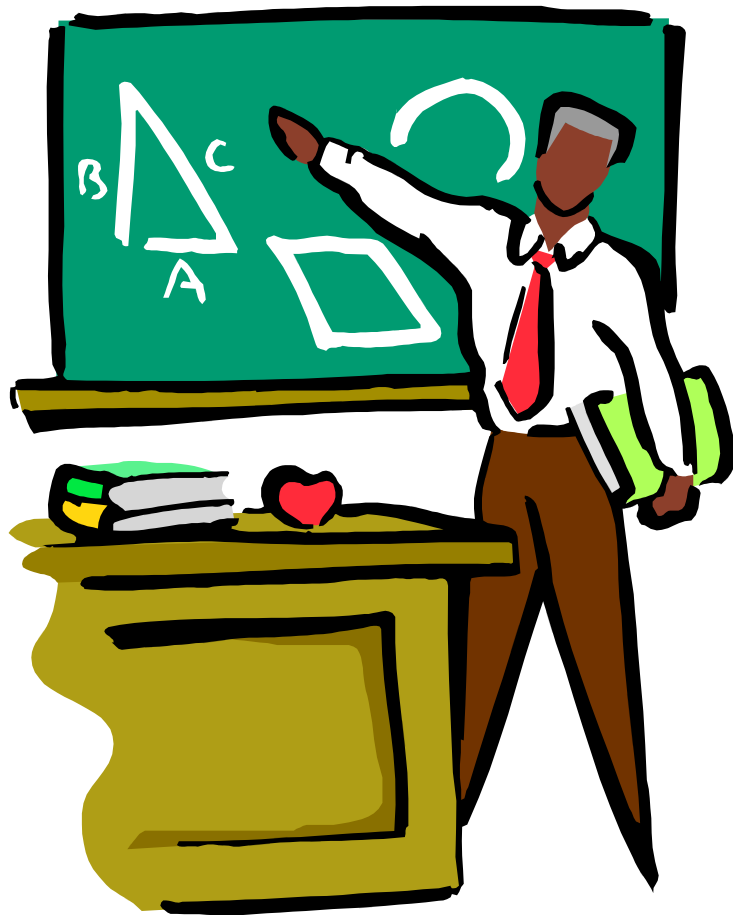




Project Specification
Social Application:
Create a conference between social friends
that have a common interest



Social Application: Create a conference between social friends that have a common interest



- 1 - Overview and architecture
- 2 - Assumptions
- 3 - Breaking it into phases
- 4 - What students should do



