

# Prattle Chat Server



Rakesh, Sourabh, Shweta, Aswath

# **System Functionality**

# Given

- A barebone messaging system
- Basic template
  - Only one type of message- broadcast.

# Goal

- Create a robust messaging system with functionalities like
  - Persist messages, users, authentication
  - Add a notion of usergroups - and related functions
  - Add additional features like parental control, wiretapping.

# User and Group of Users

- Adding user and groups of users to the system and persist them.
  - LDAP (Lightweight Directory Access Protocol) for persistence.
- Login functionality
  - User registration with username and password.
  - User can login to the system using username and password.
  - Authentication of user is done using LDAP.
- Direct Messages and Group Messages
  - User can send direct messages to online and offline users.
  - Users can send group messages online and offline users.
  - Create new users, update and delete users from the system.
  - Create user groups, update and delete them as well.



# Message Functionality

- Persistence through RDBMS (MYSQL)
- Offline message queuing.
  - User receives offline messages in the order it was sent.
- Recall message
  - Messages sent to offline receivers can be recalled by the sender.
- Message search
  - User can search messages based on sender, receiver or timestamp.

# Authority in our System

## Communications Assistance for Law Enforcement Act Regulatory Compliance Guide

Your Comprehensive Guide for Ensuring Compliance with Federal Communications Commission (FCC) Regulations Governing Law Enforcement Access under CALEA



**Communications Law Compliance Series**  
from The CommLaw Group



- Legal
  - CALEA Compliance
  - Wiretap for Subpoena
  - IP Wrapping

- Message Filter
  - Monitoring of content
  - Workplace
  - Parental Controls

**Job Quality**

# **Process Management**



# JIRA - Project Management

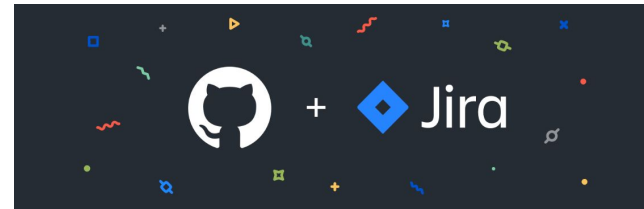
- Tickets for new features to be added.
- Issue tracking.
- Smart commits.



MSD212-68 #comment search by attribute implementation



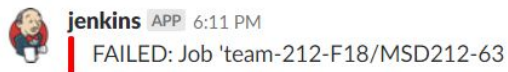
rakeshr99 committed 18 days ago



# Slack - Effective Communication

- Slackups an alternative for daily scrum

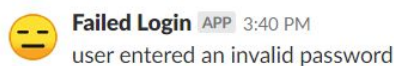
- Jenkins and Slack integration



- Github and Slack integration



- Send security concerns to slack



- Regular updates on tasks



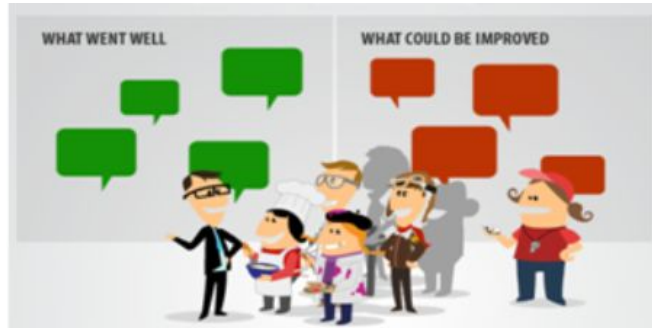
# Github - Version Control

- Used git for version control
- Git Branches with JIRA ticket number.
- Using Pull Requests(PR) for peer review.



# Sprint Review and Retrospective

- Sprint review with TA
- Tips on Source Control
  - Creating Pseudo master
  - No dangling branches
- Sprint Retrospective to identify issues in sprint and improve.



# **Automated Processes**

# Automated Processes

- Jenkins automation server for continuous integration and continuous delivery.
  - Helped us keep good test coverage on existing code and new code being added.
- SonarQube
  - Continuous inspection of code quality.
- Adjusted testing strategy to minimize issues/downtime
- Conformed to automated deployment to AWS services at end of each release cycle.



