Joran Beasley

Use Cases H.W. #3

*****Bold Words are defined in the class descriptions/ class diagrams****

Register Feed Element

Actor(s): User , (server)

Preconditions:

User is logged in.

User has **Role** That Allows Registering a new **Feed Element.**

Goal: Interface with existing data feed (rss , typically , but any parse-able feed in theory)

Steps:

1. User directs browser to administration area of the web interface
2. User navigates via the menu to “Create Feed Element”.
3. User selects from a list of pre-approved feed’s (Twitter , Facebook Events, Google Calendar , etc.)
4. User enters credential information to access feeds.
5. Server sends proper protocol to selected feeds public api interface
6. Two Possible outcomes
    a. Credentials are verified and feed is established.
    b. Credentials fail, User is notified, user may re-try

Alternatives:

(a)

1. User directs browser to administration area of the web interface
2. User navigates via the menu to “Create Feed Element”.
3. User Selects to follow a standard RSS feed at an external URL.
4. User Enters external URL and selects OK.
5. Feed is established. (URL is assumed correct, in the case of incorrect URL the output will be garbage).

(b)

1. User directs browser to administration area of the web interface
2. User navigates via the menu to “Create Feed Element”.
3. User Selects Option to follow custom format feed.
4. User is prompted for external URL.
5. User is prompted for feed structure(<tag> , or [tag],or some variation(inline help will be available)).
6. Feed is established. (URL is assumed correct, in the case of incorrect URL the output will be garbage).

Post-Condition: Feed is Created and Ready for use in a page. However it is not displayed until it is added to a page

Import Group Data From External Source

Actor(s): User , (server)

Preconditions:

User is logged in.

User has **Role** That Allows Importing Members**.**

User has admin rights at a supported 3rd party site (we will try to add any existing group management site that supports OpenAuth , or offers a secure public API.)

Goal: Interface with existing data source(FB,google groups,listserv,external database,or CSV file)

Steps:

1. User directs browser to administration area of the web interface
2. User navigates via the menu to “Import Group Members”.
3. Select supported 3rd party service.
4. Redirect to 3rd party site for Auth Token* using stored procedures to acquire proper permissions.
5. Use Auth Token to query member list.

Alternative:(Import From Database)

1. User directs browser to administration area of the web interface
2. User navigates via the menu to “Import Group Members”.
3. User Selects to import from Live Database
4. Prompt user to enter Connection and Login Credentials.
5. Attempt Database Connection
    a. on Success : Continue to step 6.
    b. on Fail : Notify User and GOTO Step 4.
6. Server Queries DB to determine table structure.
7. IF Field names matching known field names (ie :Username,user_name,username,etc)
    a. on Success: Autofill form fields (See step 8) Goto Step 9.
    b. on Fail : Goto Step 8.
8. Display User form prompting user to match fields found in remote database to member fields in GUS (auto-fill any that we can determine as known).
9. User Confirm Database Mapping.
10. Members Imported, Display Message.

Alternative [Import Database File].

1. User directs browser to administration area of the web interface
2. User navigates via the menu to “Import Group Members”.
3. User Selects to import from **CSV File
4. Prompt user with Select Local File Dialog
5. Parse file (lots of libs to parse csv)
6. IF Field names matching known field names (ie :Username,user_name,username,etc)
    a. on Success: Autofill form fields (See step 8) Goto Step 9.
    b. on Fail : Goto Step 8.
7. Display User form prompting user to match fields found in remote database to member fields in GUS (auto-fill any that we can determine as known).
8. User Confirms Database Mapping.
9. Members Imported, Display Message.

Post-Condition: Group members are added , however many profiles may be partially or completely incomplete. also they will be imported into the main group if you wish to move them to a sub group you must do so through the group management interface of GUS

*Auth Token is supplied through openAuth to grant access to specific areas of social networking , or web 2.0 sites.
**CSV : Comma Separated Value , A format that is widely supported for exports of most DB.

Donate to Group
Actors : Site Visitor , Member

Preconditions:

Visitor has permission to view page (is a member, or page is public)
page being viewed contains a (paypal donation) **Element** (pre built and stored by us)

Goal : Facilitate fundraising for groups in the university context

Steps :

1. User Visits page URL containing the paypal donation **Element**
2. User Clicks the “Donate Now” Button
3. User is redirected to paypal to create the transaction
    a. at this point every thing occurs automatically
4. On a successful donation that is completed a script on our server receives a token from paypal which it then uses to verify via paypal web service that the transaction was successful
    a. on Fail: Do Nothing.
5. we receive confirmation that the donation was successful and the amount of donation
6. we can add it to a total_donation field that could be linked to the group(for internal tracking or a target goal meter **Element**).

Notes:

Gus Has NOTHING to do with the actual money transaction at this point it is only a facilitator it is entirely up to the group leader(officer) to collect the money from paypal

Broadcast Email

Actors: User , Server

Precondition:

User is Logged in
User has a **Role** that allows email broadcast functionality

Goal : send a broadcast message to the entire group or some subset there-of.

Steps :

1. User Navigates to administration area
2. User selects email-broadcast from menu
3. User selects some subset of members of the group (Entire Group,Subgroup, Individuals,Individuals with specific **Roles**).
4. User enters email subject/body text into form elements.
5. User Selects Send.
6. User Reconfirms email contents
7. Email is sent to all users in subset.

Alternative [Automated] :

Precondition:

member has opted in for email notifications of events
event is flagged as broadcast (and a **Contact List** has been designated) by a user with the appropriat **Role**.

1. An Event (Date/Time/Place) Is occuring in the next X hours (X>0)
2. Server daemon broadcasts a templated message to selected contacts.

Post-Condition: Offer opt-out link?