



# **Slick Communications**

**Developed by: Team 105**

Chirag Shivnani

Prem Shah

Sravanthi Kethireddy

Tymofii Kryvtsun



# **Slick Communications**

## **Extending a communication server in Java**

### **Team members**

Chirag Shivnani

Prem Shah

Sravanthi Kethireddy

Tymofii Kryvtsun



Northeastern University  
CS 5500 – Managing Software Development  
Fall, 2018 – Professor Michael Weintraub

Contents

1. Overview .....	4
2. Overview of Result .....	5
3. Development Process.....	8
4. Retrospective.....	10



## Overview

The objective of this application is to extend the communication server, so it delivers a slick set of communication services. Example of communication services include user to user messaging, user to group messaging and various message related functionalities.

The following are the functionalities the system should possess:

- Create new users, groups, update and delete them accordingly
- Ability to let the users join and leave a group
- Allow users to send direct messages to individuals and groups
- Everyone in the group must be able to view the messages sent as a response in the group
- There must be a message persistence for users and groups
- System should be capable of queuing messages. An offline user must receive the messages sent to them when they are back online
- It must follow CALEA rules and regulations
- System should also allow to duplicate a message targeting a specific user or group. Duplicated messages must be forwarded to agencies who have a subpoena for the communications of users.
- There should be a parent control feature on the system
- System should allow admin to search messages based on specific attributes



## Overview of Result

The developed application allows client to create, read, delete and update user or group. User can either be a normal user, admin or agency. Only admin can assign specific role(user/admin/agency) to all users, by default all users are normal users. User has privilege to either join or leave any group from the system and if group does not exist then create new group. System is encrypting password using MD5 hashing and persisting it with other user details for all user.

User can directly communicate with other specific user, all users of a specific group or broadcast message to all users. User can send normal text message or MIME type message.

The system has ability to queue messages sent to users who are not online and when the user comes online, deliver the message in order in which they were sent. System also allows user to recall any message send by it either to other user or to group. All messages are wrapped and stored in the system with sender's and receiver's IP address.

System allows agency to wiretap any user or group for specific number of days. Agency must request admin for wiretap privileges, if approved by admin, agency will receive all the messages send or delivered to a user or a group who is wiretapped and this is done without anyone involved in the original communications being able to determine their communications are being sent to an agency. The wiretap is only good for specific period.

System also has a parent control feature, where any message having inappropriate content is flagged and delivered with altered content. Examples of inappropriate content could be vulgarities, phrases that suggest violence, sexual imposition, bigotries of many sorts.

Email and logging feature is implemented to keep track of errors and logs generated for system.



## Backlog Statistics

Sprint Number	Number of Issues Assigned	Number of Issues Completed
Sprint 1(Oct 21 - Nov 4)	18	18
Sprint 2(Nov 5 - Nov 18)	15	15
Sprint 3(Nov 19 - Dec 2)	22	22

