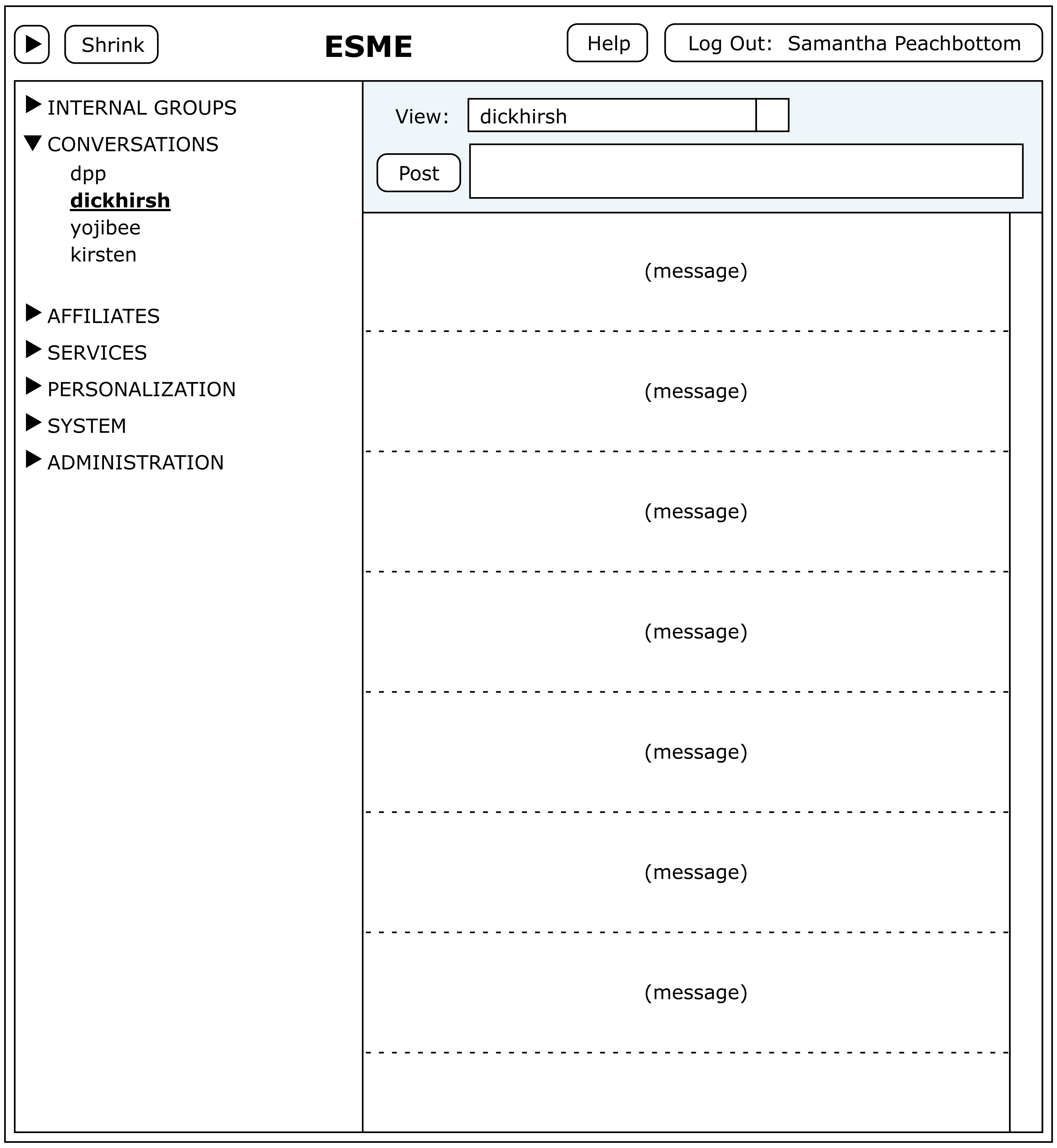
The idea here is that any two users might engage in a private conversation, and this section would list all the individuals with which you are engaged in such message exchanges.

On the surface, this is like the way the iPhone handles SMS text messaging: If you send a text message to (or receive a text message from) it adds their name to a list, and if you select a name it displays a chronological listing of all the messages you two have interchanged, and lets you send a new message to the other party.

In ESME the same thing happens whenever you send or receive a private message (a “direct” message in Twitter parlance).

We should provide a way to delete from your list any of these private message streams (these conversations) that you no longer need to keep track of.

Whether or not we allow more than two people to be members of a private conversation is something we could discuss.



View: dickhirsh

AFFILIATES

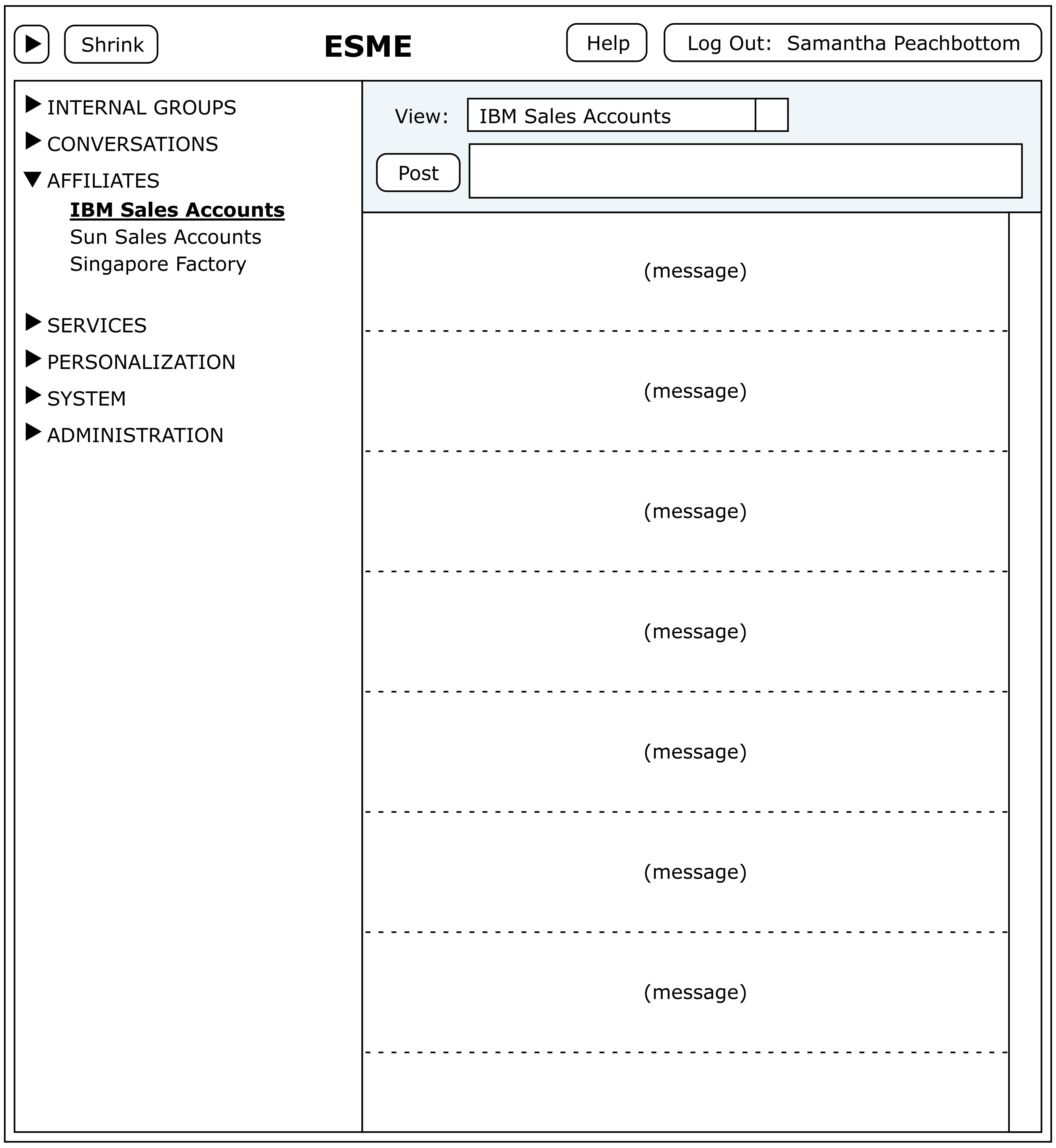
The idea here is that the organization that is running ESME might well want to use it to enhance communications with people outside the organization. These might be sales reps, vendors, consultants, etc.

In these cases a specific set of users at an outside organization, which we will refer to as an affiliate, will be allows to do messaging with a specifi set of users within the internal organization.

It will almost inevitably be true that the messages involving an affiliate will be private and confidential to that affiliate. So we must make each pool of messages to/from a given affilate separate from all other pools of messages, and from the organization's "all users" pool.

It should be a user-administration function to set up affiliate pools, affiliate user accounts, and to specify who within the organization is authorized to communicate with each affiliate, and what profile information is available to those inside and outside the organization.

Each internal user should only see listed in his/her Affiliate section the names of those affiliates with which he/she is authorized to communicate. If a user has no affiliate authorizations, then this section should not appear in his/her UI.



Services

SERVICES

I understand that we wish to make it possible for users to interact with servers. I anticipate that these interactions may take various forms, such as:

- “Info Bots” that return data (say stock prices) in response to requests (say ticker symbols).

- Interrogation, command and control of business operations systems, (e.g. “are the servers up?” or “reboot server 5").

- Calculators, etc.

I suggest we refer generically to these as “services” ?

Different Ways to Handle Services

We could have a section in the table of contents that lists only services.

Each time you use a new service we can automatically add it to the list in this section. We should also provide a way to manually add and remove services from this list.

In this way we can treat the interaction with each service as a separate stream.

Another approach is to say that you can make a service call from any stream, and that messages to/from the service become part of that stream. This would allow service calls to be intermixed with inter-human messages and become part of the transcript of events and discussions occurring in that stream.

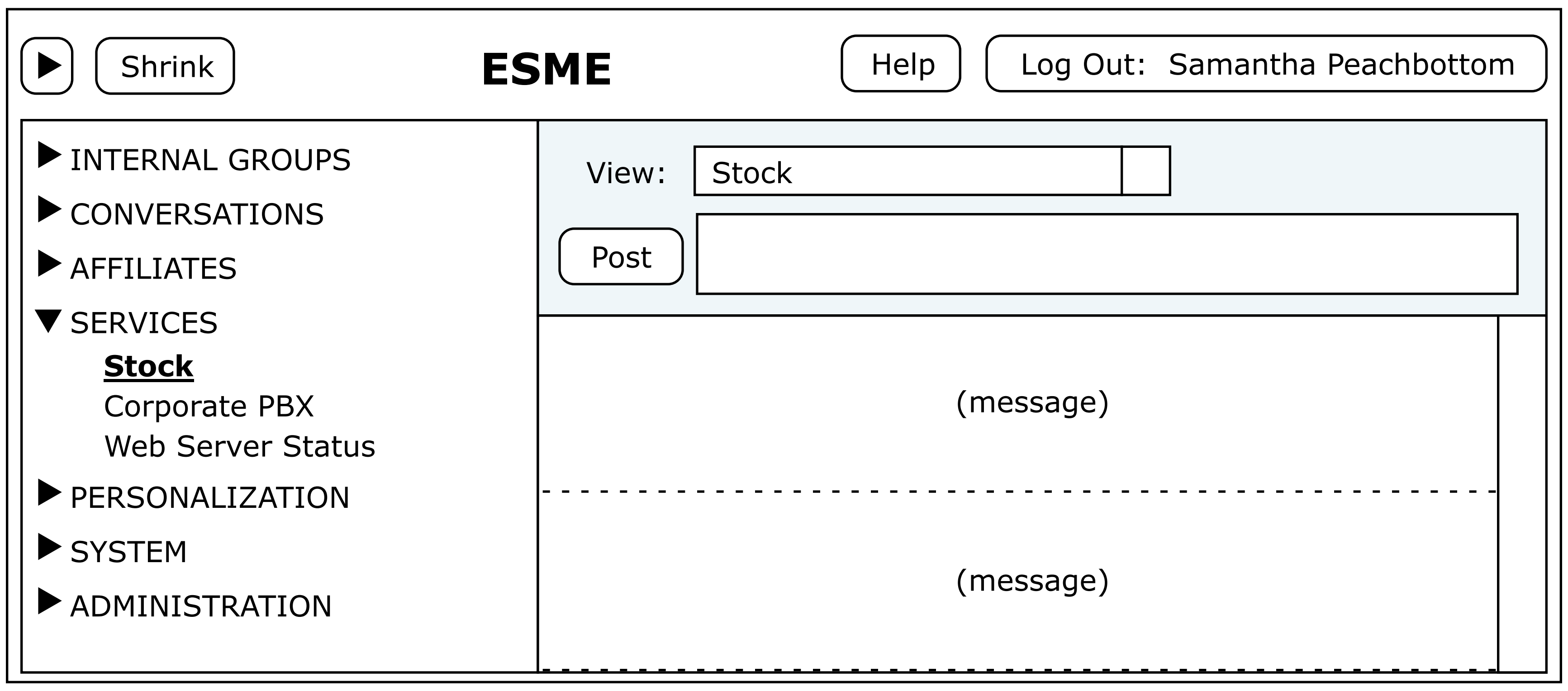
Another alternative is to say when you select a service that instead of the main body of the window showing a message stream you get a custom UI for that service. This UI could have GUI-style parameter entry fields, and user-friendly output displays such as indicators, tables, charts, etc.

Service Call Syntax

Invoking a service by typing a message is like making a subroutine call or issuing a command-line command: in that we must supply the name of the service we wish to invoke and, as needed by each service, supply any needed parameters.

We should define a syntax that can be consistently used by all service calls, so that users will only have to remember one pattern for specifying parameters.

Also, do we need special conventions for service names? For example to make them all start with dollar signs (e.g. “$server uptime”)? Or can they all be in the same namespace as human users?



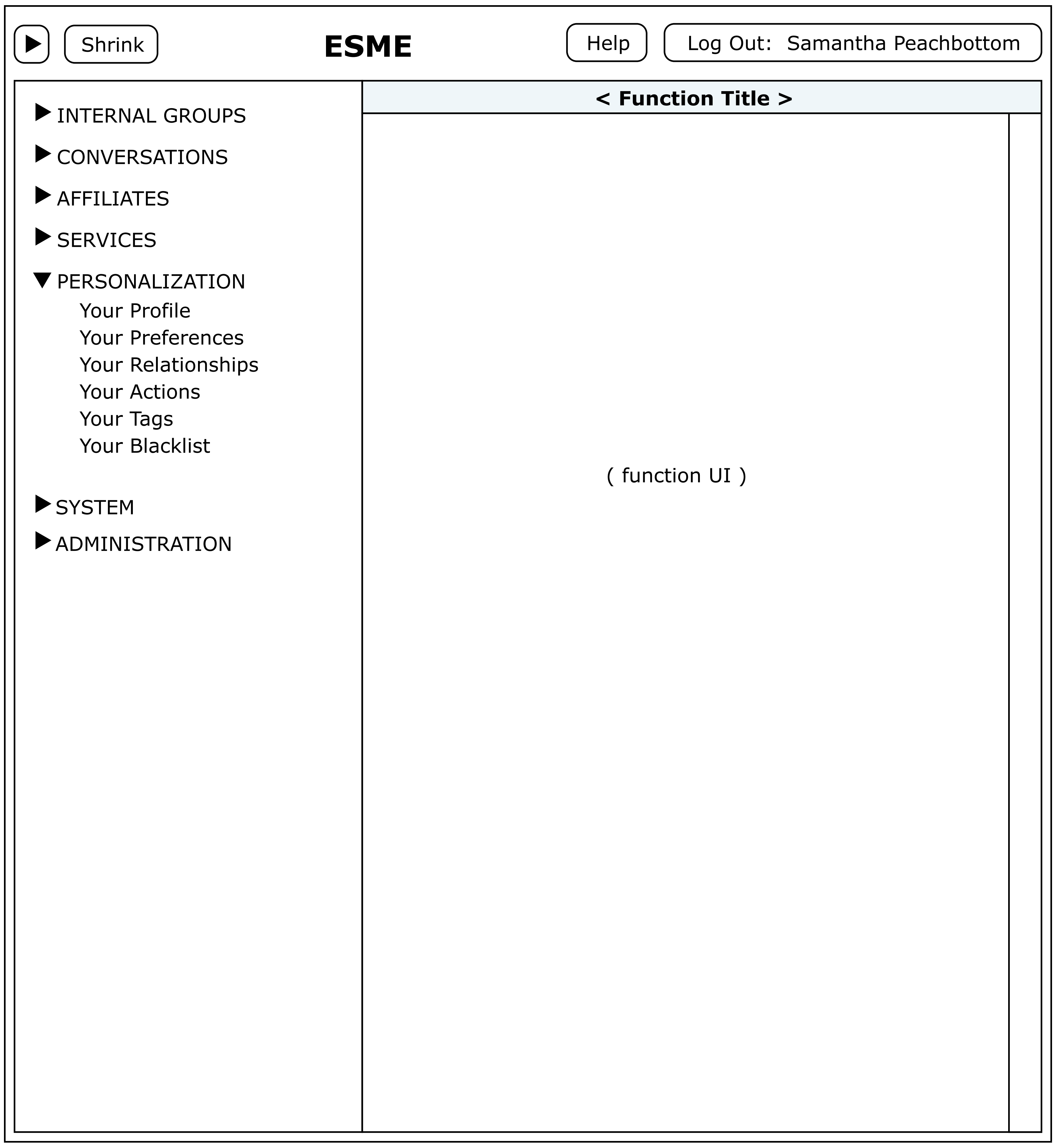
Personalization Functions

PERSONALIZATION FUNCTIONS

Whereas the previous sections have afforded a way for users to organize and access their message streams, this section provides UIs for customizing the ESME experience.

