CONNActiv

Object Oriented Analysis

Last Revision Date: 12/10/11

1.0 System Overview

2.0 Class Model

- 2.1 Classes
- 2.2 Class Diagram

3.0 Dynamic Model

- 3.1 Scenarios
- 3.2 State Diagrams

4.0 Functional Model

5.0 Revision History

1.0 System Overview

ConnActiv is a social network meant to connect people who participate in physical activities. Its goal is to allow users to meet other people in an area based on a mutual interest in sports, running, hiking, etc. Users will be able to subscribe to different activities based on their interest (e.g. a running “activity”). Each activity feeds into its own public stream containing posts that mention or "tag" the activity, as well as any other relevant updates. The stream is able to be seen in a user's “Home” view if he or she is subscribed to the particular activity. This public stream is intended for users to post what activity they are doing and when. This allows other users to join them if the original user allows invitations to his or her posts.

For example, Jon posts the following message on the site’s “Running” section: “Hey, going out for a run in Oakland at 9AM.” If Jon has allowed it, Stacy may ask to join Jon in running that day.

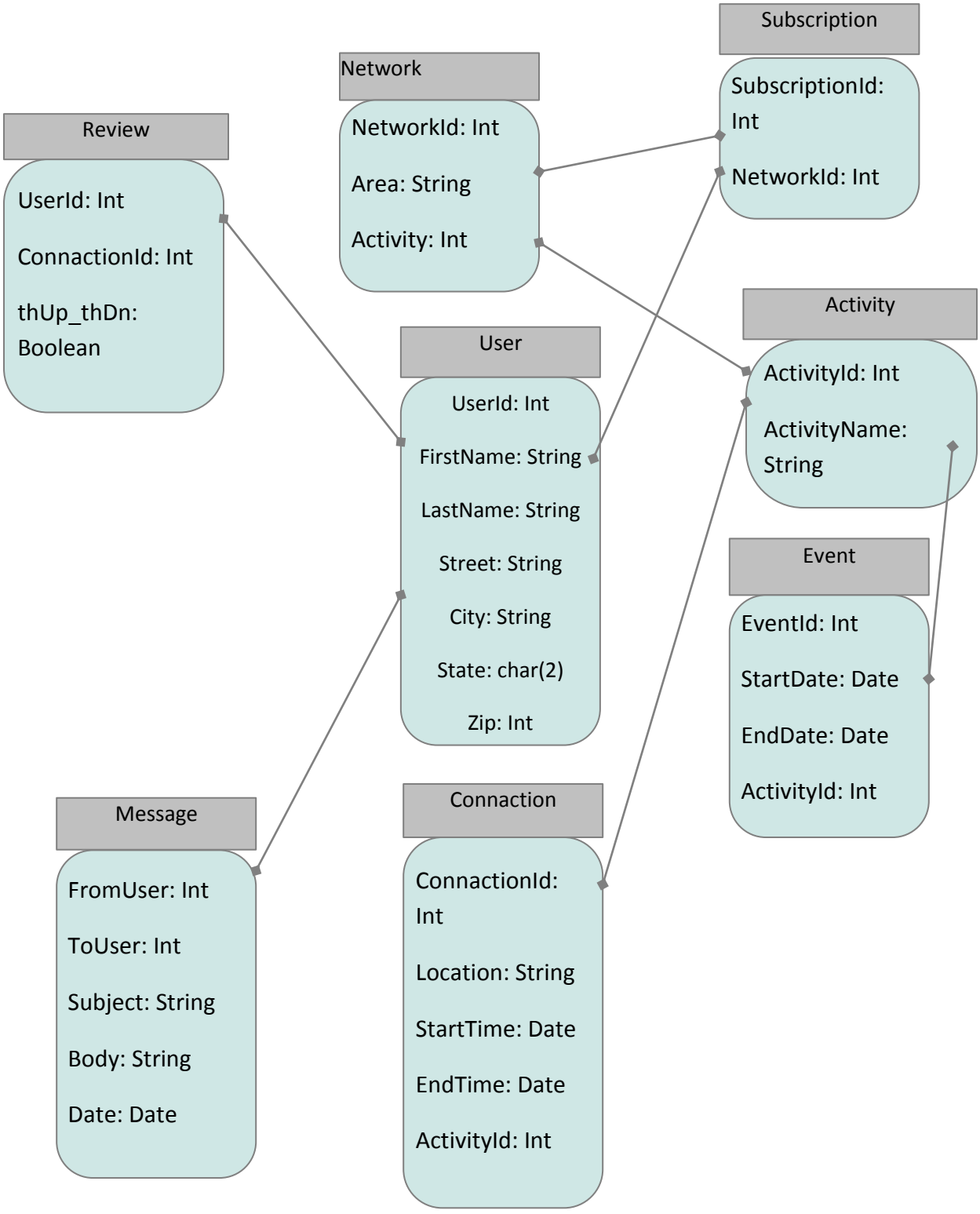
Finally, users may give other users recommendations after they have done an activity together. This gives other users an opportunity to see if the user in question would be a suitable activity partner; however, because this could potentially lead to cyber bullying in the form of malicious reviews users are limited to one review per activity with the person in question.