MSD105 board

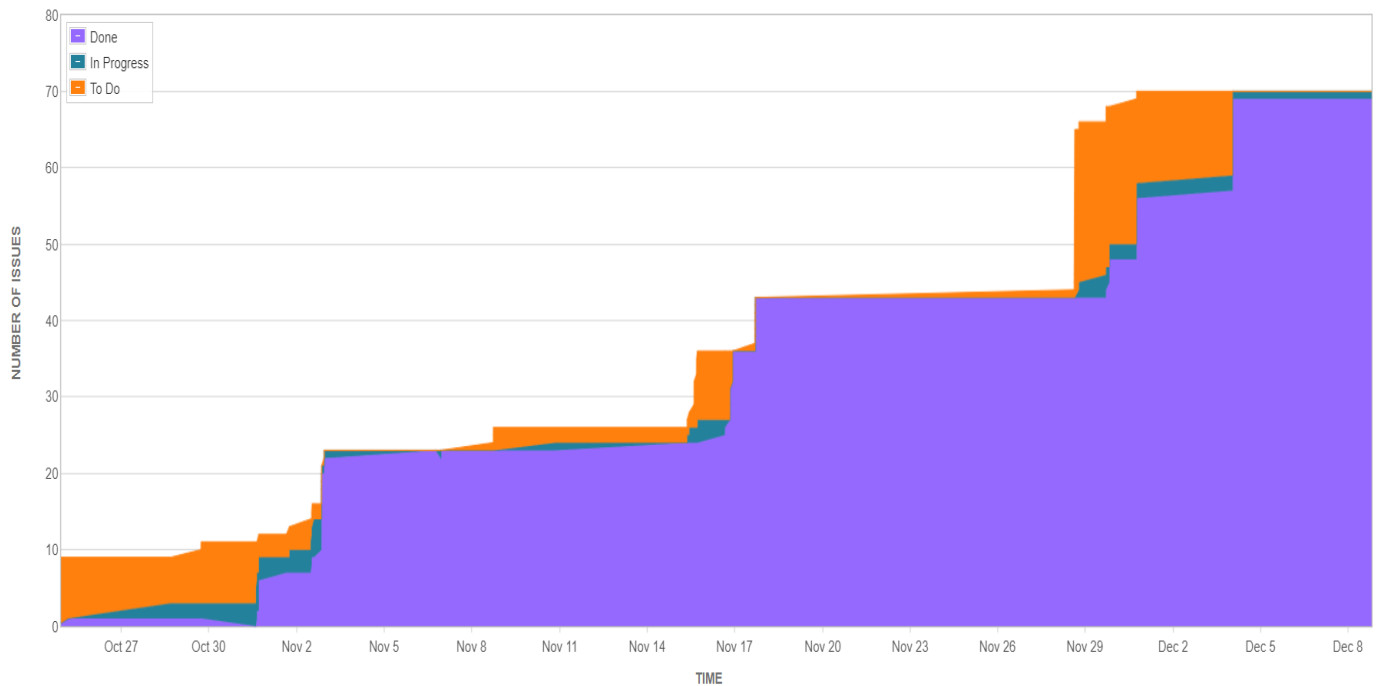
Cumulative Flow Diagram [Switch report](#)

Board



24/Oct/18 to 8/Dec/18 (All Time)

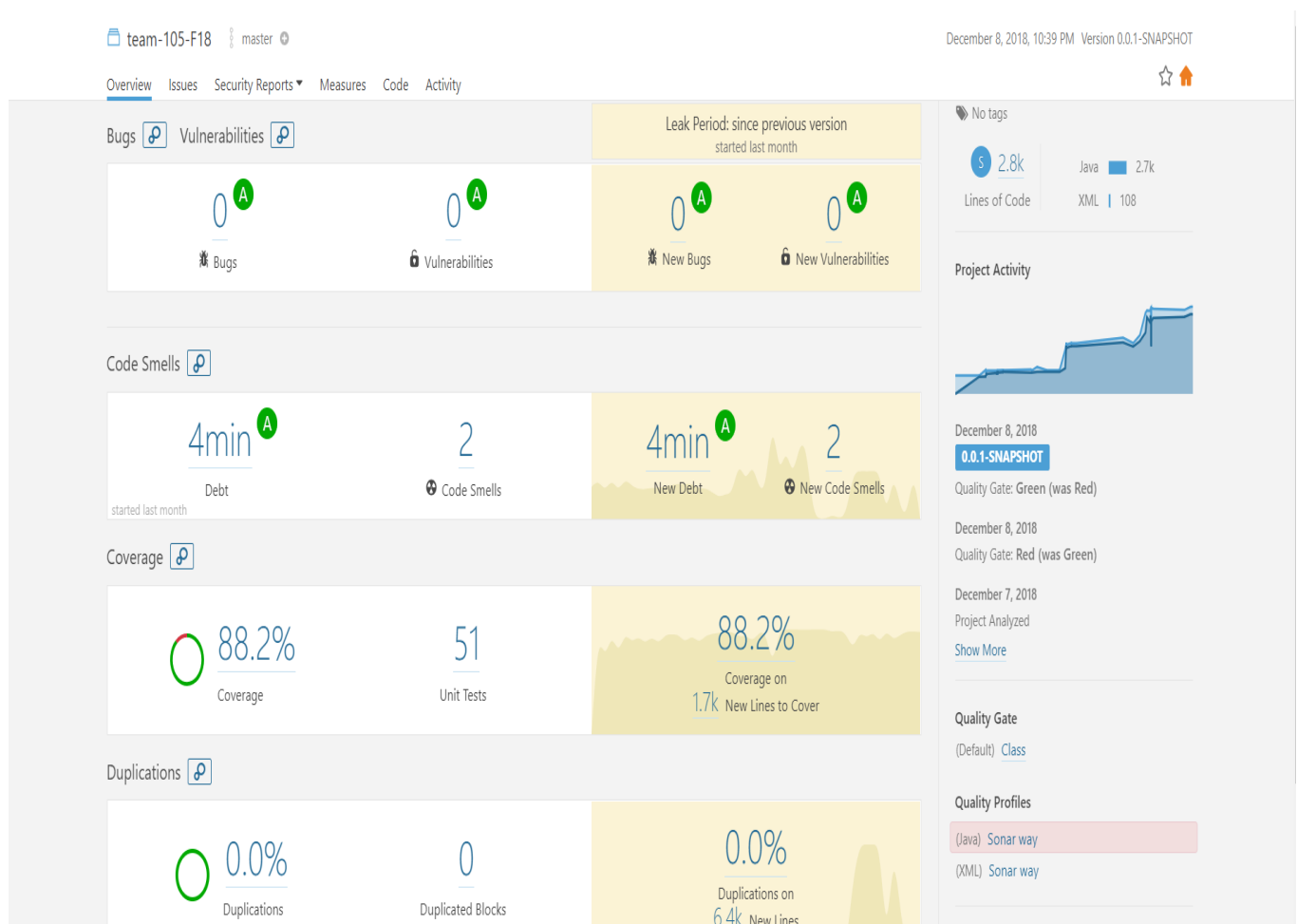
[Refine report](#)