Selecting an item from this section of the left sidebar causes a function-specific UI to fill the main body of the window (instead of a message stream).

We can divide user preferences and customization in a number of ways. I've shown one approach in the following pages.



Your Profile

YOUR PROFILE

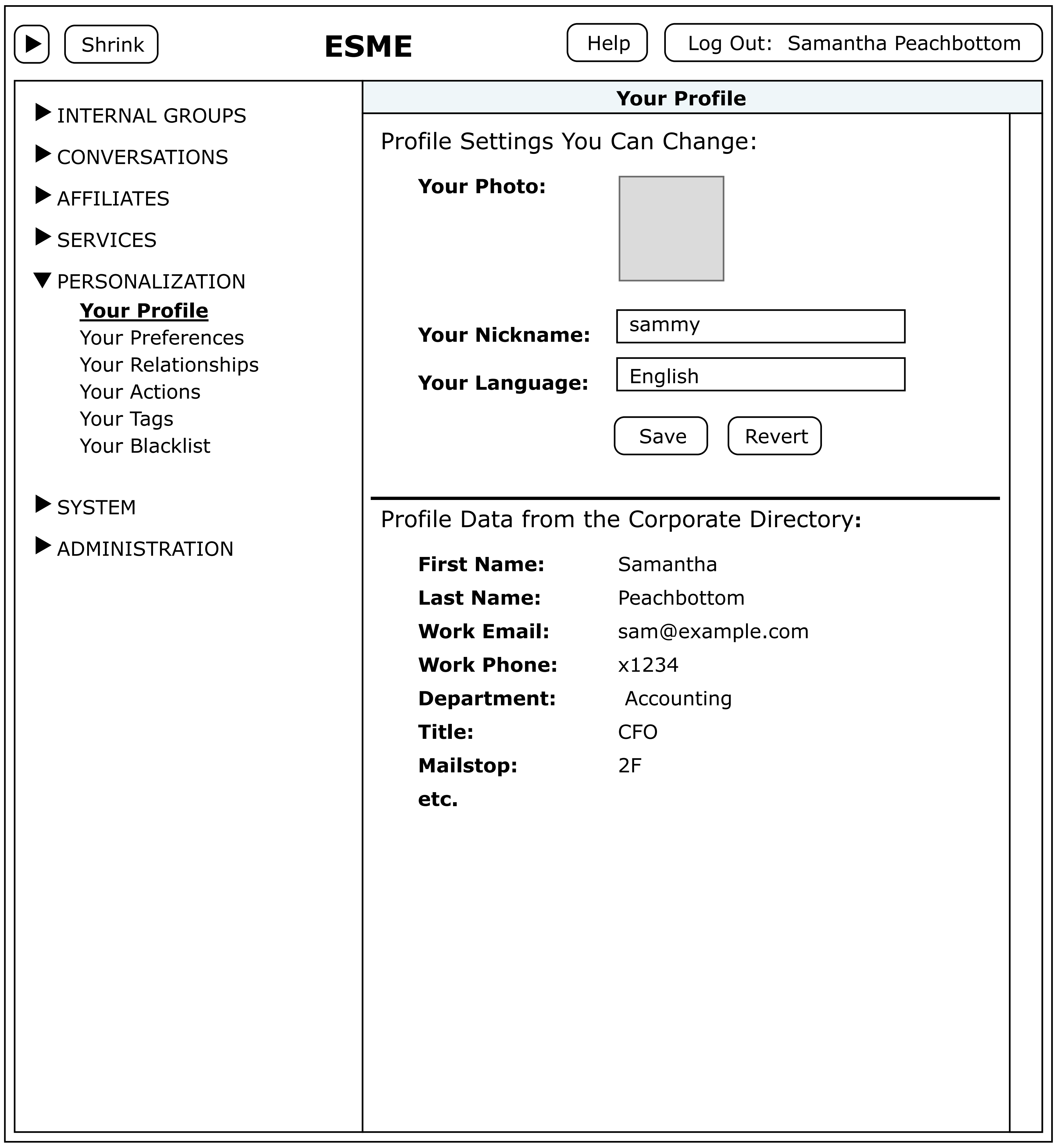
I’m assuming that we need to cover three cases:

• All user profile data is supplied by a corporate (e.g. LDAP) server, and the end user can’t change any of it.

• All user profile data is supplied by the user and can be edited at will.

• A combination of both of the above.

The illustration shows what the latter case might look like.



Your Profile

Profile Settings You Can Change:

Your Photo:

Your Nickname: sammy

Your Language: English

Profile Data from the Corporate Directory:

First Name:	Samantha
Last Name:	Peachbottom
Work Email:	sam@example.com
Work Phone:	x1234
Department:	Accounting
Title:	CFO
Mailstop:	2F
etc.	

Your Preferences

YOUR PREFERENCES

If there are user-editable settings that determine general behavior, appearance, etc. then we might wish to put them all on this page.

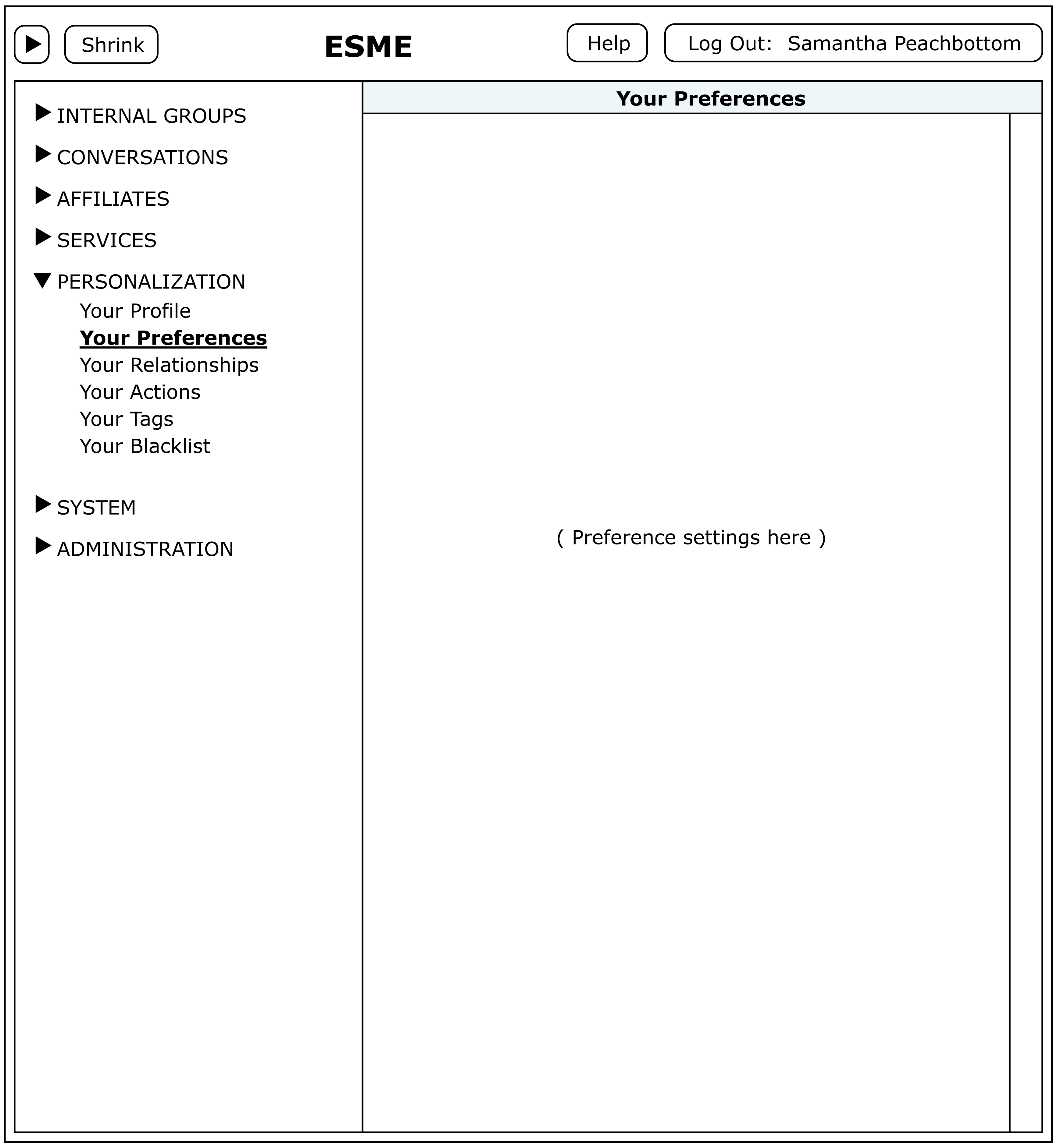
For example:

- Should ESME automatically log out after a period of inactivity?

- How many message should each stream display?

- If we offer a choice of visual themes, let the user choose it here.

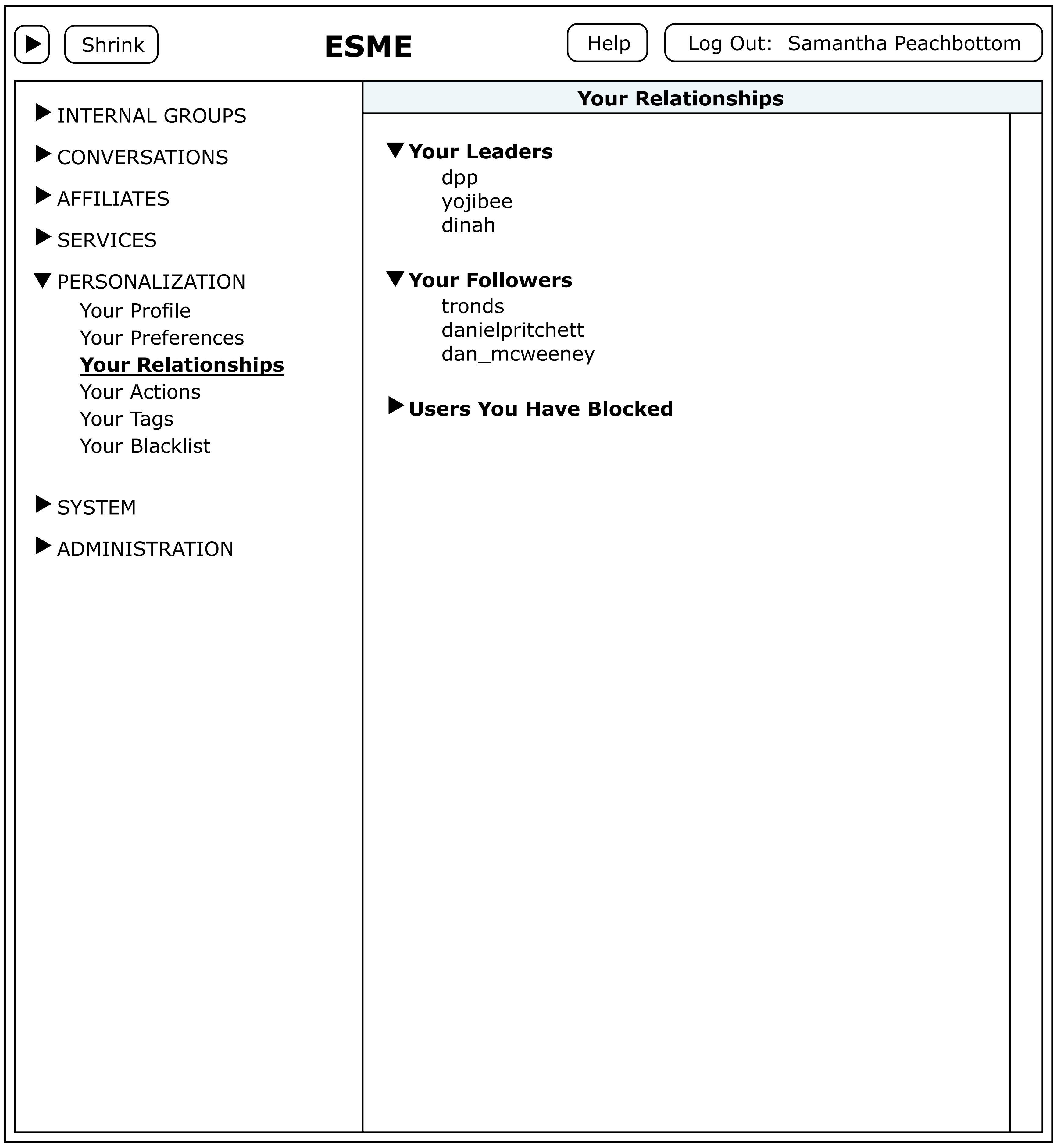
- What about letting users choose font size?



Your Relationships

YOUR RELATIONSHIPS

All your settings that determine your relationships
to other users could be managed on this page.



Your Relationships

▼ **Your Leaders**
- dpp
- yojibee
- dinah

▼ **Your Followers**
- tronds
- danielpritchett
- dan_mcweeney

▶ **Users You Have Blocked**

Your Actions

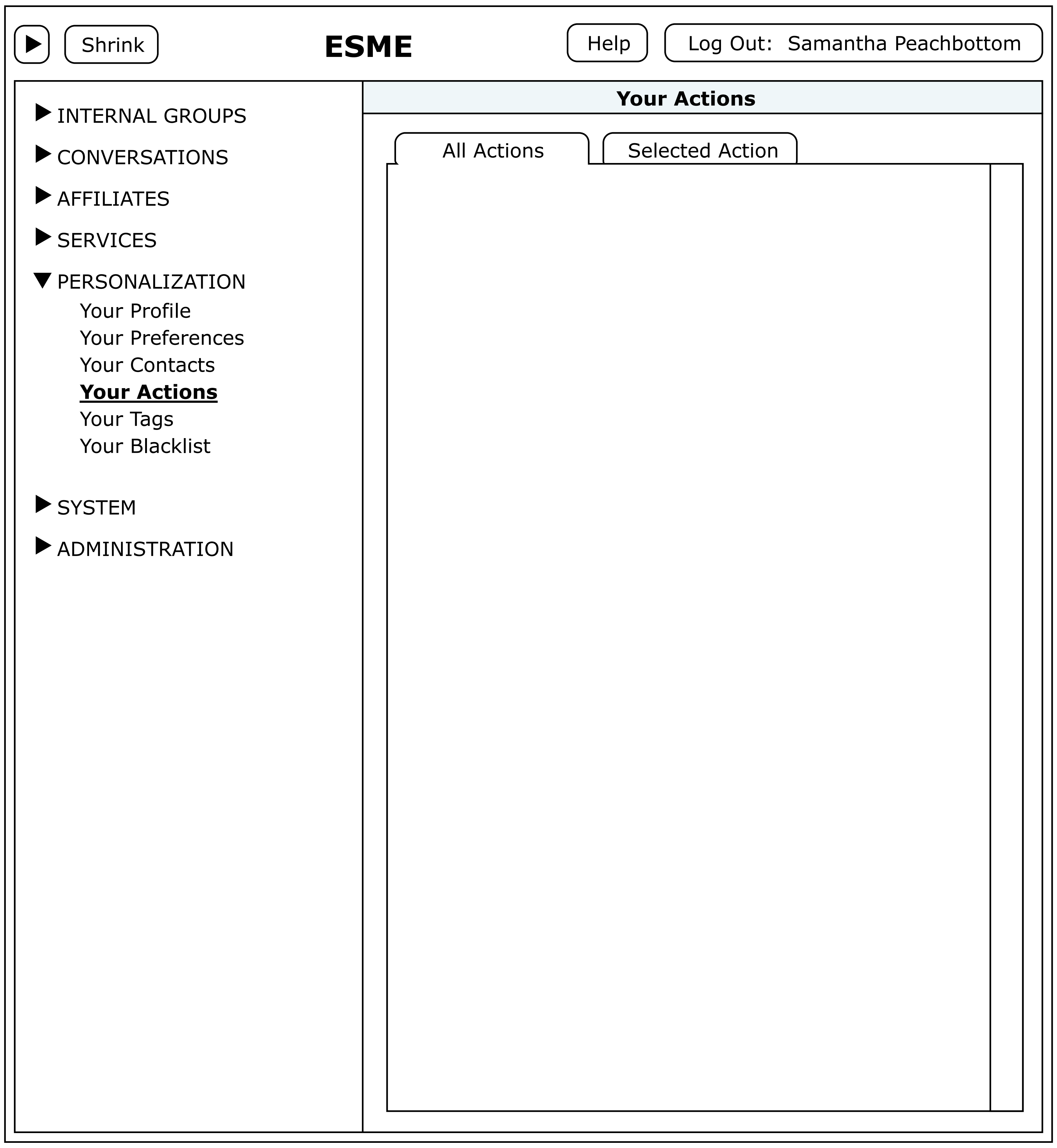
YOUR ACTIONS

I understand we want to support something called "actions" which would be akin to email filtering rules. I haven't looked into this so can't say what the UI would look like, but let's say that action management is done from this page.