2.0 Class Model

2.1 Classes

USER(user_id, firstName, lastName, street, city, zip, phone, interests, profile_pic)
NETWORK(networkId, area, activity)
CONNACTION(connactionId, location, startTime, endTime, activityId)
ACTIVITY(activityId, activityName)
REVIEW(userId. connactionId, thUp_thDn)
COMMENT(commentId, fromUser, toUser, comment, commentDate)
MESSAGE(fromUser, toUser, subject, body, date)
SUBSCRIPTION(subscriptionId, networkId, userId)
EVENT(eventId, startDate, endDate, activityId, location, recurrence)
PICTURE(pictureId, userId, path)

2.2 Class Diagram



3.0 Dynamic Model

3.1 Scenarios

Scenario 1 – Post an Activity

- User selects what Activity Stream to post to
- User selects what kind of Activity
- User can leave a small message
- User selects what time
- User specifies difficulty levels
- User's Activity gets posted to selected stream

Scenario 2 – Ask To Join an Activity

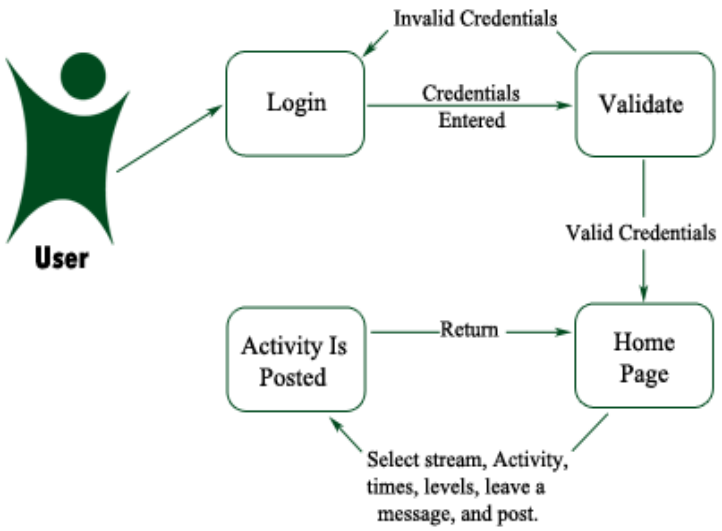
- User finds an Activity they are interested in
- User clicks ask to join this Activity
- User who created the Activity gets a notification that someone wants to join
- The user who created the Activity decides whether to allow the user interested in joining to join or not
- User who asked to join receives notification on the creator's decision

Scenario 3 – Leave Feedback

- Two or more users do an Activity together
- Each user can then give a thumps up or a thumbs down on the other users
- Each user can write a small comment about that activity on the other users' profile

