Team C Iteration 1

Iteration 1 mark 65 / 100

High-level design

Over all I'm a a bit concerned about the state of the design. I don't see any class design for the client or server.

Components, package, classes

This seems to be missing. If I've missed something, let me know.

Specification of internal interfaces (APIs, WebAPIs, sequence diagrams)

Spec of WebAPI is a good start.

- I would be more explicit about the results. You say results come in json formatted objects, but not what the fields of these objects are or what their types are. Consider getDirectory; I would imagine that it returns an object with a field that contains an array of objects that have fields for (at least) file names and fileIDs; but this is just what I imagine; it is not enough that I could sit down and write a web client.
- Don't forget to deal with error cases. What if an id is invalid.
- Missing methods? I don't see anything related to solutions, compilation, execution. Walk through your use cases and ensure that the web api is adequate to cover all use-cases.
- Possibly missing interface: I'm not clear on how students create, delete, and rename files. Do they do this by altering the directory on the client and then use saveDirectory?

Database Schemata

I'm wondering whether there are missing entities. In your lexicon you talk about "solutions" as a category of "works". In the DB you have "Work". But, I don't see how files are associated with works or how users are associated with works. Also the webapi deals with assignments and directories; neither of these entities seem to be reflected in the db schema.

Other comments

I don't only see logs for Phil, Dean, and Tim.

Test plan is very good.

Source code (optional)

Notes from Zizui

1. (team c) Database diagram can be improved to make it easier to understand. Also, database structure tables can be used here.

2. (team c) Using string as primary key is strongly discouraged. (File in Dbtablerelations.pdf)