Overview and Architecture

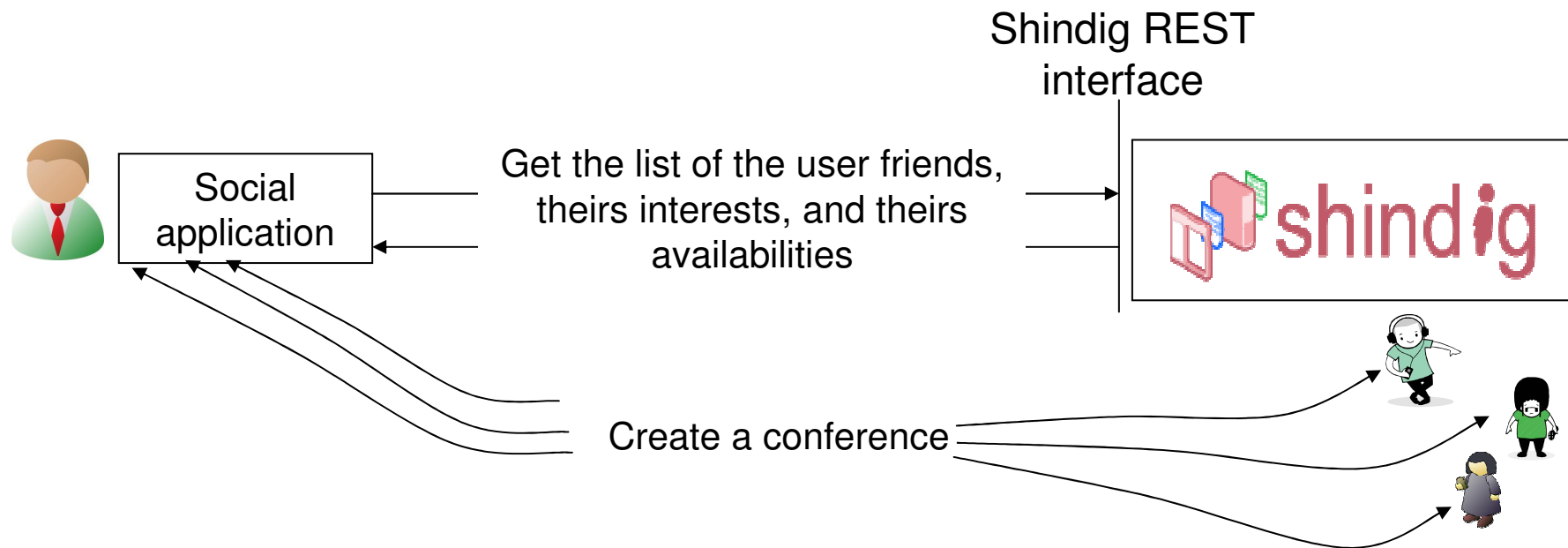
Overview

- A social application that allows an end-user to:
 1. Get and display the list of his/her social friends
 2. Get and display the interests of each of the friends, along with his/her availability (i.e. when he/she is available to participate in a discussion about one or more of his/her interests)
 3. Select a set of friends that:
 - Have a common interest that the end-user is interested in discussing (e.g. the INSE 7110 project)
 - Are available at the same time
 4. Create a conference among the selected friends

Overview and Architecture

Architecture

- Use:
 - Shindig as the OpenSocial container/server
 - Representational State Transfer (REST) APIs for communication between the social application and the Shindig server
 - SIP to create the conference



