

---

# Team A Iteration 1

Good job overall. Keep the design documentation up to date as you continue the project.

90 / 100

---

## High-level design

### Components, package, classes

This is good and well thought out. One thing that is less clear to me is how objects of the classes shown in the Software Architecture diagram are serialized to the DB.

Also the interaction with Spark is not so clear. The CodeCloudMain is shown as being associated to "All Other Classes". This seems unlikely. It also means that you have one huge class that deals with a lot of different kinds of requests; could this be modularized.

I do like that the bulk of your system has no dependence of the web framework.

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

Very good collection of sequence diagrams. The more important use cases are addressed.

Good use of JavaDoc.

WebAPI is not linked in to index.

WebAPI is incomplete, continue to work on it.

### Database Schemata

I'm not sure that all your classes that need to be persisted are represented in the schema. Comments, test reports, grades, for example.

---

## Source code (optional)

---

## Other comments

You need to design a strategy for coping with concurrent requests.

Since you are using the UNIX file system for file storage, be very careful of paths containing "..", as these could lead to security

holes.

---

## Notes from Zizui

1. (team a) Not sure the connection between FileManager and UserFile. Why permission is based on FileManager, not File, while user still directly owns files?
2. (team a) Class name should always be singular ("Comments" class).
3. (team a) It is better to wrap `java.lang.Process*` to increase portability. e.g. Switching to other methods for process creation.
4. (team a) In UC01 sequential diagram, things like "Display Login screen" should be returned to user, not loop back to GUI.
5. (team a) "term" is missing in database schema. Also, using string as foreign key is strongly discouraged.
6. (team a) It is better to create a ID field in "Users" table.
7. (team a) It is better to return valid JSON object in all cases, including non HTTP 200 responses.
8. (team a) For security reason, password should not be transmitted in cleartext. At least, sending MD5 encrypted password. (`setNewPassword`).
9. (team a) I am confused by "runApplications" in WebAPI. How can the response "is JSON encoded entity containing GUI application or command line execution"?
10. (team a) "Website Prototype" cannot be accessed.