Given the complexity of action editing and management I do envision this probably ending up as a tabbed UI: at least one tab to list all the actions, and another tab to edit one selected action at a time.

I expect we will want to be able to:

- Add a new action.
- Duplicate an existing action.
- View and Edit an existing action.
- Test and Troubleshoot an existing action.
- Enable or Disable any action.
- Delete an existing action.
- Possibly archive, export, import and/or share actions with others.

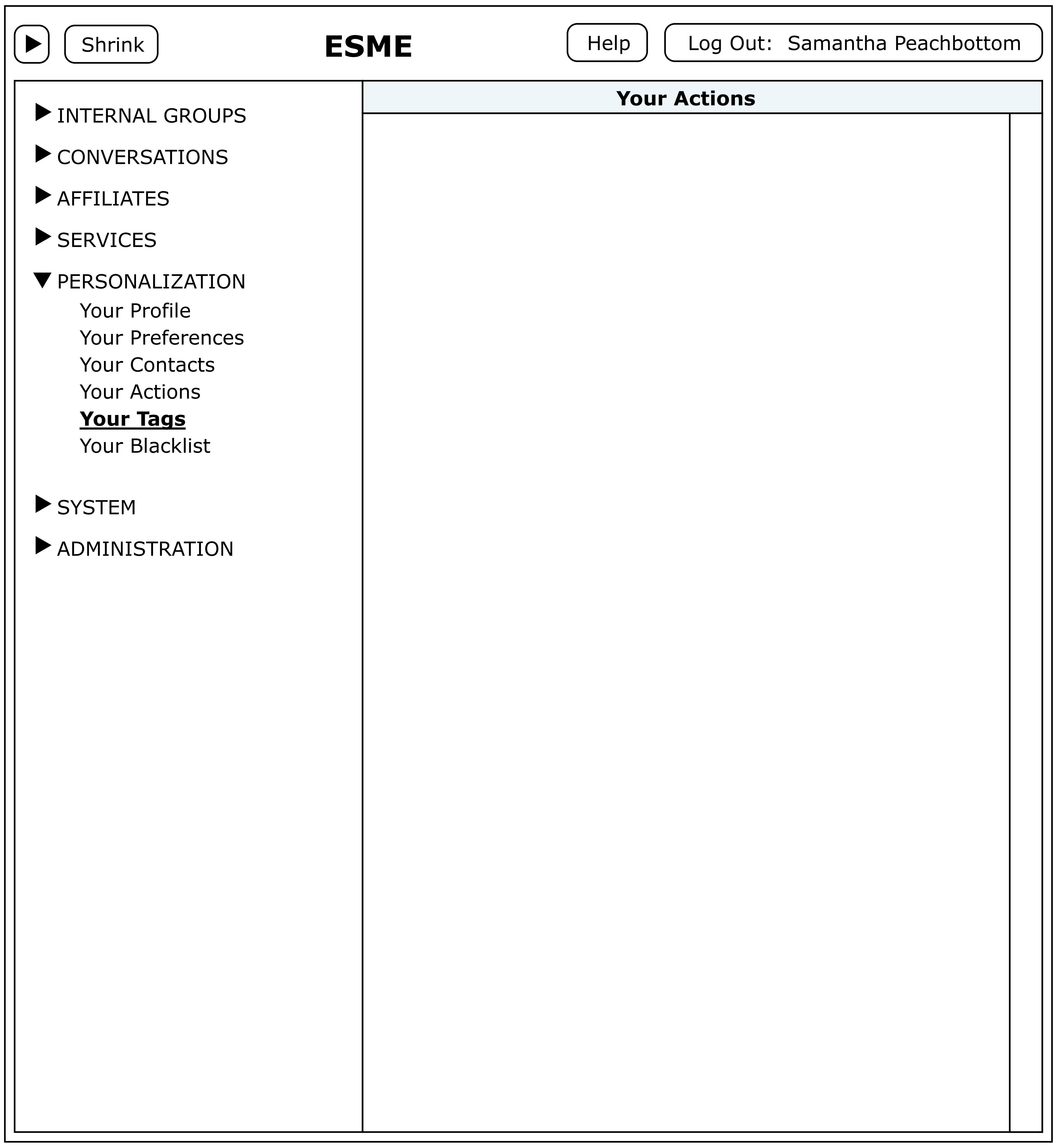


YOUR TAGS

I had this thought that maybe you'd want to create a list of tags that you want to re-use often. On this page you could create a list of those tags, and then they can be supplied to you when typing a message: either as a popup menu, or with auto-completion as you type, etc. This would insure you always spell your tags the same way.

I also had the thought that you might want to compile a list of tags that others had created — perhaps because you wanted to track them, or to use them yourself, etc.

I haven't pursued these thoughts. I've just put a page here as a placeholder until they are more fully developed.

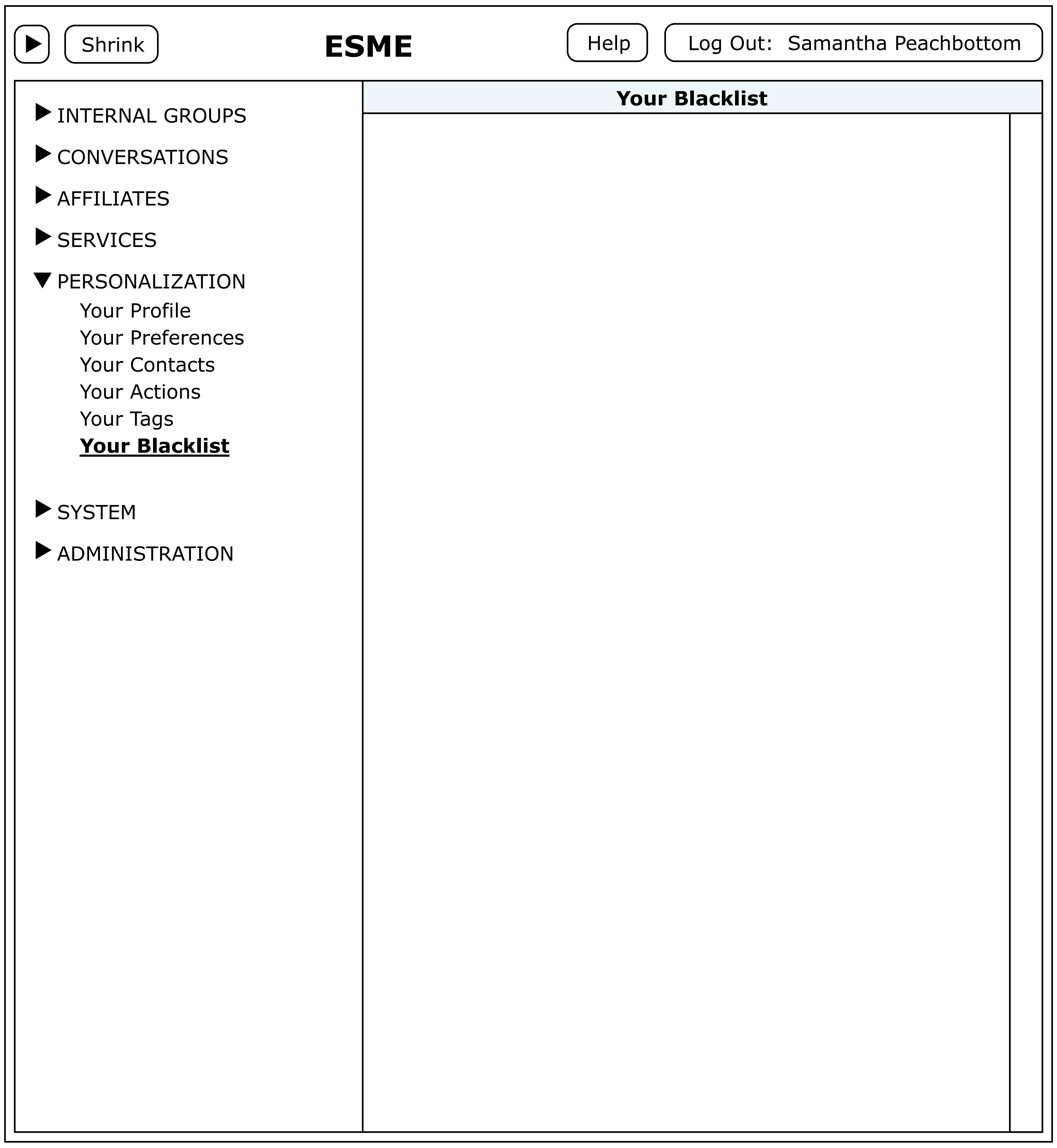


YOUR BLACKLIST

I thought that users might want to prevent ESME from showing them messages containing certain tags, or send by certain users, etc.

for example you might want to have two lists: one for banned users, and one for banned tags.

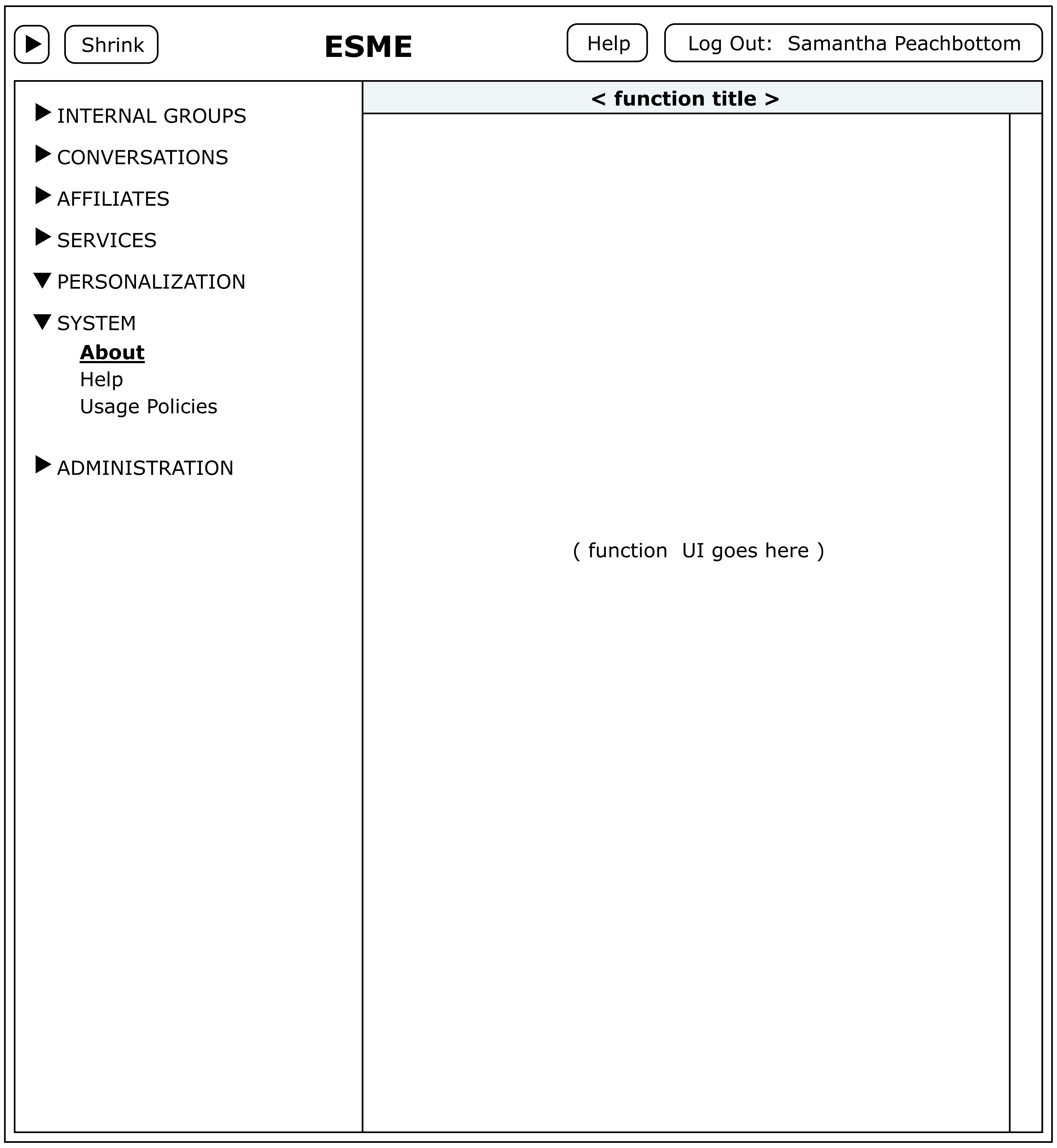
Again, I haven’t thought this through, but have put this page here as a placeholder.



SYSTEM FUNCTIONS

There are a number of system or application-level functions that we need to find homes for.

One very natural and convenient way to do this is simply to extend the table of contents to have a section of "System" functions.



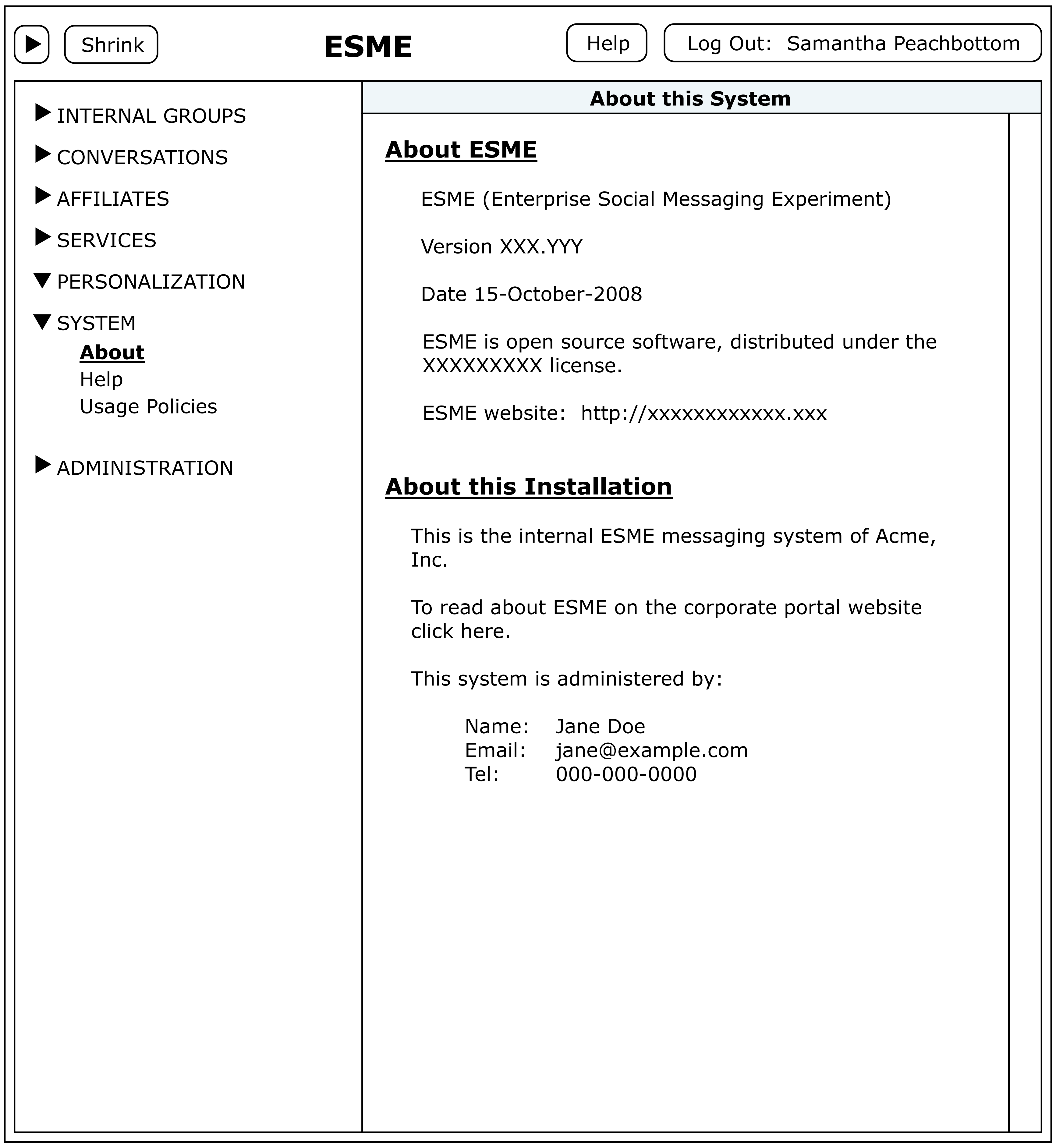
System — About

SYSTEM “ABOUT”

We need a place to display “About” info.