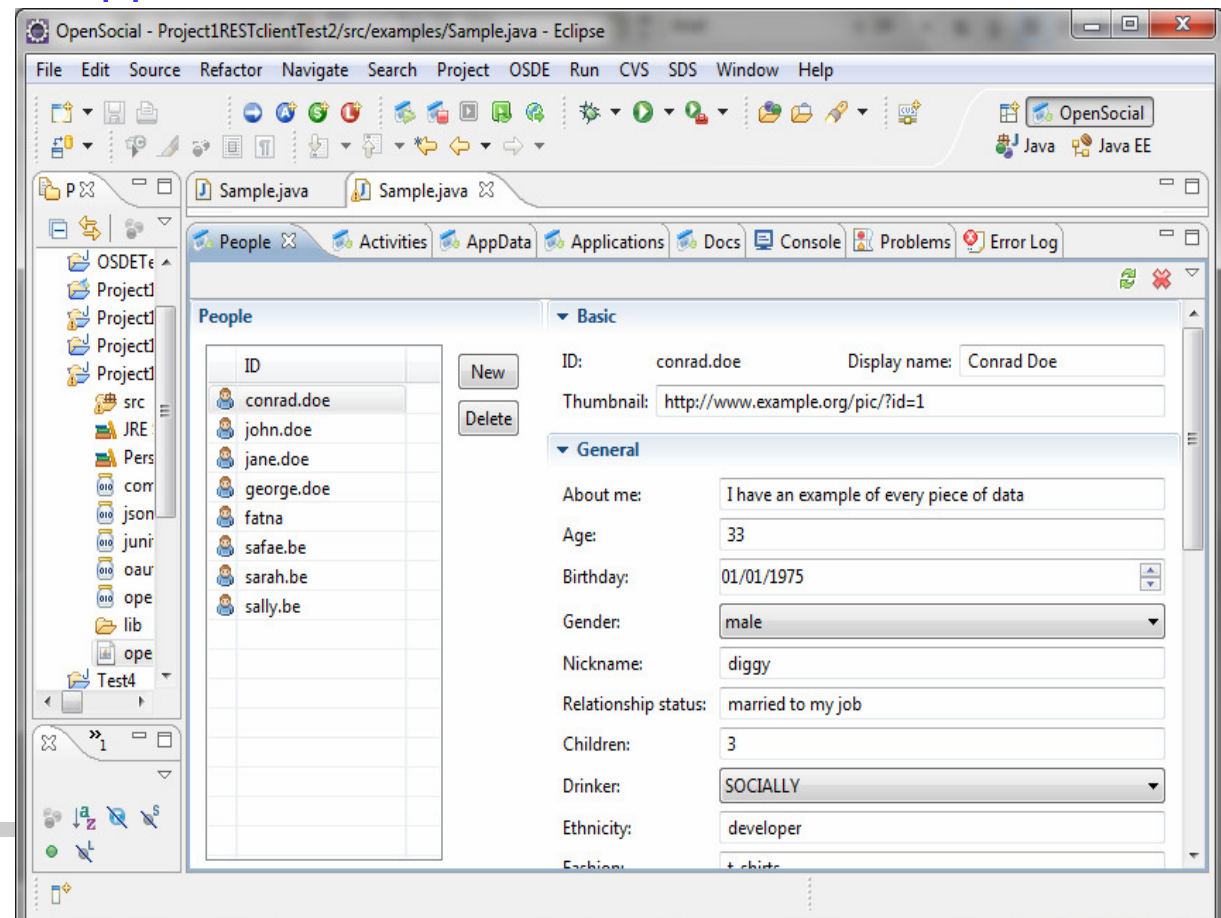
Assumptions

The following can be done via the default Shindig interface, installed with the OpenSocial Development Environment (OSDE): an Eclipse Plug-in that allows the development and testing of OpenSocial applications

- Add a user friends
- Specify a user interests and availability

An existing SIP client can be used as the conferencing client

- E.g. Xlite





Breaking it into phases

Phase I: Get the list of friends and their interests and availabilities

1. Via a user interface, the end-user willing to discuss about a given topic specifies:
 - The topic he is interested in
 - The time slot when he would like to discuss
2. The social application sends a REST request to the Shindig server, in order to get the list of the user friends and their interests and availabilities
 - This may be done using one or more REST requests
3. The social application displays to the user the list of his friends that have a similar interest, along with their availabilities



Breaking it into phases

Phase II: Create a conference to discuss about the chosen topic

1. Via the user interface, the user selects the list of friends he would like to discuss with, and asks for conference creation
2. The social application creates a new conference among the selected users
 - The invited users can either accept or reject the invitation to join the conference
 - A user can quit the conference at any time
3. The users discuss about their common interest
4. The conference is ended
 - Because the discussion is terminated and then the user that initiates the conference decides to end it
 - The user that initiated the conference quits
 - The number of current users in the conference is equal to 1 (all the users quit except the initiator)



What students should do

Scope

- Design and implement the social application, including the two phases



What students should do

Groups

- The project should ideally be done in groups of 4
- However groups of small size are allowed:
 - These groups should work on the whole project as specified, but will have bonus points.
 - Group of 3: 5 points bonus
 - Group of 2: 10 points bonus
 - Group of 1 : 15 points bonus
- Note: The project will be graded nil if none of the phases could be demonstrated.



What students should do

Output

- Short presentation
- Live demo
- Project report
 - Should include
 - The design of the social application
 - Phase one:
 - » Discuss the resources used (in term of REST), their names (URIs) and the methods you used to access them
 - » Provide the sequence diagram and discuss the communication between the social application and the Shindig server (i.e. the communication behind the API you used)
 - Phase two:
 - » Provide the sequence diagrams and discuss the communication for the conference creation and termination
 - Present and describe the user interface
 - Implementation
 - Who did what?
 - Note: The report should be 20 pages max