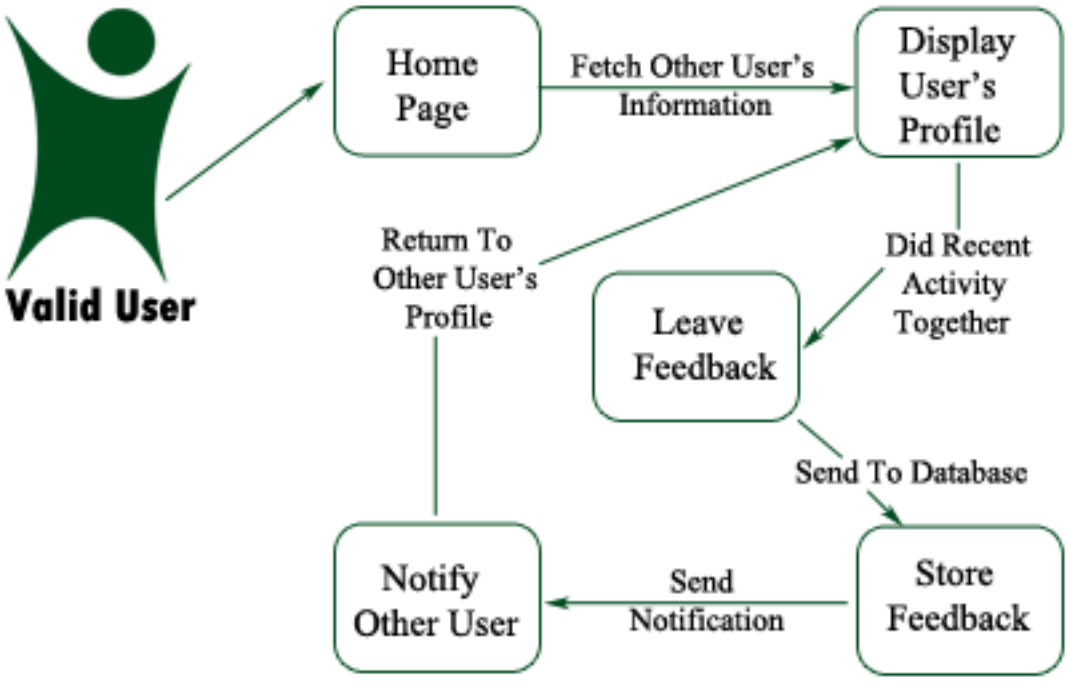
3.2 State Diagrams

Scenario 1 -

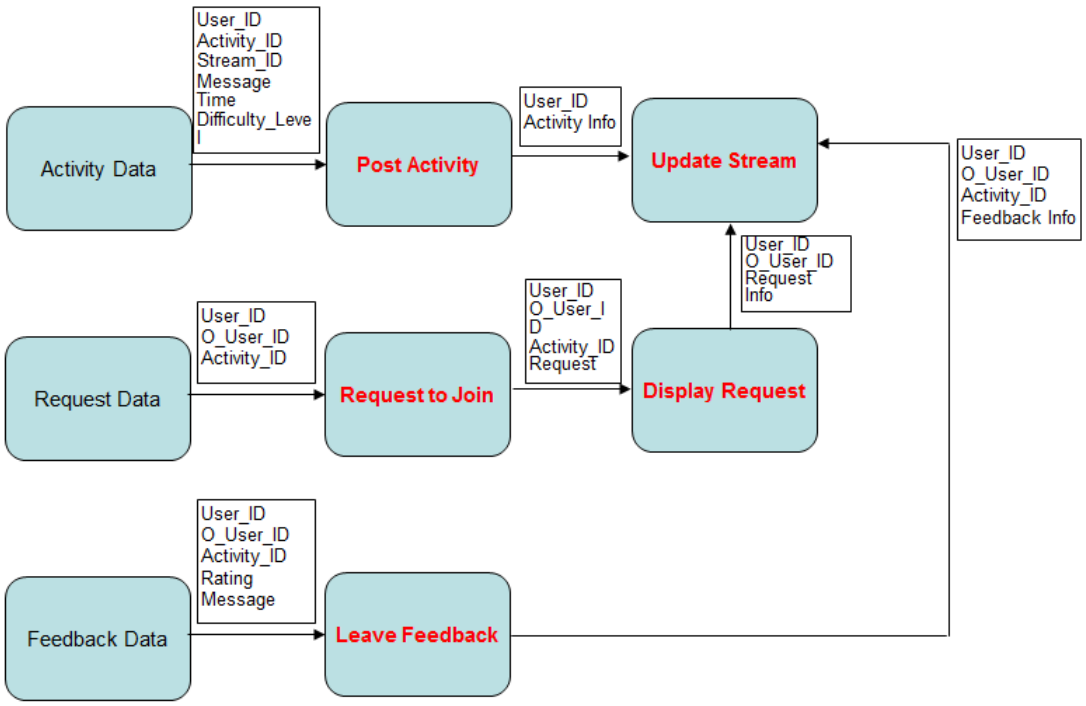


Scenario 2 –

Scenario 3 –


Valid User
Home Page
Fetch Other User's Information
Display User's Profile
Return To Other User's Profile
Did Recent Activity Together
Leave Feedback
Send To Database
Notify Other User
Send Notification
Store Feedback

4.0 Functional Model



5.0 Revision History

Person	Date Modified	Section	Description
Rob	11/2	3.1, 3.2	First draft
Ray	11/2	1, 4	First draft
Vince	11/3	/	Proofread and collated
Dave	11/3	2.1, 2.2	First draft
Vince	12/10	2.2	Removed comments module