Some of this info is about ESME, and is defined by the developers.

Some of this info is about the organization that owns and operates this instance of ESME, and should be specifiable in the system administration UI (whereever that ends up being).



About this System

About ESME

ESME (Enterprise Social Messaging Experiment)

Version XXX.YYY

Date 15-October-2008

ESME is open source software, distributed under the XXXXXXXXX license.

ESME website: http://xxxxxxxxxxxx.xxx

About this Installation

This is the internal ESME messaging system of Acme, Inc.

To read about ESME on the corporate portal website click here.

This system is administered by:

Name:	Jane Doe
Email:	jane@example.com
Tel:	000-000-0000

SYSTEM "HELP"

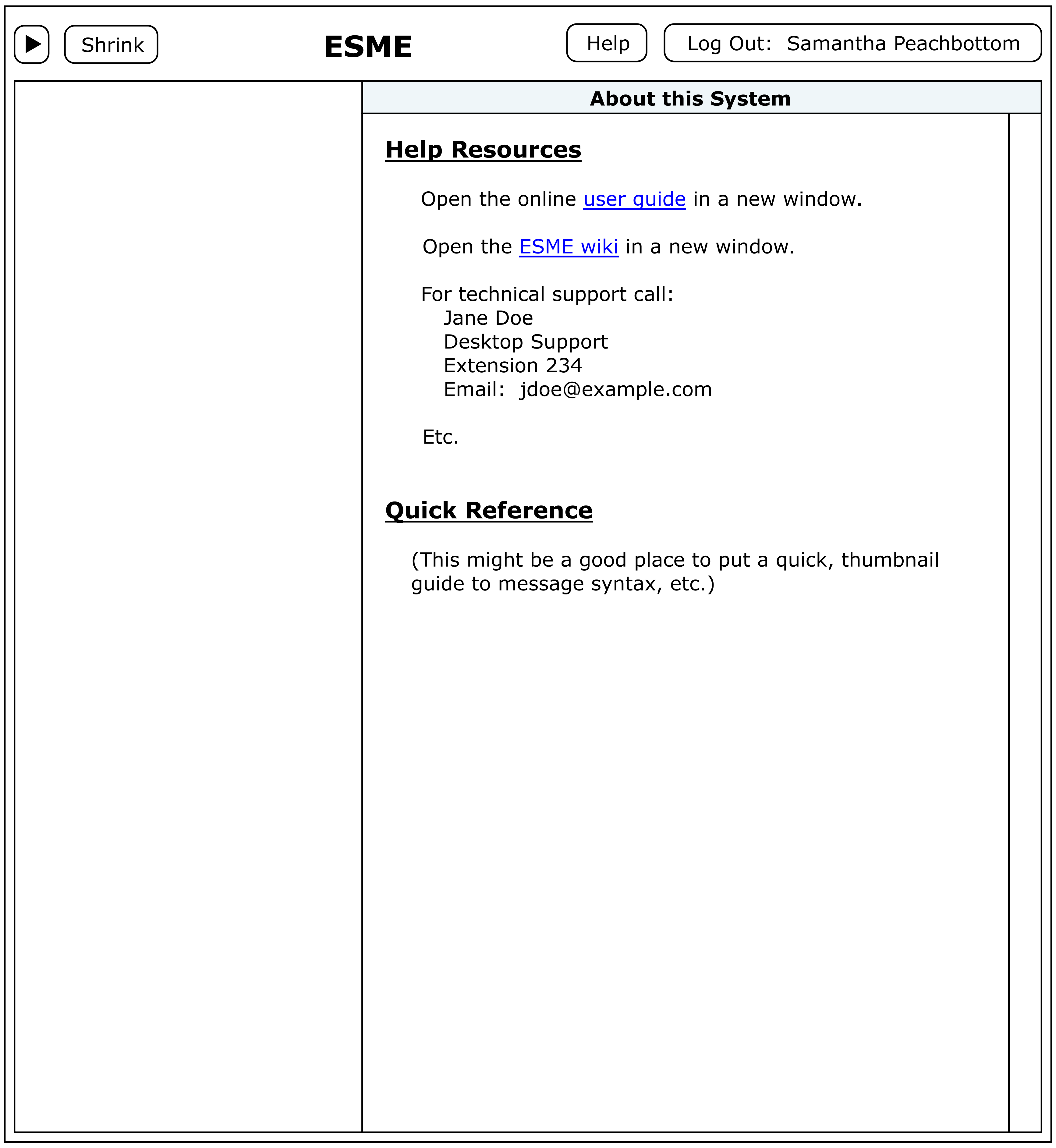
In addition to the "Help" button at the top of the window, it might be nice to have a comprehensive list of support resources.

I envision the "Help" button leading to an ESME-supplied user guide.

I envision the info on this page being configured by the system administrator, and pointing to resources supplied by the organization.

This might also be a very nice place to put some terse quick reference guides: service call syntax, ESME message syntax, etc.

Since quick-reference guides are most likely to be supplied by ESME, we could have a tabbed UI, with the first tab being organization-supplied info, and the following tab(s) containing the ESME-supplied quick reference(s).



Help Resources

Open the online user guide in a new window.

Open the ESME wiki in a new window.

For technical support call:
Jane Doe
Desktop Support
Extension 234
Email: jdoe@example.com

Etc.

Quick Reference

(This might be a good place to put a quick, thumbnail guide to message syntax, etc.)

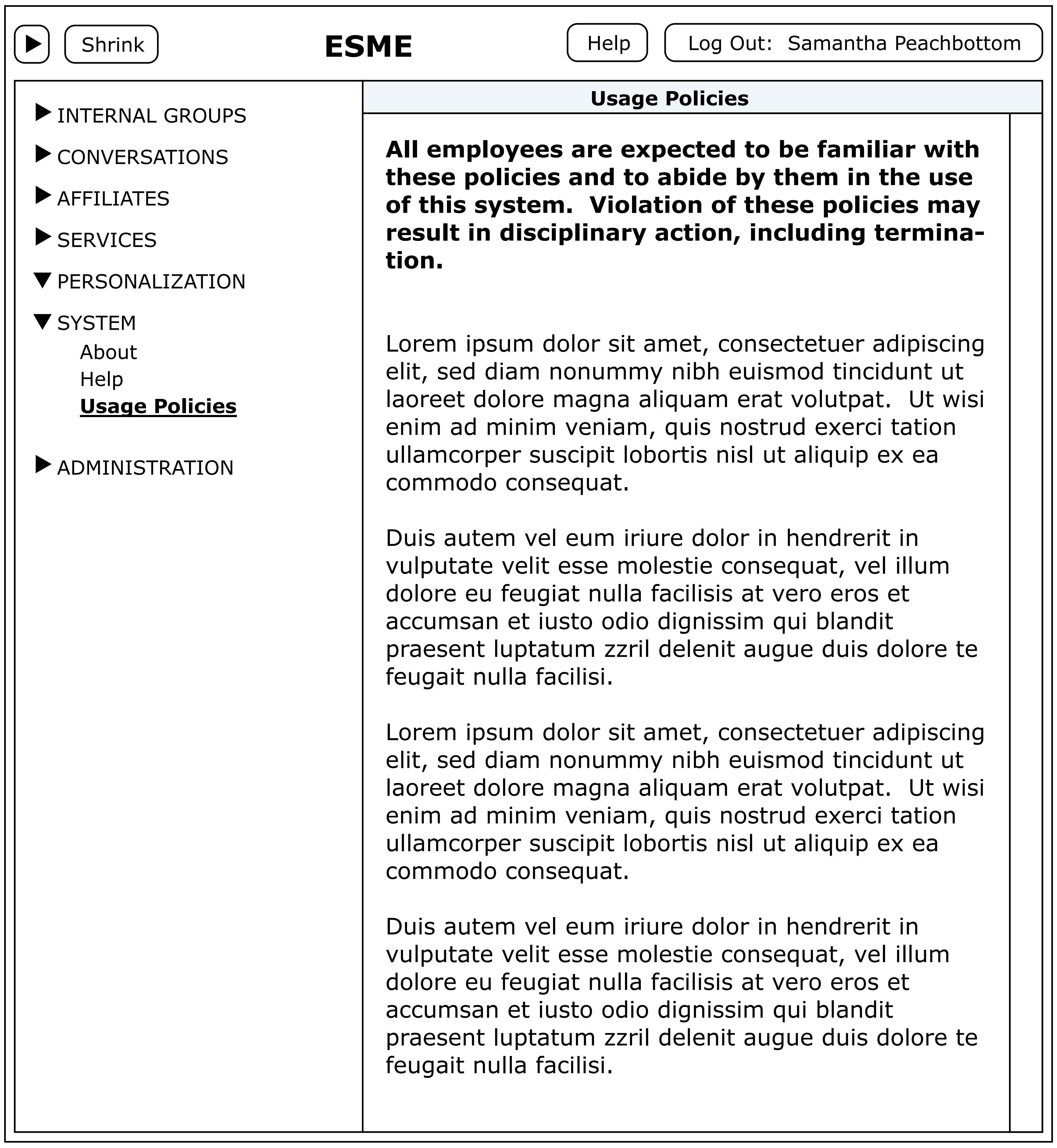
SYSTEM "POLICIES"

Most big organizations will have official usage policies for everything, including micro-messaging.

We should give them a place to display these policies to users.

These policies should be entered/configured by the system administrator.

The system should allow for a different set of policies being displayed to affiliates than to internal users.



All employees are expected to be familiar with these policies and to abide by them in the use of this system. Violation of these policies may result in disciplinary action, including termination.

Lorem ipsum dolor sit amet, consectetuer adipiscing elit, sed diam nonummy nibh euismod tincidunt ut laoreet dolore magna aliquam erat volutpat. Ut wisi enim ad minim veniam, quis nostrud exerci tation ullamcorper suscipit lobortis nisl ut aliquip ex ea commodo consequat.

Duis autem vel eum iriure dolor in hendrerit in vulputate velit esse molestie consequat, vel illum dolore eu feugiat nulla facilisis at vero eros et accumsan et iusto odio dignissim qui blandit praesent luptatum zzril delenit augue duis dolore te feugait nulla facilisi.

Lorem ipsum dolor sit amet, consectetuer adipiscing elit, sed diam nonummy nibh euismod tincidunt ut laoreet dolore magna aliquam erat volutpat. Ut wisi enim ad minim veniam, quis nostrud exerci tation ullamcorper suscipit lobortis nisl ut aliquip ex ea commodo consequat.

Duis autem vel eum iriure dolor in hendrerit in vulputate velit esse molestie consequat, vel illum dolore eu feugiat nulla facilisis at vero eros et accumsan et iusto odio dignissim qui blandit praesent luptatum zzril delenit augue duis dolore te feugait nulla facilisi.

ADMINISTRATION FUNCTIONS

Two endpoints of the ESME user spectrum are the system administrator who installs, updates, configures and runs the machinery, and the end users who message each other.

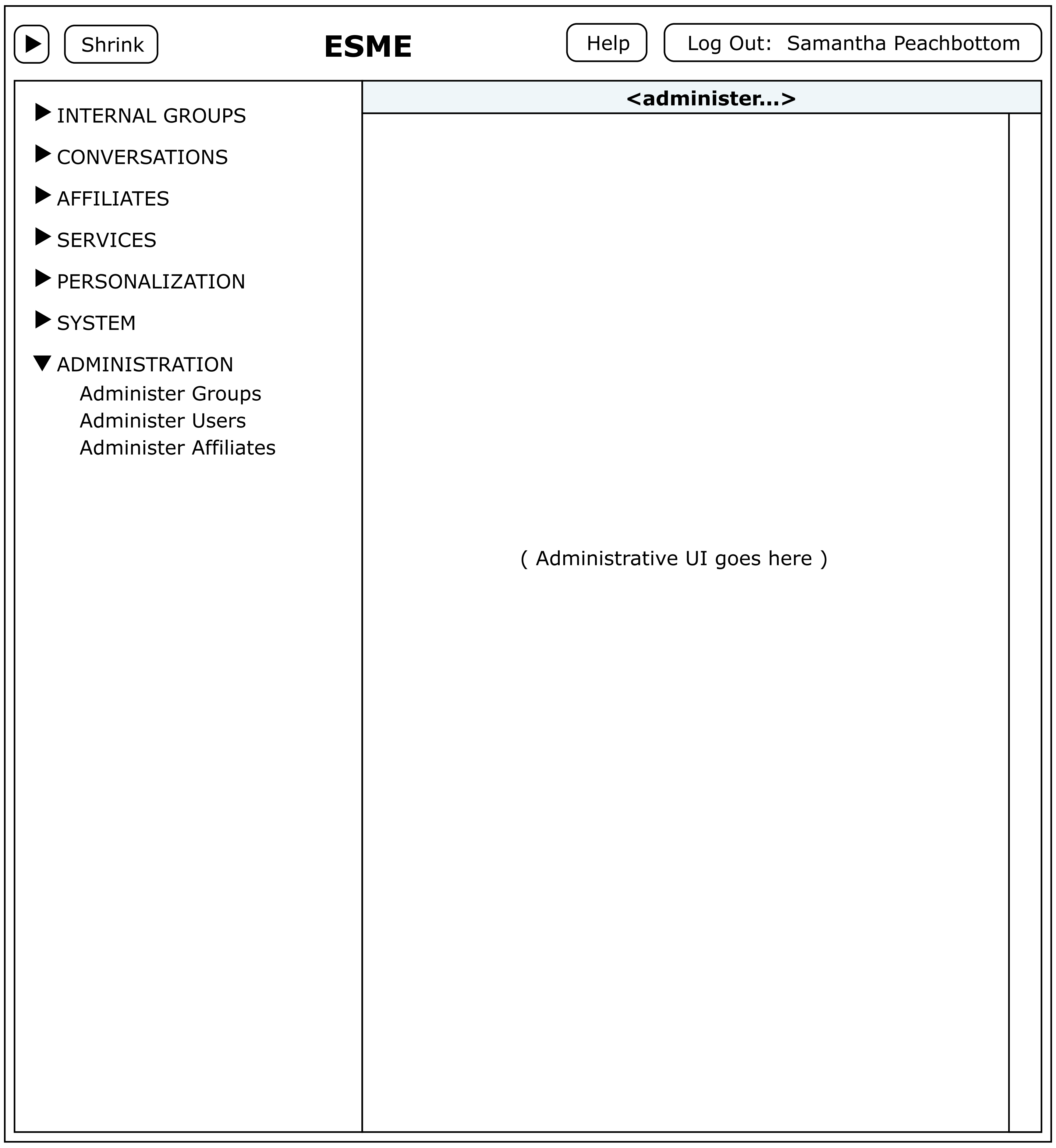
Between these ends of the spectrum I think there are one or more places for users with special responsibilities and privileges, who will need appropriate UIs.

It seems to me that some of these cases could be modelled as system preferences, in contrast to the user preferences/personalizations that are available to every user, and that the UIs for these functions can be placed within the ESME end-user window rather than in separate windows.

