Activity Stream Management for Open Massive Online Courses Requirements & Design Document

Background

Open massive online course platforms provide a task management system for the education community, but mostly without an organised activity streaming system. The aim of this system is to provide a highly interactive activity streaming system for open massive online courses, where users can be kept updated and connected amongst educational peers.

Research Materials

Compilation of activity types	System of research
Course registration	MapleTA
Course material uploads	Blackboard
Tasks: Quizzes, assignments, peer review and peer assessment	WebCMS, Moodle
Submission	WebCMS
Tutorial specific messageboard	Blackboard, WebCMS
Marking	MapleTA, Blackboard

Activity streaming plans

- Notifications for related recipients
- Notification commenting, rating
- Possible integration with Google+, Skype video discussions
- Calendar event page (Moodle, WebCMS)
- Hashtag feature (Twitter) .e.g. a hash tag for an 'Assignment' post can trigger a pop-up menu to insert 'Assignment' attributes such as 'Deadline' and etc
- Communicate and chat directly with other users (Moodle)

Requirements

	Goal Level Requirements
GL-1	The system will provide streaming for a variety of open massive online course activities.
GL-2	The system will provide support to different groups of users.

	Domain Level Requirements
DL-1.1	The system will provide notification of activities to recipients.
DL-1.2	The system will provide the ability to respond to activities.
DL-1.3	The system will provide the ability to hold meetings
DL-2.1	The system will provide identity management support for users.
DL-2.2	The system will provide a role for 'Lecturer', 'Tutor/Mentor', 'Student', and 'Administrator'.

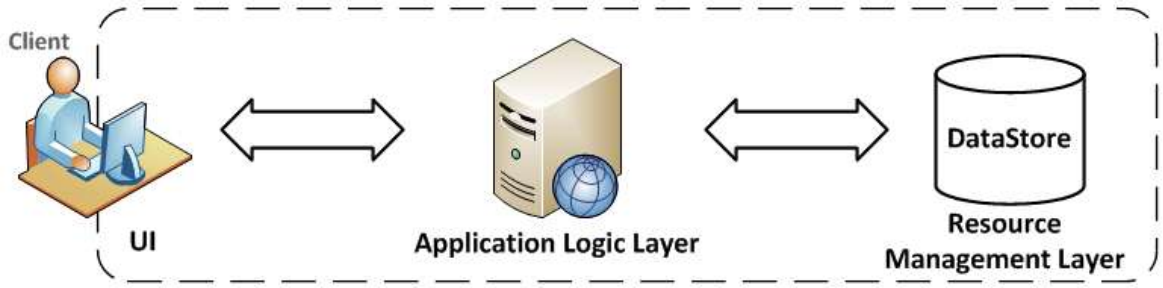
	Product Level Requirements
PD-1.1.1	The system will be able to simulate the broadcast of course materials.
PD-1.1.2	The system will be able to simulate the broadcast of assignments
PD-1.1.3	The system will provide the ability to broadcast messages.
PD-1.2.1	The system will provide the ability to comment on activities.
PD-1.2.2	The system will provide the ability to rate activities.
PD-1.3.1	The system will provide functionality for users to have video/voice/instant messaging discussions through an integrated API.
PD-1.3.2	The system will provide functionality for scheduling meetings related to the activity streamed.
PD-2.1.1	The system will allow new user registration.
PD-2.1.2	The system will allow existing users to reset their password.
PD-2.2.1	The system will allow different categories of users to have different access, posting, group creating and joining privileges.

	Design Level Requirements
DZ-1.1.1.1	The system will send a notification of course material to all users in the course.
DZ-1.1.2.1	The system will send a notification of tasks to all users in the course or a particular group of the course.
DZ-1.1.3.1	The system will send a notification of a message to the recipient. Recipient may be a single user, group, or course.
DZ-1.2.1.1	The system will send a notification of comments/rates/likes to all participating users of the activity.
DZ-1.2.2.1	The system will only allow users to 'Like' the activity.
DZ-1.3.1.1	The system will enable discussions through the integration of Google+ Hangouts API.
DZ-1.3.2.1	The system will use a simple calendar planning plugin to schedule meetings.
DZ-2.1.2.1	Users will receive an email with their new auto-generated password if forgotten.

DZ-1.5.2.2	Users will be able to change their password if they are already logged in.