## Quality claim supported by testing data

We have used test driven approach for software development. Functional testing is done to validate system functionality. Automated testing is done using Junit. Incorporated Jenkins for continuous integration. Code quality is achieved through SonarLint as well as SonarQube analysis.





## Development Process

### Sprint 1

#### Environment setup:

1. Setup Git Repository and restricted the master branch. The work done can be pushed to the master only via pull-requests.
2. The project uses maven for builds, the project uses junit5 for unit tests.
3. The project is using at least version 1.8 of Java.
4. Work is being managed in Jira.
5. Used smart commit in git.
6. Setup Jenkins to run on all pull-request submissions. Jenkins informs the team about build status through Slack and email.
7. GitHub informs the team about pull-requests via slack and email.
8. Deployed the system on Amazon AWS.

#### Functionality Implemented:

1. Got legacy code base running.
2. Added a notion of user and groups of users to the system.
3. Direct message to individual functionality implemented.
4. Direct messaging to multiple users implemented.





## Sprint 2

### Functionality Implemented:

1. Implemented user and group persistent functionality using MySQL.
2. Added create, read, update and delete capabilities for user and group.
3. Implemented login functionality using user and password and stored password in encrypted form.
4. Implemented direct messaging to individual and groups.
5. Implemented message persistence functionality.
6. Added retrieve messages for user and group functionality.

## Sprint 3

### Environment setup:

1. Send failure notices to Slack.
2. Send security concerns to Slack.

### Functionality Implemented:

1. Added last seen attribute for user and queued messages for user which are currently offline.
2. When user comes online, all messages from queue are delivered in order they were received.
3. Added agency role and implemented wiretap functionality. Agency needs admin's permission to wiretap any user or group.
4. Implemented recall message functionality.
5. Wrapped messages with sender and receiver's IP address and stored them in MySQL.
6. Added parent control functionality for system, used Google's list of bad words and used mutual information method.
7. Added message search based on sender, receiver, content and date feature. This functionality is only available for admin.
8. System logs activity using log4j.
9. Implemented dynamically turning on or off logging functionality.



## Retrospective

The project was a simulated industrial experience where we explored various integration and inspection tools like JIRA, SonarQube and Jenkins. Apart from the technology being used we experienced how a team work is conducted and carried out.

The course helped us understand the workflow for a project. Since we used an agile process we were successful in implementing features and functionalities in sprints. With each sprint carrying the backlogs and new features the team was able to communicate better and work smarter. Integration with Slack kept the team updated with pull requests, GitHub details, failed merges and updates from the server. The report on SonarQube enabled the team to follow proper coding standards and patterns.

The first sprint taught us about the server socket connection and the sprints that followed allowed us to build a complete communication system. The course work helped with the implementation of design patterns, industrial standards and agile methodologies. It was a good experience working with a team and achieve the goals set for the project. We learnt that hustling to only build mediocre software applications falls short in its scope of capturing the essence of quality software.

Though the course taught us about managing a software development, we think instead of having same project for every team we should be given a chance to pick project from specific domain and given liberty to work with any technologies we want to explore. That will allow us to learn industry standards and explore new domains and new technologies.