So maybe we can say that if a user has any of these intermediary administrative roles/privileges then his/her left sidebar displays an additional section called (for example) “Administration”, and which gives access to the UIs for the configuration and management functions associated with those roles.

Examples of roles/responsibilities that might fit this model include:

- Administering any groups for which you are a moderator or group owner.
- Managing the user base.
- Managing one or more affiliates.



The Right Sidebar

Right Sidebar

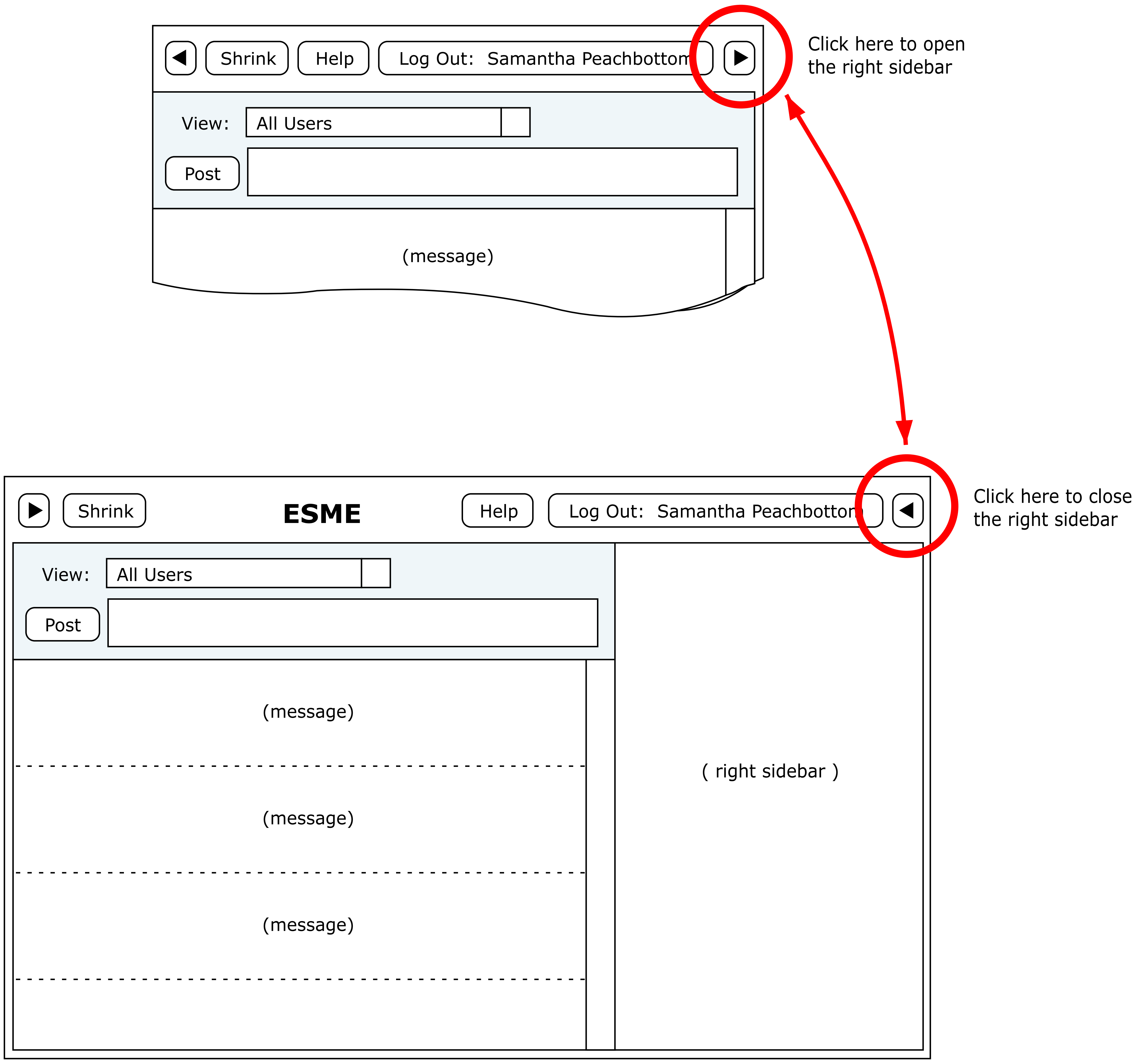
RIGHT SIDEBAR

There are many functions that ESME will need that have not yet been addressed. In a conventional GUI many of these functions would be accessed through a menubar and set of dialog boxes or secondary windows. I suggest that a better way to provide access to these functions is through an optional right sidebar whose contents will change as different items are selected in the main body and left sidebar. This will provide interactive, modeless, context-sensitive UI switching; all within the single ESME window.

Accessing the right sidebar will be done in the same way as accessing the left sidebar: Let’s add a button to the upper-right corner of the ESME window that toggles the sidebar open and closed.

Of course the user can choose whether to have none, one or both sidebars open at the same time.

Having both sidebars open puts pretty much all the functionality of ESME at your fingertips.



Click here to open the right sidebar

Click here to close the right sidebar

Right Sidebar

HELPER PANELS

The right sidebar is envisioned to be a selection-sensitive control panel. This means that whenever you select an item in the main panel or in the left sidebar the content of the right sidebar changes to display information about, and controls that act upon, the selected item. Let us call these “helper” panels.

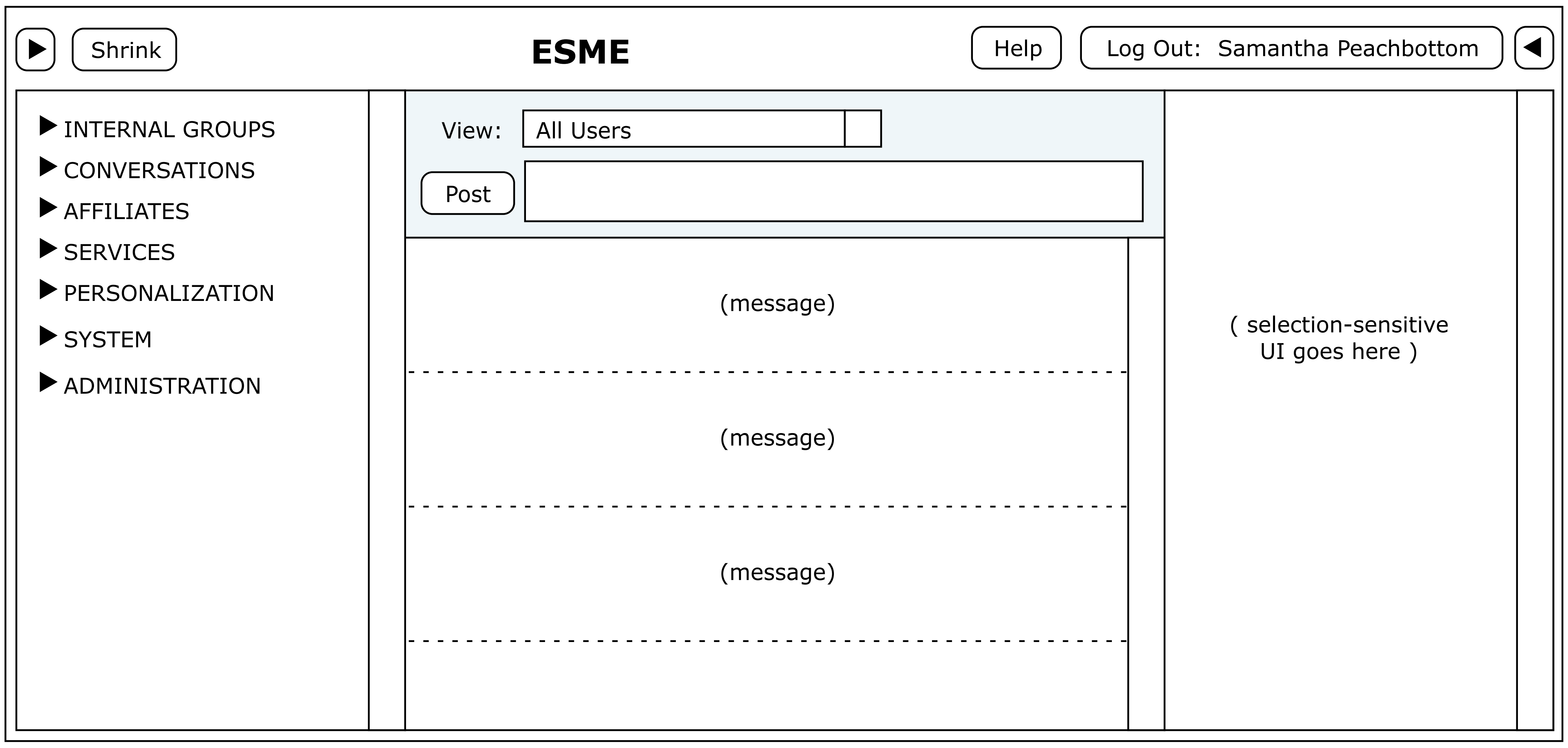
Here are some of the types of items that can be selected, and for which helper panels can be displayed:

- A user name
- A tag
- A message
- A service call
- The current message stream
- And... ?

BUTTON ALIGNMENTS

Note that, as shown in the illustration below:

- The left sidebar button and the “Shrink” button are grouped together and left-aligned.
- The right sidebar button, “Log Out” button and “Help” button are grouped together and right-aligned.
- The ESME logo (etc.) is centered in the remaining space (or hidden if the window is too narrow.



TWO CASES TO CONSIDER

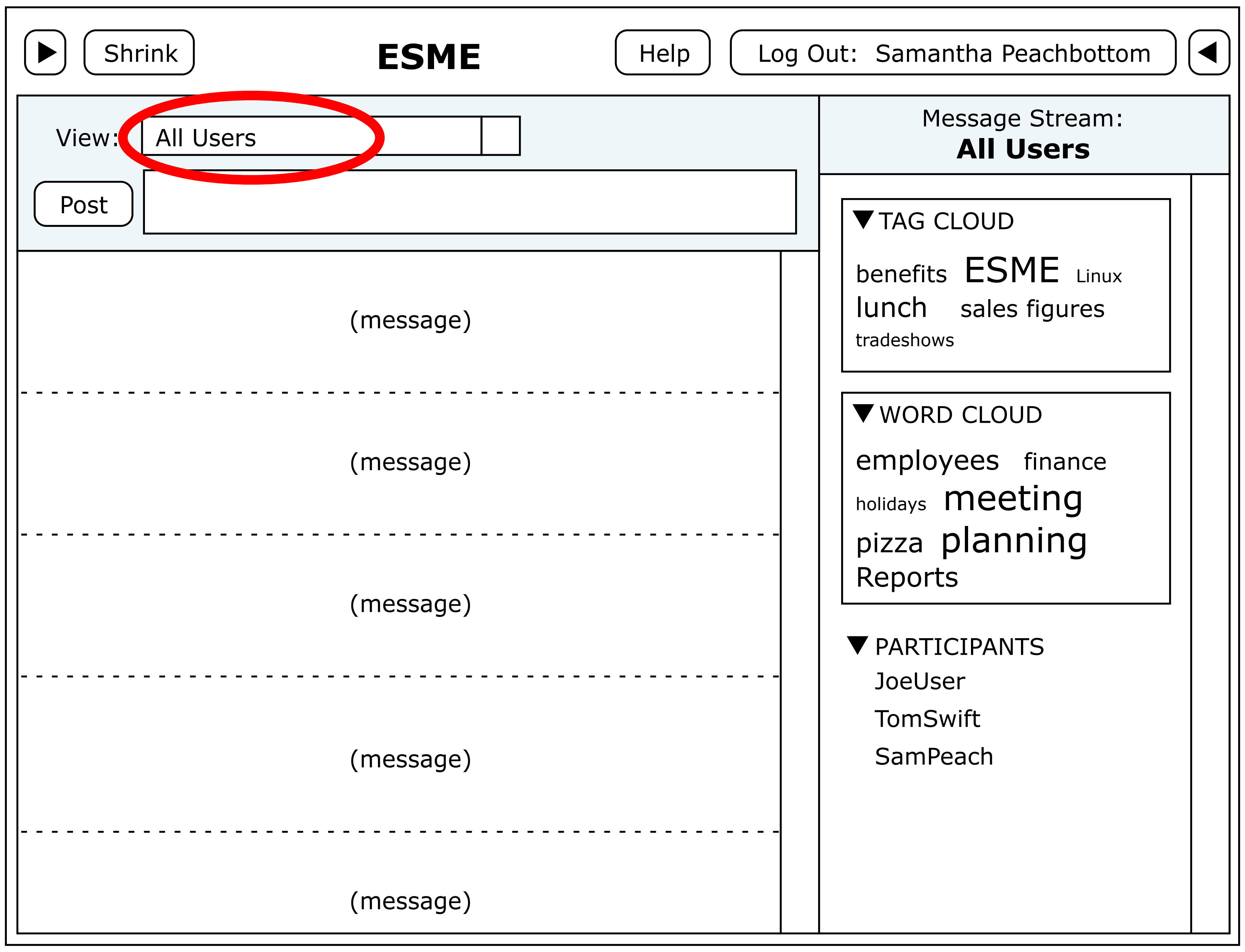
When the right sidebar is open there are two general cases that affect its content. The first is when a message stream is being displayed in the main panel, and the second is when the main panel is displaying a UI invoked from the Personalization, System or Administration sections of the left sidebar. In this latter case the question is what to put in the right sidebar.

My preliminary thought is that we should combine the main panel and the right sidebar into one area, thus devoting the whole center and right regions of the window to the non-message-stream-UI. Since that UI has to be able to scale horizontally to respond to changes in the window width, this would produce natural, predicable behavior; and it would maximize the screen real-estate available to the user’s current task focus.

Message Stream Helper

MESSAGE STREAM HELPER PANEL

Here’s an example of what the helper panel might look like for a message stream. I have not really thought through everything you might want in this helper, but have illustrated some example content.



Selecting the Stream

So how do you select a message stream? Certainly any time you switch to one using the left sidebar or the “View” menu the chosen stream should become selected.

After that, if you’ve been clicking around and want to re-establish the overall stream as the selection, perhaps just clicking in the “View” menu without changing streams, or clicking on the “View” label could reset the selection to be the entire stream.

The Clouds and the Participants List

Clearly we’ll have to limit the length of these lists to something managable. Perhaps just show the top ten in each category?

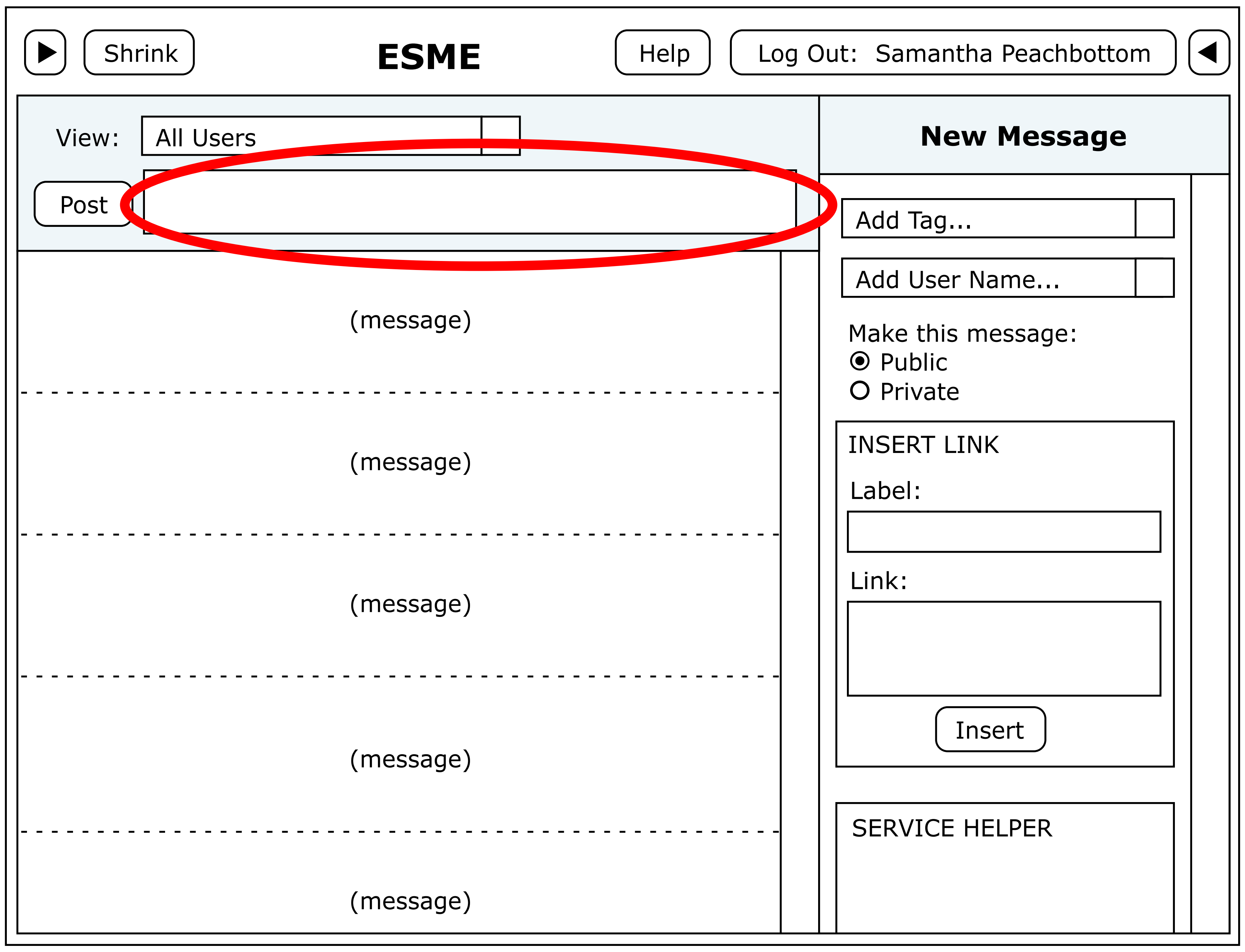
Message Editor Helper

MESSAGE EDITOR HELPER PANEL

Here’s an example of what the helper panel might look like for a message being edited.