# **Quality Assurance**

# Quality Assurance

- Project meets total coverage of 89.1%.
- No Bugs/Vulnerabilities.





# Quality Assurance

- Well documented java docs
- Programmed to an interface

```
import java.util.List;
```

```
You, 9 days ago | 2 authors (You and others)
```

```
/**
```

```
 * The interface Message DAO layer.
```

```
 *
```

```
 * @author Sourabh Punja on 11/13/2018
```

```
 */
```

```
public interface MessageDao extends BaseDao{
```

```
    int createMessage(Message message);
```

```
    /**
```

```
     * Persist message in RDS.
```

```
     *
```

```
     * @param message the user message to be persisted.
```

```
     * @return the int denoting if the message has been added.
```

```
     */
```

```
    int createUserAndGroupMessage(Message message);
```

```
    /**
```

```
     * Find all messages by username and isSeen False.
```

```
     *
```

```
     * @param userName the user name
```

```
     * @return the list of messages for the user
```

```
     */
```

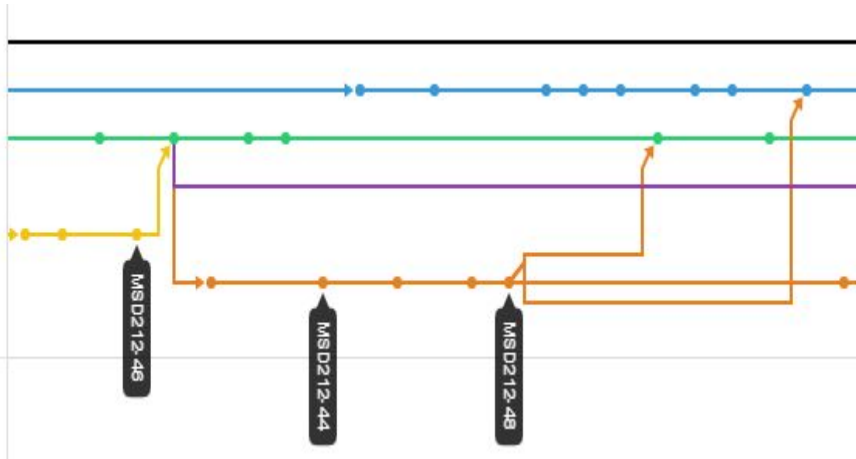
```
    List<Message> findMessageByUsernameAndUnSeen(String userName);
```

# **Performance Improvements**

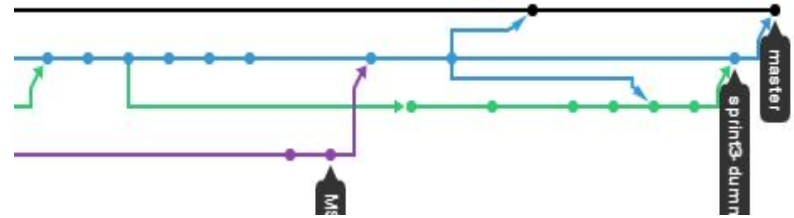
# Performance Improvements

- Improved our branching strategy.

