



## Sprint 2 Review

**Team:** **TEAM 203**

**TA:** **Siddhesh Salgaonkar**

**Review Date:** **11/17/2018**

### Base Expectations

1. Getting the legacy code base running if that's still an issue.  
*Comments:*  
*Up and running.*
2. Adding a notion of user and groups of users to the system.
  - a. The entities should be persistent. It's the team's call what to use for persistence.
  - b. Provide basic user/group management: Create, Read, Update, and Delete capabilities.*Comments: Completed*
3. Directing messages to individuals. Directing messages to groups. Replies to messages sent to a group go to all other members of the group.  
*Comments: Completed*
4. The system is deployed to a cloud environment such as Amazon AWS or Mass OpenCloud. You are encouraged to find a free service. You must provide administrator access to the project executives.  
*Comments: Completed in previous sprint. Should be continuously deployed from master.*
5. (small): team is using smart commits in git.  
*Comments: Completed in previous sprint.*
6. (small): Jenkins should inform the team of failure either in Slack or email.  
*Comments: Completed in previous sprint.*
7. (small): GitHub should inform the team of PRs via slack or email.  
*Comments: Completed in previous sprint.*

## Stretches

8. Stretch (medium): Adding login with user name and password. The storage must encrypt passwords.  
*Comments: +3*
9. Stretch (medium to large): Add message persistence to messages sent to individuals and groups.  
*Comments: +5. You can add logic to display messages for a particular group.*
10. Stretch (large): add MIME types to messages. MIME [Multi-purpose Internet Mail Extensions] are a standard way of identifying the type of a file. It's used in messaging systems, so senders can transfer multiple types of files without needing special mechanisms, and clients can first identify and then deal with these files as appropriate.  
*Comments: +2. Code is in place. But couldn't verify the changes on AWS. Receiver should actually get the file.*

Comments:

*Good job overall.*

*Logger not yet implemented.*

*Trivial code smells should be removed i.e. std outs, TO-DOS, long methods*

*Database connection URL etc. should be maintained via properties.*

*Nice use of colors and message history on login +1*

*Update pom.xml for fat jar creation (discussed in the meeting as well, changes are not pushed ahead)*

*Next time there will be penalties for sloppy code practices. Master should be polished.*

**Team Score: 95**