This helper would be displayed whenever the keyboard focus is in the message box.

I have not really thought through everything you might want in this helper, but have illustrated some example content.



Miscellaneous Notes

- Remember that the message will be sent to or in the context of the message stream that is currently being viewed.

- There may be privacy conflicts that the above example does not handle (e.g. when you try to send a public message to a private group).

- We may want to switch to other helpers when content within the message is selected (e.g. a user name, a tag or a service call).

The “Insert Link” Box

This is an idea for how to handle an issue that David Pollak raised, and which I’ve seen manifested in Twitter; where you want to insert a long URL but you want to keep the text of the message short.

The idea here is that, Like an HTML anchor tag, you can have a short “label” that is displayed in the message stream, and that hiding behind it can be a long URL that is opened in a new window when you click on the label.

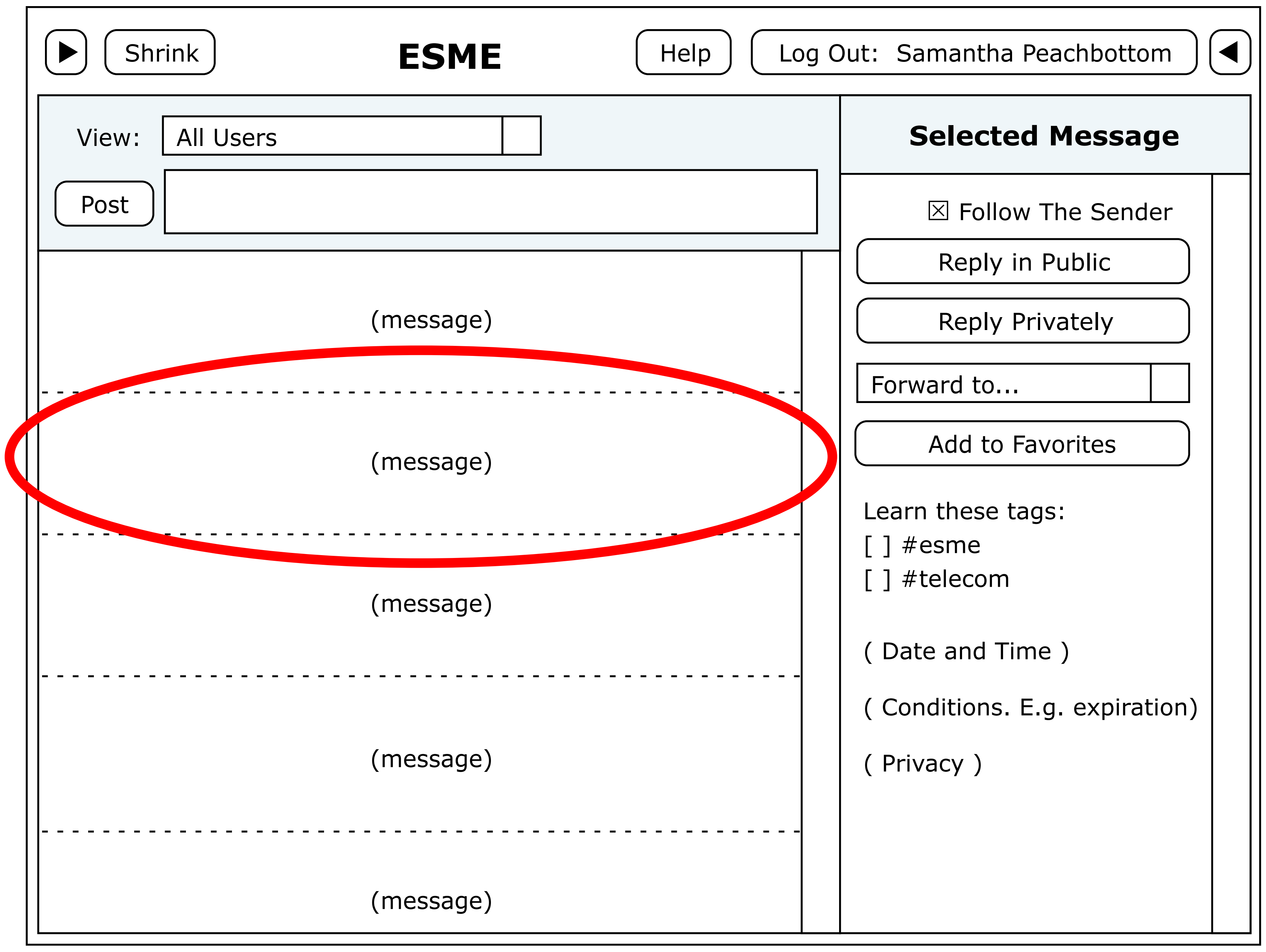
The “Service Helper” Box

Maybe for each service there can be a small, custom UI that helps users build useful, syntactically correct parameter strings...?

MESSAGE HELPER PANEL

Here's an example of what the helper panel might look like after clicking on a message in the message stream.

I have not really thought through everything you might want in this helper, but have illustrated some example content.

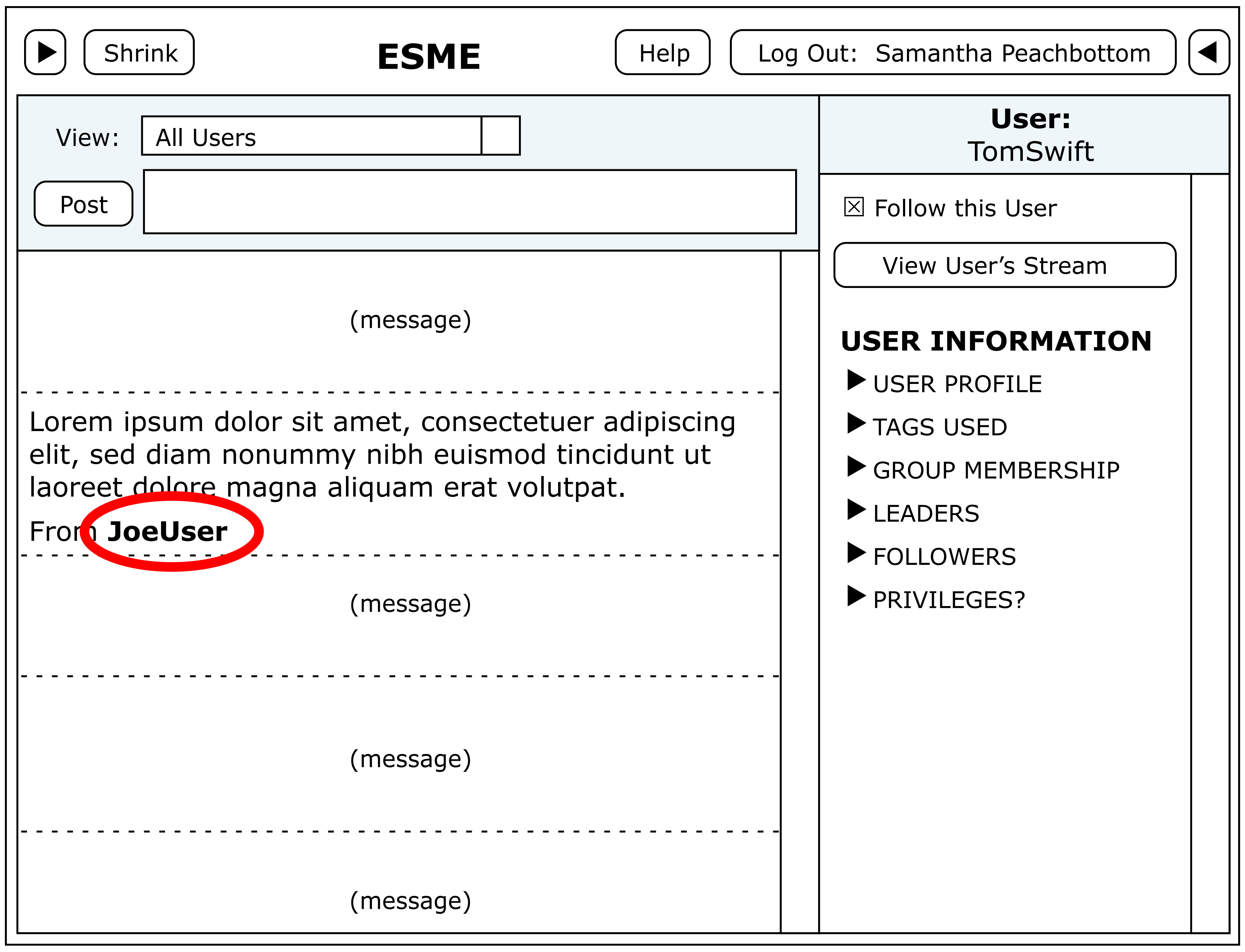


User Helper

MESSAGE HELPER PANEL

Here’s an example of what the helper panel might look like after clicking on the name of a user.

I have not really thought through everything you might want in this helper, but have illustrated some example content.



Lorem ipsum dolor sit amet, consectetuer adipiscing elit, sed diam nonummy nibh euismod tincidunt ut laoreet dolore magna aliquam erat volutpat.

From **JoeUser**

User Information

I envision that we could have whatever categories of information we want about a user.

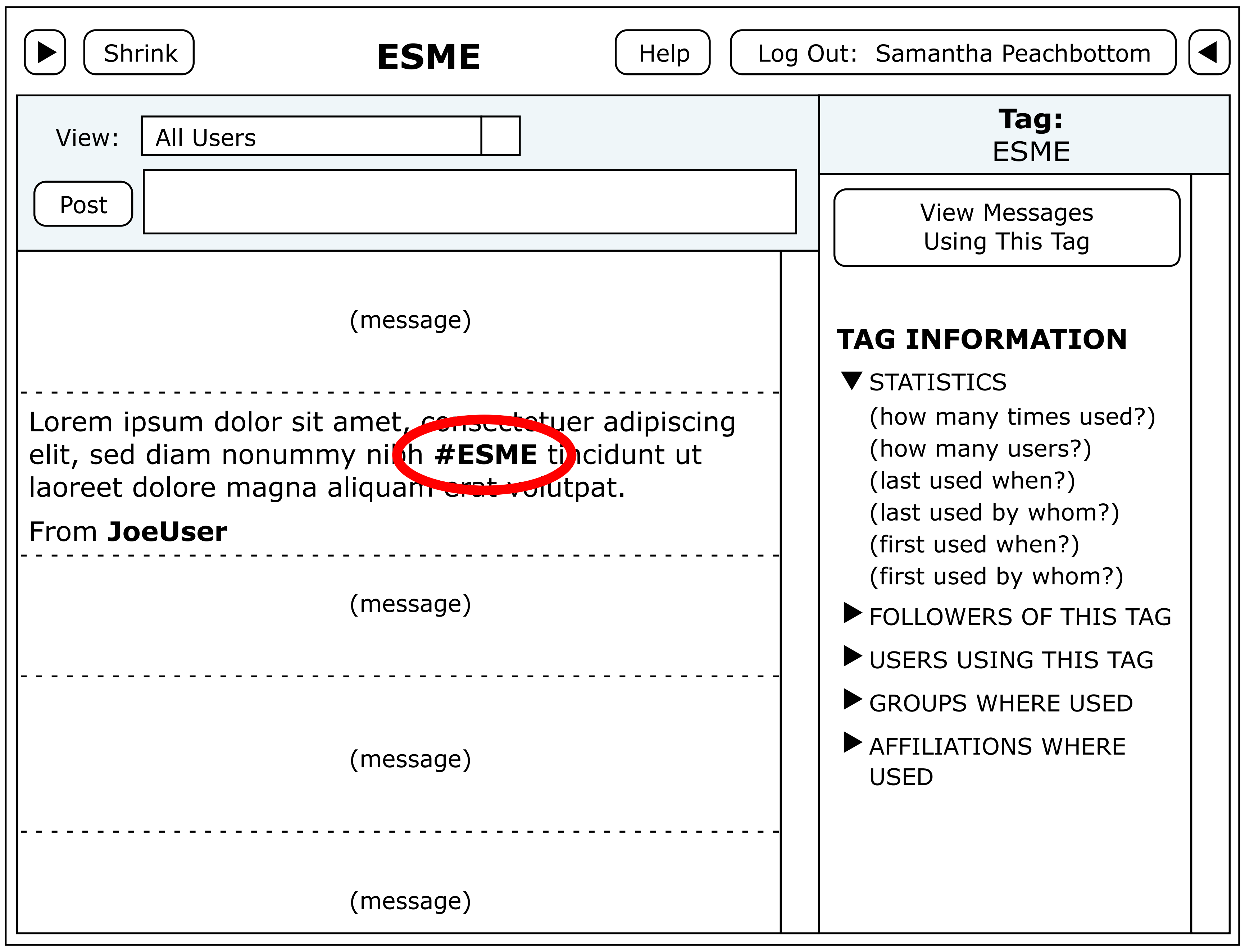
I envision that clicking on the name of a user or tag or group, etc. in the user info display would shift the display to show you info about the item you clicked on. Haven’t yet thought about how this would change the message stream view.

Tag Helper

TAG HELPER PANEL

Here’s an example of what the helper panel might look like after clicking on a tag.

I have not really thought through everything you might want in this helper, but have illustrated some example content.



(message)

Lorem ipsum dolor sit amet, consectetuer adipiscing elit, sed diam nonummy nibh **#ESME** tincidunt ut laoreet dolore magna aliquam erat volutpat.
From **JoeUser**

(message)

(message)

(message)

TAG INFORMATION

▼ STATISTICS
- (how many times used?)
- (how many users?)
- (last used when?)
- (last used by whom?)
- (first used when?)
- (first used by whom?)

▶ FOLLOWERS OF THIS TAG

▶ USERS USING THIS TAG

▶ GROUPS WHERE USED

▶ AFFILIATIONS WHERE USED

Searching

OVERVIEW OF SEARCHING

The Message Dataspace

The ESME dataspace consists primarily of an "ocean" of messages from the organization's users, plus smaller "pools" of messages for the affiliates and private groups.

The way you navigate this dataspace is by searching for messages with the attributes you want, then displaying the list of qualifying messages in chronological order.

All of the named message streams in the left sidebar are really just the names of stored searches (queries) against the database of messages.

Attribute Datasets

In addition to the set of messages there are datasets that provide the attributes of messages that are searched for in queries. These consist of the sets of:

- Users
- Groups
- Tags
- Words used in messages
- Affiliates
- Services

Search Types

We should support searching for items of specific types (e.g. to find a specific user or group).

We should also be able to search for all messages that share a combination of attributes drawn from all of the attribute types, plus any other message metadata (e.g. "all messages from user TomSwift tagged with ESME send last week").