Design

Architecture



The system will adopt 3-tier architecture with a User Interface Layer, Business Logic Layer, and Resource Management Layer.

The User Interface/Client side will be created by either Google Web Toolkit based off a previous provided by our mentor, Moshe. If the team decides to create the User Interface from scratch, JSP (Java's solution to CGI) will be used.

Application Logic Layer

It consists of a Web Server and business logic (Servlet engine and EJB Server). The Web Server that will be used is tomcat.

The business logic will hold the responsibility of maintaining the functionalities and the correctness of the system. Business logic layer will directly alter the UI once triggered without any bridge such as RestAPI from client or UI layer. The reason why we agreed on implementing direct connection to UI is that building the RestAPIs is not in our focus anymore. Therefore, java and servlets will be heavily used in this project.

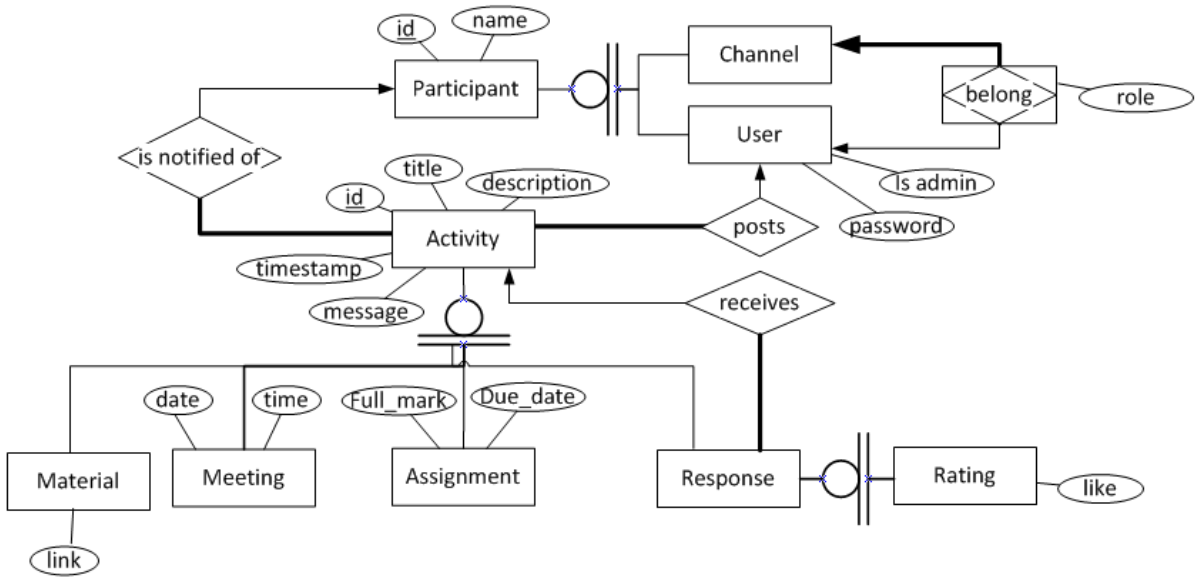
The Business logic structure is defined in detail on UML class Diagram section below.