Initially



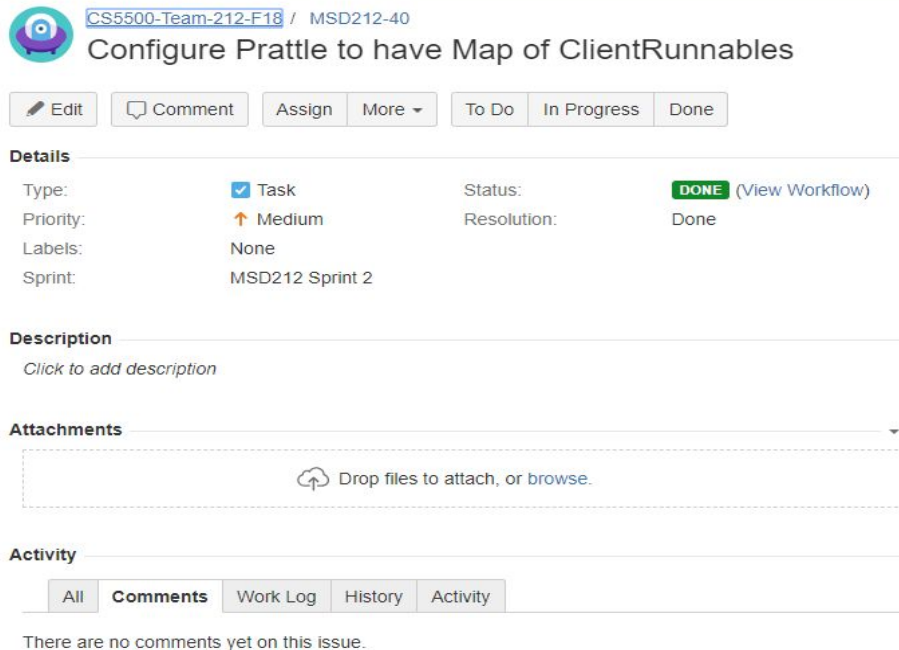
Finally



# Performance Improvements

- Developed features connected to jira tickets and used smart commits.

Initially:: No Smart Commits



The screenshot shows a Jira issue page. At the top, there's a header with a purple robot icon, the text 'CS5500-Team-212-F18 / MSD212-40', and the issue title 'Configure Prattle to have Map of ClientRunnables'. Below the title are buttons for 'Edit', 'Comment', 'Assign', 'More', and a status bar with 'To Do', 'In Progress', and 'Done'. The 'Details' section shows fields for Type (Task), Priority (Medium), Labels (None), Sprint (MSD212 Sprint 2), and Status (DONE). The 'Description' section has a placeholder 'Click to add description'. The 'Attachments' section has a dashed box with a cloud icon and the text 'Drop files to attach, or browse.'. The 'Activity' section has tabs for 'All', 'Comments', 'Work Log', 'History', and 'Activity'. At the bottom, it says 'There are no comments yet on this issue.'

**CS5500-Team-212-F18 / MSD212-40**

## Configure Prattle to have Map of ClientRunnables

Edit Comment Assign More To Do In Progress Done

**Details**

Type: ☒ Task Status: **DONE** (View Workflow)  
Priority: ☒ Medium Resolution: Done  
Labels: None  
Sprint: MSD212 Sprint 2

**Description**

Click to add description

**Attachments**

Drop files to attach, or browse.

**Activity**

All Comments Work Log History Activity

There are no comments yet on this issue.

# Performance Improvements

## Smart commits

**Activity**

All

Comments


Work Log

History


Activity

↑


6 older comments

✓  Rakesh Krishna Radhakrishnan added a comment - 30/Nov/18 5:17 PM


validating the input string provided for recall message

✓  Rakesh Krishna Radhakrishnan added a comment - 30/Nov/18 6:29 PM

added test cases for recall and search feature

✓  Rakesh Krishna Radhakrishnan added a comment - 30/Nov/18 7:44 PM

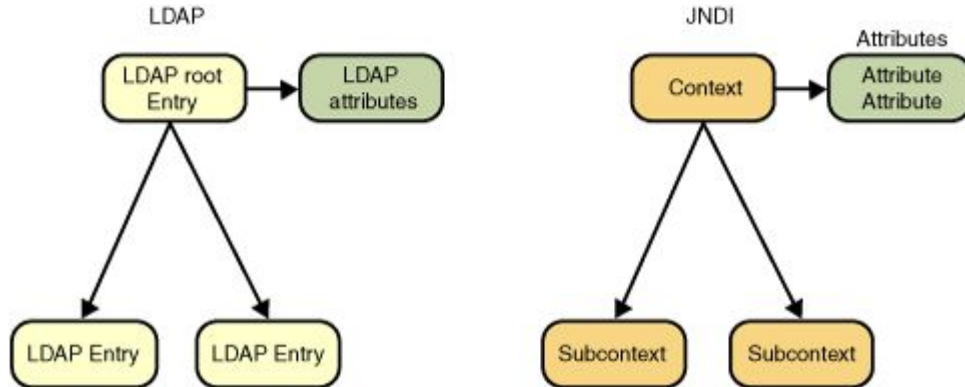
added comments for recall

✓  Rakesh Krishna Radhakrishnan added a comment - 30/Nov/18 7:55 PM

triggering build

# Performance Improvements

- Switched from JNDI to LDAP API support.