Saving Searches

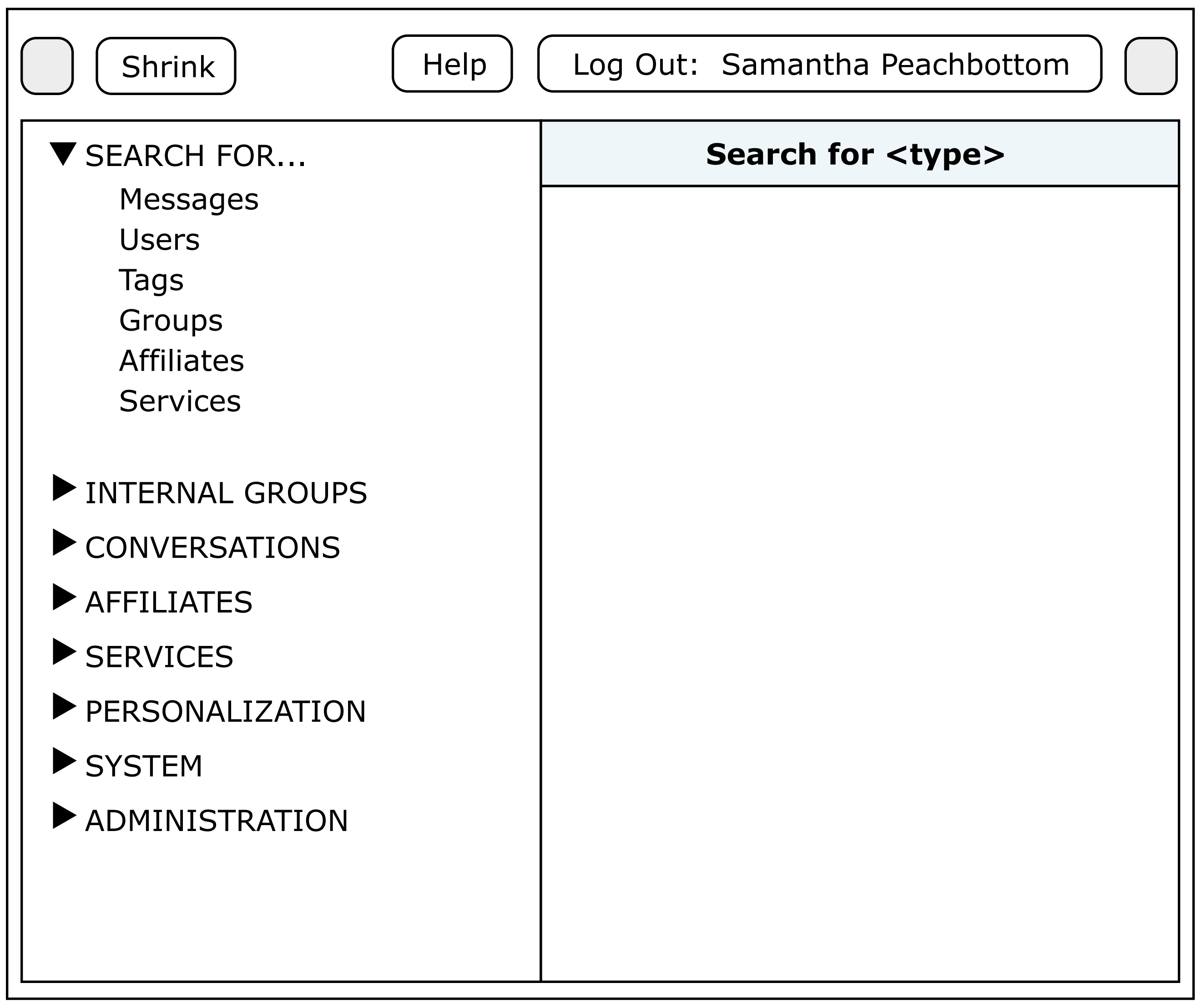
Searches that find messages conforming to certain criteria should be able to be saved as user-defined, user-specific "message streams" that are added to the list of named streams available in the left sidebar.

The UI Framework for Searching

My initial thought is that we should have a "Search" section in the left sidebar, and that in that section we should offer various types of searches.

We might also want to make it that when a search option has been chosen we disable the sidebar open/close buttons, and leave them disabled until the user chooses a message stream from the left sidebar (but then this introduces an awkward mode, so I'd like to think of something better).

Below is an illustration of this framework. Note that for illustration I've place the "Search" section at the top of the table of contents:



About Messages

MESSAGE ELEMENTS

In this section let us explore the components, attributes, content, aka the “elements” that make up messages. Some of these elements appear in the end-user visible content of a message, others are attributes or metadata attached thereto.

MESSAGE POOLS

A given ESME installation can have multiple pools of messages. Each user has access to one or more of those pools. A user having access to a given pool has access to all messages within that pool. Users who do not have access to a given pool may never see any of the messages in that pool. Messages may not be cross-posted between pools. This model allows for clusters of users to have secure or confidential conversations among themselves without having to resort to other communications systems.

The default pool that all ESME installations have is the organization’s public, internal pool. All internal users have access to this pool.

Optionally, private groups may be created. Each private group has its own pool of messages. Only members of a given private group may access messages in the pool.

Note that public groups may also be created, but these do not have private message pools. Thus all internal users have access to all public groups, and hence to all their messages.

Optionally, affiliates may be created. An affiliate is essentially a private group that has both internal and external users. This is envisioned as a way for an organization to extend the benefit of ESME to its communications with outside contractors, vendors, etc. while maintaining strong walls of secrecy and confidentiality.

MESSAGE STREAMS

At the UI level, a message stream is a set of messages, that all have a meaningful relationship to each other, and that are listed in chronological order.

At the implementation level, I assume that a stream is a stored database query that is run whenever a user wants to view the stream.

Pool Streams

Each message pool by default has a message stream consisting of all the messages in the pool.

Built-in, User-Specific Streams

For each user the system provides a set of built-in, undeletable streams.

All Users — This is the organization’s internal, public stream.

Your Stream — This is the set of public messages that: you posted, your leaders posted, or are replies to your posts.

Your Leaders — This is the set of public messages that: your leaders posted.

Your Followers — This is the set of public messages that: your followers posted.

Replies — This is the set of all public messages that were marked as being posted in response to your posts.

Your Favorites — All messages you mark as favorites are added to this stream.

Auto-Generated, User-Specific Streams

For each user the system automatically generates certain streams based upon your usage.

Conversations — Whenever you send or receive a private message a separate "conversation" stream is created that involves just you and the other user. (Hmm, I guess this is another form of private stream...)

Groups — Whenever you become a member of a group, that group's stream becomes accessible to you.

Services — Whenever you invoke a service, a separate "service" stream is created that involves just you and the service. (Hmm, I guess this is another form of private stream...)

Affiliates — Whenever you become a member of an affiliate group (either as a user internal or external to the organization), that group's stream becomes accessible to you.

User-Defined User-Specific Streams

In addition to all the above, any time a user does a search for messages I would like him/her to be able to save that query as a named message stream, and have this added to their set of available message streams.

Temporary, or Ad-Hoc Streams

I've noticed as I work back and forth across this design that there are times when we need to show streams (e.g. you click on a user name and ask to see their personal stream) that aren't already in the user's stream list. We'll have to figure out a nice way to show that these are ad-hoc streams, and perhaps to let you add them to the user's list of user-defined streams.

MESSAGE ELEMENTS

Messages, as they are displayed to users in message streams, are comprised of a message body and some visible metadata.

The message body we limit to (say) 140 characters.

The metadata we display might include:

- The sender's picture
- The sender's name
- When the message was posted
- Any tags not in the message body
- Any "conditions" (see below)

The Body of the Message

The body of the message is the text string that readers are supposed to read. This content can contain a variety of special terms (commands, user names, tags, etc.) as well as "ordinary" text.

Commands

Commands are special terms or sub-strings that can be embedded into the body of a message to give instruct ESME of process the message in a special way. Commands will be addressed in more detail later.

Sender

One piece of metadata associated with a message is a reference to the user who sent it, which we usually display as the picture and the name (or nickname?) from the senders' profile.

Recipient(s)

A sender may force a message to appear in a reader's personal stream by including a recipient (or routing?) command (in the form *@userName)* in the body of the message.

Links

ESME messages are forced to be short in length, yet senders will often want to use messages as a way of making internet-resident resources available to readers; but URLs can be longer than the message character limit.

To address this I propose that we mimic HTML links (anchor tags), in that we have a way (described later) to specify a short label which appears in the message body, and to associate it with a longer URL that is hidden from the reader (but invoked when a reader clicks on the label).

Tags

Senders may embed tag commands (in the form *#tagName*) in the body of messages, or can attach tags as metadata to messages.

When embedded in the message body tags count against the message character limit, and also are displayed to readers within the message body.

When attached as metadata, tags do not count against the character limit and appear as metadata displayed outside the message body.

Message Privacy

Messages sent within the context of a private group or an affiliate are by nature private to the associated pool of users.

In any pool, you might wish to send a message privately to one (or more) user(s), meaning that only the designated user(s) will receive the message. In this case you must embed a "send privately" command into the message body.

In Twitter this is done by starting the message with the letter "d", then adding the target user's name, then following with your message.

I'm not yet sure how we should do this in ESME.

Message Expiration

It seems to me that senders should have the option of making messages expire after a certain time.

This would be useful, for example, when trying to plan where and when to meet for lunch with a group of fellow employees.

I'm not sure yet how we should do this, but it sounds useful.

Message Retention

Different organizations will want to impose different message-retention policies. For example, there may be a regulatory requirement that all communications (including via micro-messaging) with a given affiliate be archived and available for subpoena for a certain number of years. On the other hand, there may be a private group (say of C-level executives) where to protect the organization in case of a lawsuit you might want all messages over, say, a week old to be deleted.

Certainly this argues for some form of configuration that administrators can make against each pool (from an admin UI).

so now we have the concept of a user-imposed "expiration" and a policy-imposed "deletion" date, both of which become metadata to messages.

Service Calls

A “service call” is a special type of message. Rather than being intended for (typically) public distribution to human readers, a service call is directed at a machine (a server) that will obey any authorized and properly formatted command within its power.

In the Twitter world services like seem to be called “bots”.

These commands (and the servers’ responses) should always be sent privately between the sender and the responding server.

Note this parallel: A private interchange between two humans we label a “conversation”, a private interchange between a human and a machine we label a “service”. In both cases private, named message streams are created for the human user(s) and listed in his/her set of known/available streams.

I believe that we need to make a syntactical differentiation between messages that are addressed to humans and messages that are addressed to machines.

In the human case, we assume that the primary interpreter and respondent is a human, and special embedded substrings are commands for special handling that are handled by ESME. Therefore we can be very flexible on the message format, and just look for a handful of specific substrings.

In the machine case, we are in essence having a command-line discussion with a piece of software, where we will have to communicate parameters and values with great precision.

It may be prudent to say something like: Any ESME message intended to be a service call must begin with *$serviceName*, where the dollar sign tells ESME that the message is a service call, the *serviceName* tells ESME the name of the service, and the rest of the text is the command and parameter string that should be passed to the service verbatim (minus any security checks we may need to make).

I think we would also want to define a standard format for parameters so that users would only have to learn one set of formatting rules (although they’d still have to learn all the commands and parameters of each service).

One format might be:

```
$serviceName  parameterName=`value`
```

Where:

- You can have as many parameter/value pairs as needed.
- Those single-quotes are actually backticks (usually on the upper-left key of a US English keyboard). This allows you to have single and double-quotes in the value.

NAMESPACES

I suggest that, when referred to within a message body:

- Each user name should be preceded by a “@” character.
- Each tag name should be preceded by a “#” character.
- Each service name should be preceded by a “$” character.

This will make it unambiguous as to when we are referring to a human, machine or topic (tag), and will prevent namespace collisions between these three things. (What should we do for group or affilate names?)

ESME COMMANDS

As mentioned before, commands directed to the ESME system may be embedded into message bodys.

This page is mostly to acknowledge that we will need to enumerate the set of commands we wish ESME to recognize and to work out a robust and consistent syntax for them.

COMMANDS I'VE SEEN

For reference, here are some of the commands I've seen in ESME and Twitter so far:

- Send a public message
- Send a public message to <user>
- Send a private message to <user>
- Reply to <user>
- Tag message with <tag>
- Nudge <user>
- Start tracking <word or tag>
- Stop tracking <word or tag>
- Start following <user>
- Stop following <user>
- Block messages from <user>
- Unblock messages from <user>
- Get statistics about me
- Get the most recent post from <user>

Show me profile information for <user>

TEXT MESSAGING COMMANDS

Twitter supports sending and receiving twits via text messaging on your mobile phone, and has some commands relative to that function. I haven't see yet that ESME intends to support this function.

OTHER COMMANDS I'VE THOUGHT OF

Group Management Commands:

- Create private group
- Add group member
- Remove group member
- Add group leader
- Remove group leader

Search Commands:

- Search for <query string>
- Add query as stream

COMMAND LINE VS GUI INTERACTION

I believe that a great many users will need GUI-style UIs for issuing all these commands, and a non-trivial percentage of users will want to issue these commands command-line style as ESME messages. I believe that we should fully support both methods.

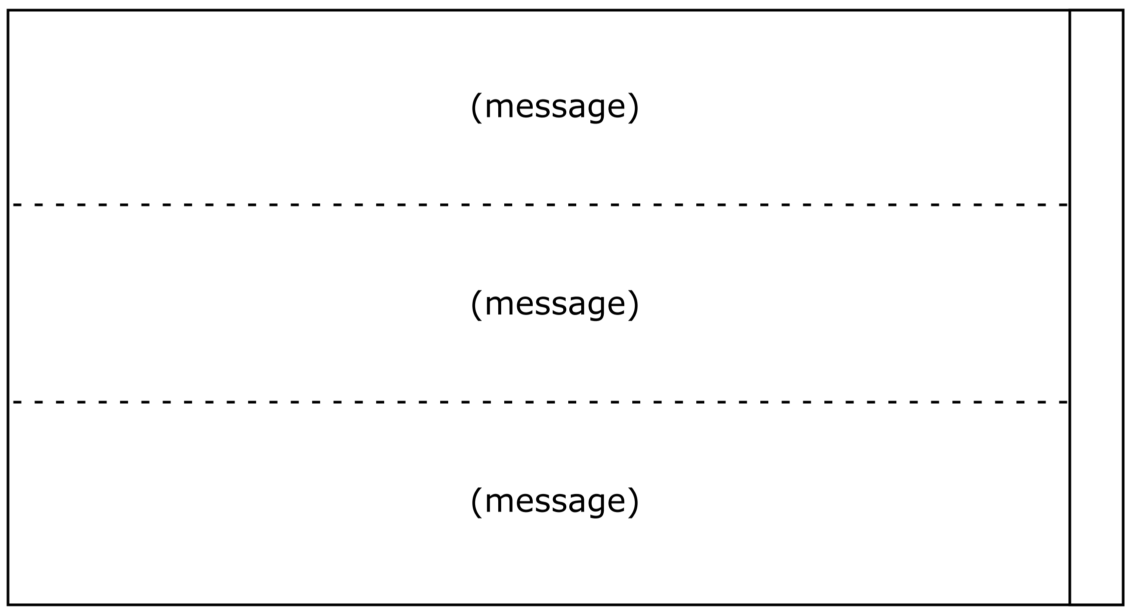
Message Display Format

MESSAGE DISPLAY FORMAT

Now I'd like to make some suggestions about the layout and formatting of the list of messages we display in a message stream.

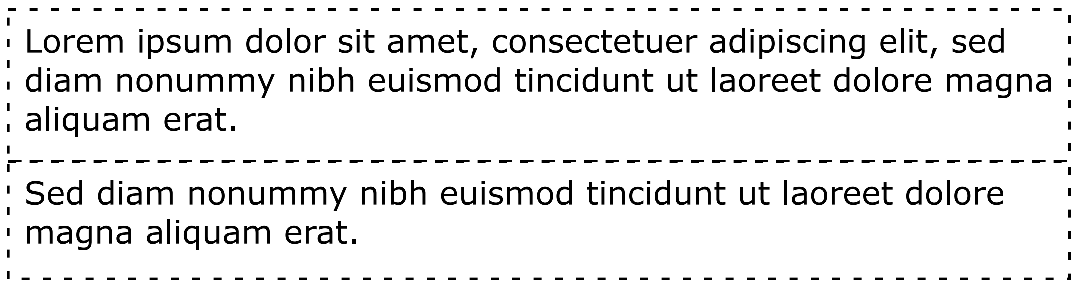
A Stack of Rectangles

I think it's safe to assume that a message stream should be displayed as a vertically scrolling, single-column stack of short, wide rectangular regions. I think that this list should not be allowed to scroll horizontally, and that layout of each message within its rectangle should be flexible such that the height of a message's rectangle can vary to accommodate variations in window width, font size and message length.



Headline Scanning

I believe that the primary cognitive activity and mindset of readers should be one characterized by the phrase "scanning headlines". Thus the priority within each rectangle should be to make the message body (the "headline") fast and easy to identify, encompass and read, and between rectangles to be fast, easy and unerring to skip from one headline to another, either at random or in sequence.

