An Intelligent Event Notifying System for Web Service Offered Information

System Specification Documentation
Tanima Dey td8h@virginia.edu
Sirajum Munir sm7hr@virginia.edu

Specification

In the system, the users should be able to provide their preferences of information which will be the different attributes of the event in which they are interested to be notified. These events can be of various categories. Such as, e-mail notification, notification from networking sites, notification from websites, such as, banking, news, blogs, related to hobbies, websites containing favorite songs, videos etc. The system should be able to send alerts to the users in one of the several ways. These can be in the form of pop-up windows, message in the cell phone or a phone call.

In order to specify the requirements of the system, we break down the system into two separate parts, such as, the client server and the application server. The client server interacts with the users and saves the events specified by them into the database. The application server provides a common interface with the server associated with the different web services. It maintains communication with the e-mail server, social networking web server, server for the banking sites etc. It has access to the database where the events for the users are saved. It reads the information from the database and using those, interacts with the other web servers to pull necessary and related information.

We discuss the specification of various components below:

The system should be developed in Java programming language. The source code should be managed by GIT and the main repository should be maintained at github.com. The IDE that should be used is Eclipse as it provides plug-in supports for GIT and GUI designing editor, such as, Jigloo. The client and application server should maintain communication using socket programming provided in the JDK API. This provides a simple and efficient way to exchange information between the servers and clients. The database server should be of any standard, Oracle or MySQL.

The interaction of the users and the client server can be specified in the following parts:

- Signing up: The users who are interested to use the system should be able to sign up for the services. There will some service charge depending on how frequent they want to be notified, such as, notified within 1 minute of the event or 5 minutes. The longer the period is, the cheaper the service charge will be. The signing up process would take users information, e.g., username, password, contact information, address, cell phone number, credit card information for the payment. In the GUI, there would be textfields associated with each field. The username should be unique in the rest of the users username set and the password should be verified by re-entering it. The credit card should be properly authorized and checked for proper balance. After checking and verifying everything, all the information should be entered into the database if those are valid.

- Login: After the signup, the user should be able to login to the system using the username and password as provided during the sign up phase. The system should pull the associated fields from the database and check the password. If the password entered is correct for the username, the authentication is successful and user is allowed enter the system. After the successful login, the GUI window for the user should provide options to add, edit and delete the event specification.
• Adding events: In order to add an event, the GUI should provide enough options to choose from. There should be options for specifying events for e-mail, updates in social network website, news websites, blogs, youtube for audio and video etc. The interface for each type of event should be different and relevant. For example, for adding the description for an e-mail event, there should be textfields to take the input for each related attributes of the events. These should include the e-mail address of the user, i.e., the e-mail account s/he wants to monitor, the password to this account, the e-mail address of the sender from which s/he is interested to get e-mails notification from, the keywords in the subject and body. For adding event description for notification from Twitter, there should be similar textfields, such as, the login information to the account, i.e., username, password, either message or status update of a particular friend, keywords in the message or status updates. The GUI for a blog entry event adding should contain textfields for the address of the blog website, name of the blogger, and a keyword to search in the blogs. All the GUIs for adding event should also contain a drop down list to take input for notification type, how the user would like to be notified, by pop-up window, message or call at the cell phone. After each entry of adding event, the information should be saved in the database in a general format using event description language.

• Editing events: For each of the events added in the previous step, the user should be able to edit those. The edit GUI window should list all the events set by the user. When a particular event is selected to be edited, there should be new window populated with the attributes of that event. Such as, for editing e-mail event specification, there should be textfields populated with the associated information, the username, password to the account, the senders e-mail, keywords to look for in the subject and body field of the e-mail, notification type. The user should be able to edit those and when the final edit button will be pressed, the database should be updated accordingly. Basically, the edit operation is similar to the adding events, the difference is that the textfields will be populated beforehand rather than being blank.

• Deleting events: The GUI for this operation should contain the list of all the events for the user and the selected event should be removed from the database.

• Getting event notification: The application server maintains communication with the corresponding e-mail (Gmail) server or Twitter server and gets event updates from those depending on the event attributes as read from the database. There should be a button in the home page of the users that gets all the event notification from the application server instantly and shows in a pop-up window to the user. If the user is logged into the system and connected to the application server, any specified event updates should be sent to the users in a pop-up if the notification medium is set such. Otherwise, the application server gets the event updates from the web servers and sends notification as specified by the user either a message or call at his/her number.

• Help Menu: There should be detailed instruction in the help section about how to use the system and GUI.