**Team Unit #2**

Overview:

Our app will function similarly to sites like StackOverflow and Yahoo! Answers in that users can sign up and then ask questions or respond to these questions. Answers are voted for by registered users that then structure the responses based upon user feedback. The goal of the app is to provide a centralized location to ask questions of users that may have experience.

This incorporates crowd sourcing by providing the framework for asking questions, and then letting registered users create the questions and run the responses. These are then up/downvoted, and this determines which questions are displayed more.

Feature List:

* + Login page that offers log in or sign up
  + User registration page
  + User profile page with image upload
  + Ability to ask questions
  + Ability to answer questions
  + Ability to vote questions up or down
  + Search feature
  + Admin profiles with more access than standard user
  + Facebook web API integration

Database Schema:

* + In the User table, we’ll store first and last names, usernames, passwords, name of uploaded photo.
  + In the Question table, we’ll store the id of the user that posted the question, the question, the date posted, and any associated tags (i.e. C++, Java, PHP, etc.), the number of likes, and the number of dislikes.
  + The the Responses table, we’ll store the id of the user that posted the response, the id of the question associated with it, the response, the date posted, the number of likes, and the number of dislikes.
  + The User\_Question table will be a join table that stores the id of the user, the id of the question post, and a column that says if they liked or disliked the post.
  + The User\_Response table will be a join table that stores the id of the user, the id of the response post, and a column that says if they liked or disliked the post.