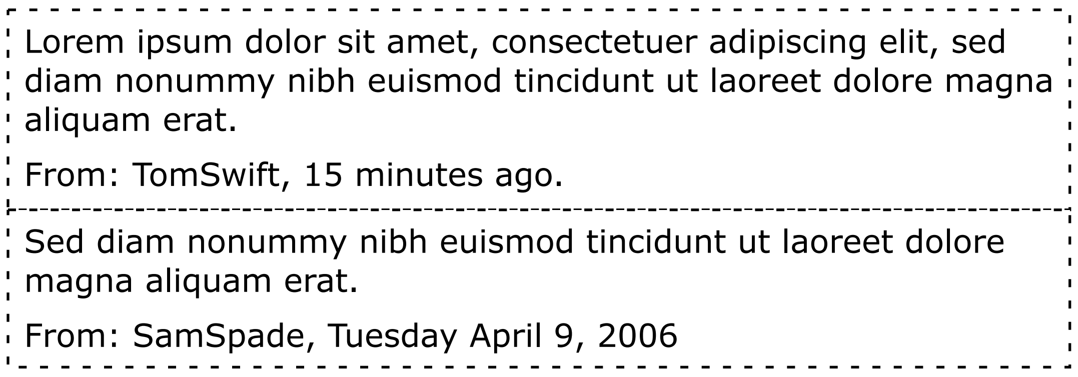


Lorem ipsum dolor sit amet, consectetuer adipiscing elit, sed diam nonummy nibh euismod tincidunt ut laoreet dolore magna aliquam erat.

Sed diam nonummy nibh euismod tincidunt ut laoreet dolore magna aliquam erat.

Metadata

I suggest that metadata, most often being of subsidiary importance, should be placed below the message body. The only metadata that all messages should show are the name of the sender and the time of posting:



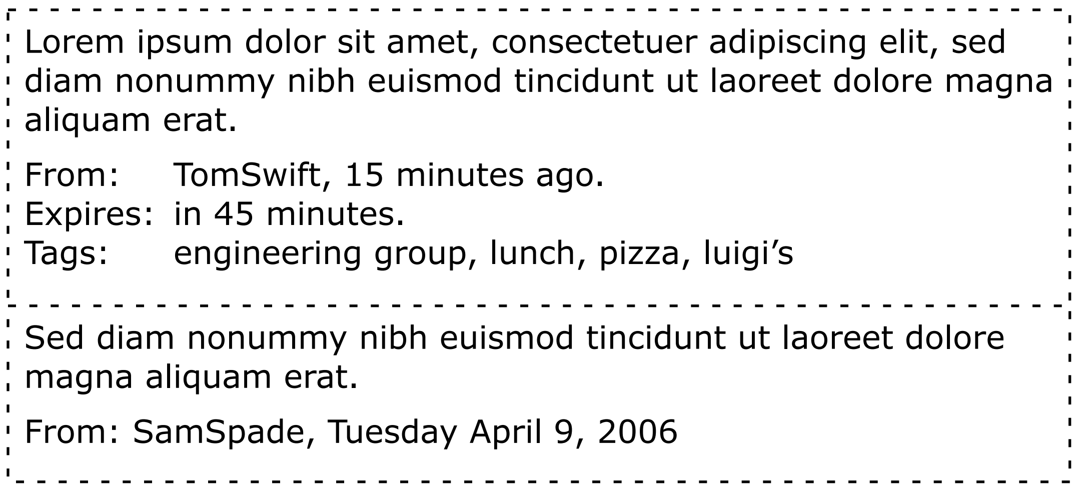
Lorem ipsum dolor sit amet, consectetuer adipiscing elit, sed diam nonummy nibh euismod tincidunt ut laoreet dolore magna aliquam erat.

From: TomSwift, 15 minutes ago.

Sed diam nonummy nibh euismod tincidunt ut laoreet dolore magna aliquam erat.

From: SamSpade, Tuesday April 9, 2006

If there is additional metadata pertaining to a particular message, it should be added below the message body. I suggest making sure that any tags in the metadata are listed last to make it easy for readers to scan for tags of interest:



Lorem ipsum dolor sit amet, consectetuer adipiscing elit, sed diam nonummy nibh euismod tincidunt ut laoreet dolore magna aliquam erat.

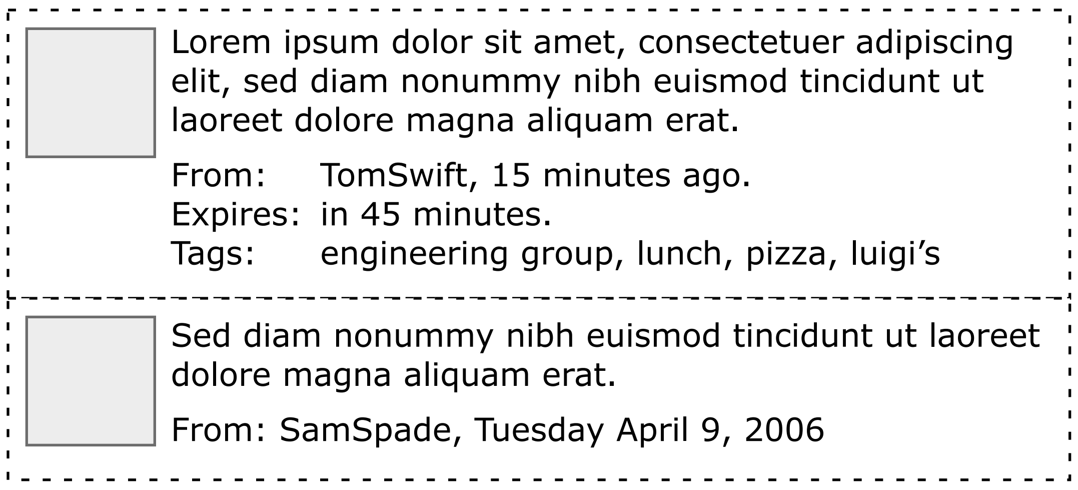
From: TomSwift, 15 minutes ago.
Expires: in 45 minutes.
Tags: engineering group, lunch, pizza, luigi's

Sed diam nonummy nibh euismod tincidunt ut laoreet dolore magna aliquam erat.

From: SamSpade, Tuesday April 9, 2006

Sender's Icon

I agree with the personalizing and visually anchoring effect provided by displaying an icon representing the sender of each message, and that the upper-left corner of each message rectangle is probably the best place for this:



Lorem ipsum dolor sit amet, consectetuer adipiscing elit, sed diam nonummy nibh euismod tincidunt ut laoreet dolore magna aliquam erat.

From: TomSwift, 15 minutes ago.
Expires: in 45 minutes.
Tags: engineering group, lunch, pizza, luigi's

Sed diam nonummy nibh euismod tincidunt ut laoreet dolore magna aliquam erat.

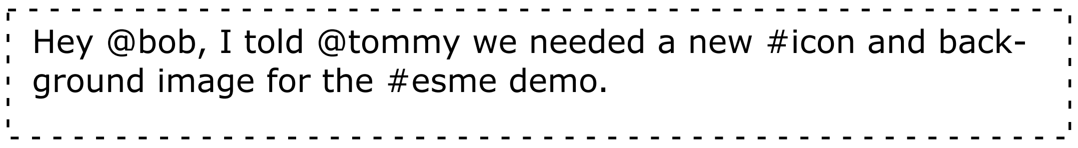
From: SamSpade, Tuesday April 9, 2006

In-Body Special Sub-Strings

Since special substrings (e.g. “@userName” and “#tagName”) can be placed within the message body we have to display them where the sender placed them, but we should also realize that if we highlight them stylistically it’s likely to disrupt the flow of headline scanning.

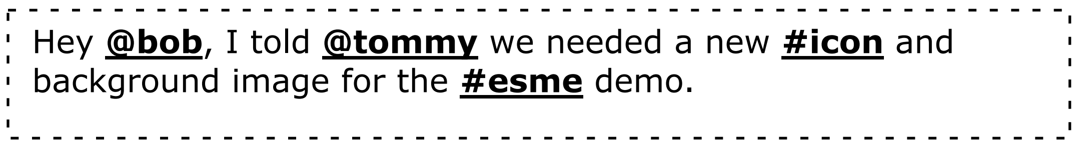
On the other had we’ll want to make these substrings responsive to mouse clicks and should provide users with visual indication of what is clickable. We should probably do this with visual highlighting that happens on mouse-over, rather than having them permanently highlighted.

Thus the preferred appearance for the message body would be something like this:



Hey @bob, I told @tommy we needed a new #icon and background image for the #esme demo.

And a deprecated appearance would be anything like this:



Hey **@bob**, I told **@tommy** we needed a new **#icon** and background image for the **#esme** demo.

Typography and Visual Treatment

The illustrations above are skeletal. Good typography and visual design in the production UI will be essential to good usability.

Message Command Buttons

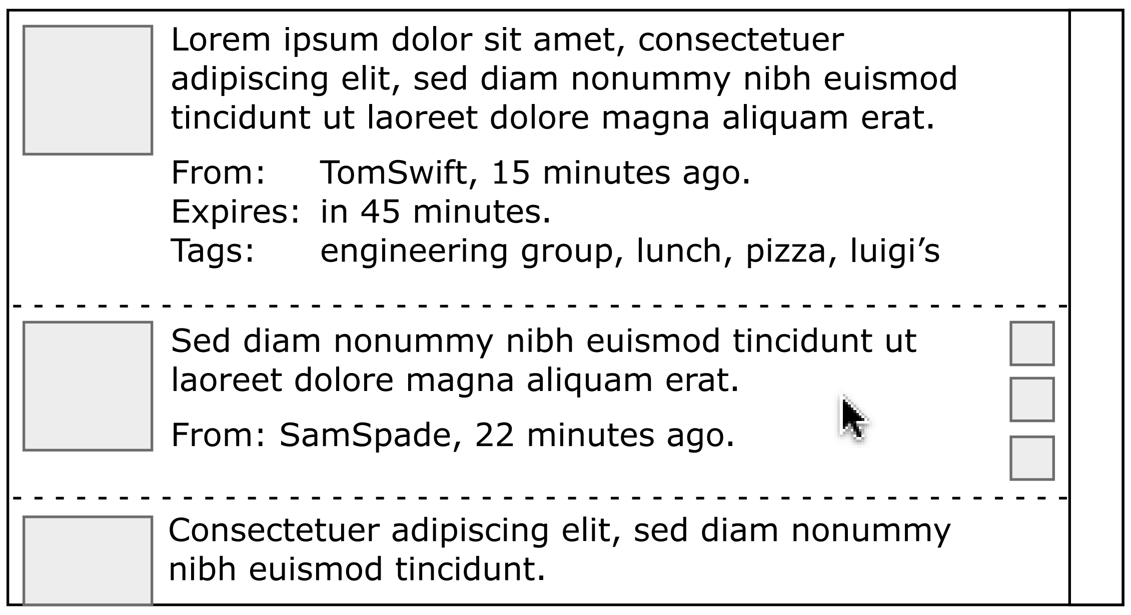
There are certain commands that readers often wish to invoke against messages, such as: Reply, Add to Favorites, etc.

In this design I have placed buttons for these commands in the right sidebar (which is useful if the right sidebar is open), and in the the contextual menus (described in the next section).

But it is also common (e.g. in Twitter, and in the current ESME UIs) to place icon buttons for these commands directly in the message rectangles.

I would advise against cluttering up the message list display by having these icons always-visible and repeated in all message rectangles.

On the other hand it would be fine if they appeared in one message rectangle at a time as you roll the mouse over successive message rectangles. This is illustrated below, with the mouse over the second message:



Lorem ipsum dolor sit amet, consectetuer adipiscing elit, sed diam nonummy nibh euismod tincidunt ut laoreet dolore magna aliquam erat.

From: TomSwift, 15 minutes ago.
Expires: in 45 minutes.
Tags: engineering group, lunch, pizza, luigi’s

Sed diam nonummy nibh euismod tincidunt ut laoreet dolore magna aliquam erat.

From: SamSpade, 22 minutes ago.

Consectetuer adipiscing elit, sed diam nonummy nibh euismod tincidunt.

Contextual Menus in Messages

CONTEXTUAL MESSAGES IN MESSAGES

I think that ESME should have contextual menus that can be displayed when the mouse hovers over various elements of a message.

Contextual menus change their content depending upon what is under the mouse at the time they are invoked, and present a set of commands that use the object under the mouse as the object of the commands.

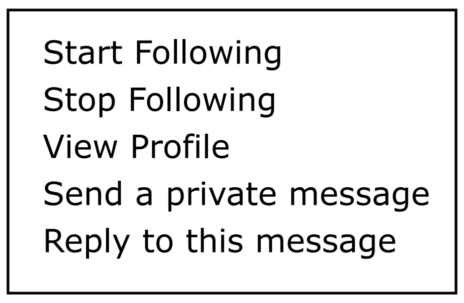
In the Windows operating system contextual menus are typically invoked by pressing the right mouse button (“right-clicking” an item).

In the Macintosh operating system contextual menus are typically invoked by holding down the CONTROL key while pressing the single mouse button (“CONTROL-clicking” an item).

When invoking a contextual menu the item that is the target of the menu’s commands should be visually highlighted.

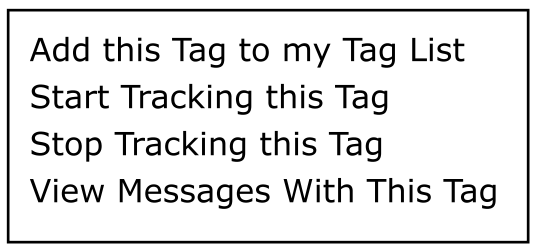
UserName Contextual Menu

When hovering over the sender’s icon, or any *@userName* substring, a contextual menu such as the following should be available:



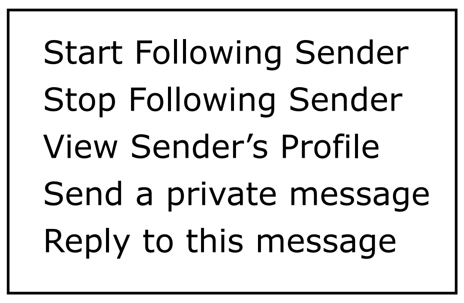
TagName Contextual Menu

When hovering over a *#tagName* substring, either in the message body or in the metadata readout, a contextual menu such as the following should be available:



Message Body Contextual Menu

When hovering over any part of the message body that is not a special sub-string, a contextual menu such as the following should be available:



Other Contextual Menus

There may be other contextual menus that we’ll want to add to this list. For example we might be able to use contextual menus as “parameter helpers” for service calls, etc.

ADDENDUM

Before sharing this document with the general ESME community I asked David Pollak to do a quick read-through and reality check. Here are some comments from the resulting discussion.

Private Pools

Affilates and groups are preliminary concepts I came up with for addressing certain business requirements. It appears that these may simply be variations on the theme of private pools, and that perhaps we can simplify the model by having appropriate, flexible support for private pools and still support these and other business cases.

Federations

We noted that private pools within a single ESME server, and federations of independent ESME servers, both have privacy and access control requirements, and that it would be nice to work out a unified access control model that covers both.

Replies vs Conversations

David and I are both dissatisfied with replies as implemented in Twitter. We would like to support the notion of conversations between two or more users that show connected threads of messages, rather than disjointed, unconnected "replies" that are displayed out of context.

Favorites

David stated his dissatisfaction with Twitter's notion of favorates, and shared his idea of being able to assign colors to messages of interest, and then later to be able to view streams of messages by color. Whether we use "colors" or something else, the essential idea is to be able to mark messages as belonging to one of several categories (rather that the single "favorites" category), and I think this is a good idea.

Services

We acknowledged that interacting with simple ask/answer bots, e.g "@stock AAPL" to ask the price of Apple stock, is essentially the same as asking a human user a question, so maybe these don't have to be handled as human/machine interactions, and maybe they don't have to be private.

Shrunken Window

David really likes the shrunken window concept.

"It's all only Preliminary"

Despite the length and detail of this document it's still just a preliminary set of thoughts and ideas, not a fully thought out design. If the direction is worth pursuing there are clearly concepts that will need to be reworked and refined.

The Creative Commons License

I chose this license because it seemed like the most "open source" -style license suitable for design documents like this, whereas the Apache license seems oriented towards programmatic source code, object code, and documentation thereof (please correct me if I'm wrong). I can change to a different license in future versions if needed.

The Apache Federation website says this license is compatible with the Apache 2.0 license.