Resource Management Layer

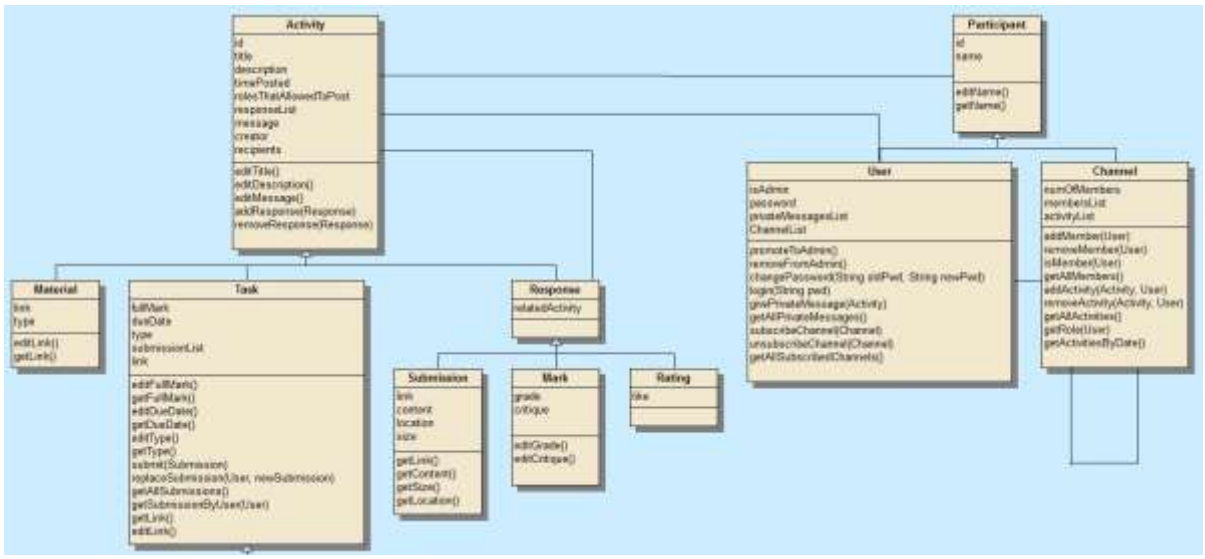
As persistency is one of our main focuses, we use the resource management layer to keep the data and the changes in the system. Resource Management Layer essentially is the database layer of the system that will be accesses by Business Logic Layer. The details of how the database is going to be designed can be seen at the ER-Diagram explanation section below.

To support our needs, we decided to use PostgreSQL as it is supported by university environment.

Entity-Relationship Diagram



UML Class Diagram



Note: Please zoom in to view

Explanation of the Entities and Classes Involved in the System

Participant	Represents a 'User' or 'Channel' in the system.Used for identifying recipients of activities in the system.
User	Represents the end-users of the system. An administrator is determined by the 'is_admin' attribute within the User entity.
Channel	May consist of several Users, or may belong to another Channel. In this context, a Channel represents either a 'Course' or a 'Group' belonging to a 'Course'. A 'User' has a defined role for a particular 'Channel'. The roles a User may have in a channel are as follows : • Lecturer • Mentor • Student The 'role' attribute is determined within the 'belong' composite entity. Channel.addActivity(Activity) is the function used to add an activity to a channel. The activity will be broadcasted to all participating users. Users will be notified. Channel.getRole(User) method will be used to identify the role of a particular user in a channel to determine access privileges.
Activity	Refers to an instance created by a 'User' that notifies a 'Participant'. If the 'Participant' refers to a primary 'Channel', the notification will be sent to the 'Users' belonging to the primary 'Channel' and any subsidiary 'Channels'/Groups. If the 'Participant' is a 'User', a private notification will be sent. The contents of the Activity are located in the 'message' attribute. An activity has several subtypes and has a timestamp.
Material	Material will contain an attribute which corresponds to its location/link.
Assignment	Subtype of 'Activity'. Every 'Assignment' has a 'due date' and 'maximum mark' as attributes.
Response	'Subtype of 'Activity' that also has an association to a predecessor 'Activity'. In this context, it is used for 'Users' to reply to an existing 'Activity'. An 'Activity' may have several 'Responses'. A 'Response' contains text which represents the content/message of the Response. A 'response' has many subtypes.
Rating	Subtype of 'Response'. A 'User' can 'Rate' an 'Activity'. He can like an 'Activity'.

Scenarios

Use Case 1: Broadcast tasks

Trigger	Lecturer intends to broadcast a task.
Preconditions	The user was authenticated by the system and has to be logged in. The current user has to have a role as a 'Lecturer' of the course.
Actors	Lecturer
Event Flow	

	The lecturer may choose one of the course, which he/she's in charge of, or the specified student group in the course. The lecturer then broadcast a task towards the intended recipients.
Postconditions	Task is broadcasted. Notification about the broadcasted task will be sent to the intended recipients.

Use Case 4: Broadcast Course Materials

Trigger	Lecturer or Mentor/Tutor intends to broadcast course material.
Preconditions	The user was authenticated by the system and has to be logged in. The current user has to have a role as a 'Lecturer' or 'Mentor/Tutor' of the course.
Actors	Lecturer or Mentor/Tutor
Event Flow	Lecturer or Mentor/Tutor chooses to upload course material
Postconditions	The course material is broadcasted. Notification about the broadcasted course material will be sent to intended recipients.