# Performance Improvements

## Better Description



CS5500-Team-212-F18 / MSD212-70

Creating a message to recall messages sent.



Edit



Comment

Assign

More ▾

To Do

In Progress

Done



Export ▾

### Details

Type: ☒ Task Status: **DONE** (View Workflow)  
Priority: Medium Resolution: Done  
Labels: None  
Sprint: MSD212 Sprint 3

### Description

Creating a message to recall messages sent. Recalling a message may only be done by the message's sender. Recall messages should be delivered to clients. Recalled messages are not delivered as part of the message backlog delivery (from item 3. Recalled messages are not deleted from the server

### > Attachments

### Activity

### > People

### > Dates

### Development

2 branches

13 commits

Latest 01/Dec/18 1:05 AM

1 pull request

**MERGED**

Updated 01/Dec/18 11:57 AM

### Agile

Active Sprint:

MSD212 Sprint 3 ends 04/Dec/18

[View on Board](#)

# **Teamwork and Process**



# In person meetings before a sprint

- Discuss the division of work
- Assign work based on comfort/ strengths of team mates
- Create JIRA tickets

Backlog 10 issues

✓ ↑ MSD212-29 Configuring security groups for AWS

✓ ↑ MSD212-23 Mock objects for socket connection

✓ ↑ MSD212-7 Stretch (large) - Directing messages to groups, including a reply-all function.

✓ ↑ MSD212-22 SocketNB connection

✓ ↑ MSD212-24 Find how ClientRunnable needs to be tested

✓ ↑ MSD212-25 Coverage issues - ClientRunnable

✓ ↑ MSD212-27 Heroku deployment

✓ ↑ MSD212-28 Jar not recognized

✓ ↑ MSD212-33 Design general functionality for Groups

✓ ↑ MSD212-34 Extend Groups for direct messaging

+ Create issue

# SLACK

- Main channel of communication.
- Slackups - daily update
- Design decisions were always communicated.



# Helped each other with issues

- Stuck with something? Ask for help



# **Camaraderie through Fire**

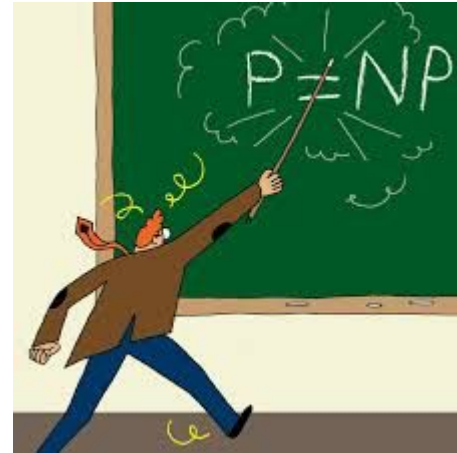


# **Challenges / Shortcomings**

# Scheduling

- One member works full time
- One member proctors until 8AM

**WHEN? WHEN? WHEN?**



# Managing Work and Time

SPRINT1 : PANIC until it works!



Rest of Project:



# Communication

*“The single biggest problem in communication is the illusion that it has taken place.”*

-- George Bernard Shaw

- Communication means a LOT more than saying words, and it MUST not be one sided



# Finally....



Code walk as a team.  
Gain Understanding!!

IT WAS HARD!!



Pair Programming

# **Tech. Transfer & Beyond**

# Client Readiness

- The system is in a working state with clear expectations that have been met
- Core functions expected for a text messaging system is delivered
- Tech Stack :
  - Java
  - LDAP
  - MySQL
- Well documented at appropriate levels with a system setup video
- Clear paths for extension available, i.e:
  - Administrator Privileges
  - Message Types

**Thank you!!**