Project Review

Overall the development of this prototype went smoothly. Deadlines were met, and there was little to no conflict within the team. There wasn't really a solid leader figure.So a downfall we had was that the deadlines were made up as the project moved forward. This led to the "perfecting" stage occurring quite late. So learning from this the team as a whole should work to enforce earlier deadlines. That way there is more "free" time to perfect the project.

Current State:
Currently the project has met all of the requirements specified in the beginning. All functionality stated(as part of the prototype) is currently working except for the deletion of module instances. E.g There isn't an ability to delete posts, topics, users, and groups.

Issues/Concerns:
Refer to the HLD UML diagram for a general layout of the current prototype.
From a design view, it may have not been the best. While each module was separated from each other, each of them was dependant on the database.Within each module, specific information was obtained and shoved into the database through the database interface. These methods were highly specific , and only worked in the given scenario.Each module's code consisted entirely of methods. There was basically no OOP involved in the design.

So the in the current state, it is entirely dependant upon that specific database interface, and sqlite3. Which becomes an issue for the code to work independently. So if something needed to be changed, huge amounts of code would have to be re-written.

Future suggestions:
First it should be noted that an attempt was made to implement the below plan. Having classes with attributes and utilizing Object Relational Mapping was attempted. However it kept giving errors, so the easier path towards having a fully functional project was taken.

Each module should have been turned into a class, which would have instances of the database as attributes of the class. The objects would be mapped to the appropriate database table. So Object Relational Mapping would be highly useful. So instead of having every module eventually call back to the database interface to interact with the database, modules would be able to get and send info with other modules.(*Refer to the attached future prototype design for the layout.)*

Once that is fixed, improving upon current features/creating new ones would be the next step. Many aspects of the design are quite restrictive. For example deletion of groups and posts is not available. There are minor inconveniences in the the basic design. Making the features more user friendly would be the next consideration. Then adding more functional features would be considered after that. The new HLD would be presumably correct so all future prototypes should maintain the separation of the database.