

# Final Presentation

Farha Jawed  
Pratik Devikar  
Yang Xia

# Team Member Introduction

- Team 111
  - Farha Jawed
  - Pratik Devikar
  - Yang Xia

# System Functionalities

**Did the team achieve a sufficient amount of functionality?**

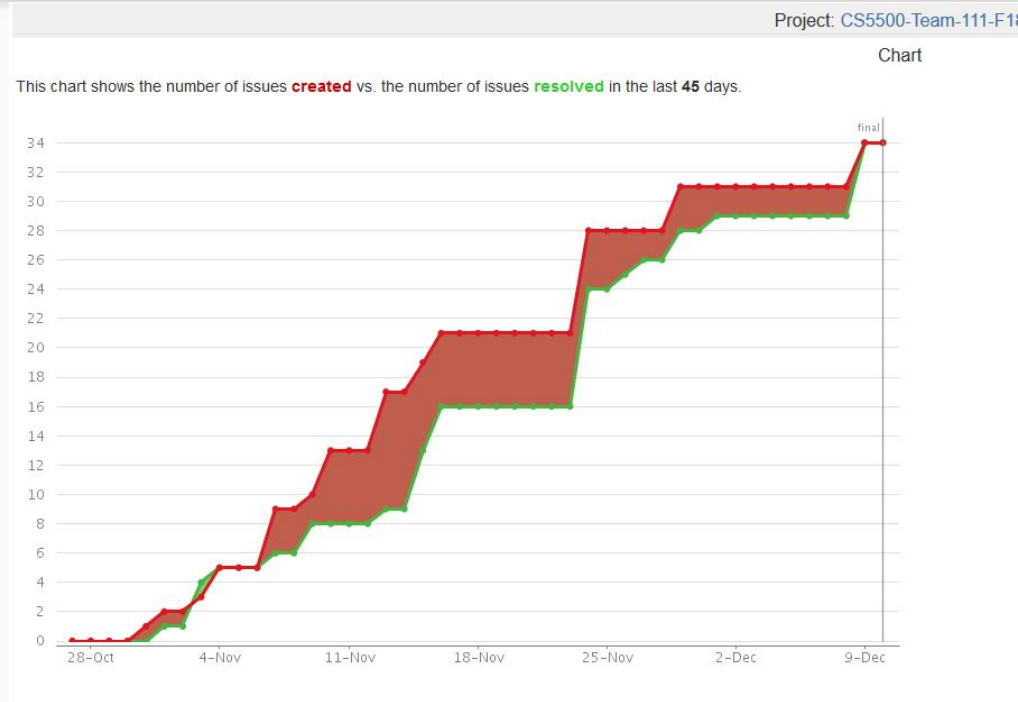
Yes. The system is working properly.

All the basic functionalities that a simple messaging application should have are being implemented.

Ex: Sending messages to other users and groups, storing messages for later retrieval and sending messages to offline users.

# System Functionalities

Fig: Issues created vs resolved



# System Functionalities

## **Did the team achieve what it set out to do?**

Yes. We did complete all the sprint expectations.

Getting the legacy code running which contains Prattle and ClientRunnable, Crud functionality for users and groups, sending messages to and from users and groups, delivering messages to offline users and providing a service for an agency to view the messages. Our system also provides the users with parental control.

# System Functionalities

Although being a member of 3 people, we weren't able to complete some of the stretches such as adding MIME(Multi-purpose Internet Mail Extension) type to messages and using Apache JMeter to test the system for stress testing and Performance testing.

# System Functionalities

## **Is it useful for a client?**

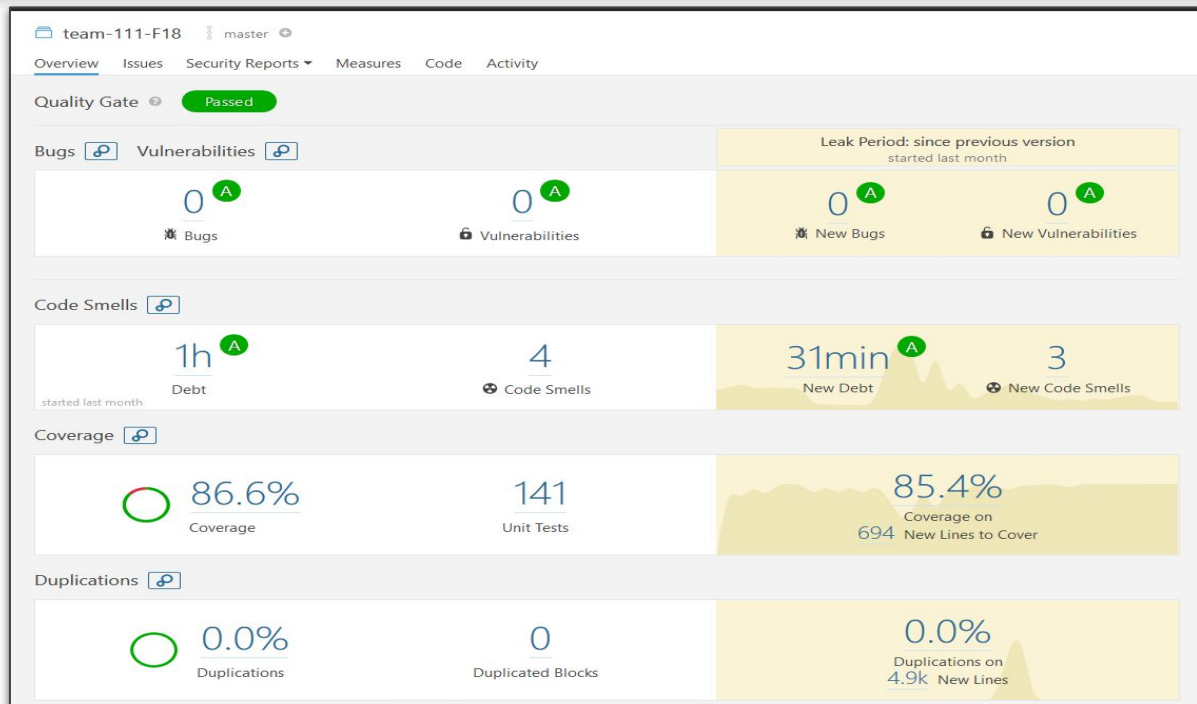
Yes, most of the basic back-end system functionalities have been implemented. But a user also desires of a good GUI for any application. We didn't do the chatter/user interface. So in the future if someone develops the chatter side of the project, it will be ready for deployment.

# Job Quality

- Did the team present evidence that it did a quality job?
- Did the team improve its performance over time? (And is this claim supported with evidence?)



# Job Quality Contd.



## SonarQube Report

- 86.6% code Coverage
- 141 unit tests
- No bugs and vulnerabilities
- 0% code duplication

# Job Quality Contd.

## Performance over time

- Sprint 1: barely met the minimum requirement
  - Wrote test cases for Prattle

# Job Quality Contd.

## Performance over time

- Sprint 2 and 3: Could do more stretches
  - Worked efficiently by prioritizing tasks on Jira
  - Distributed works evenly among team members

# Job Quality Contd.

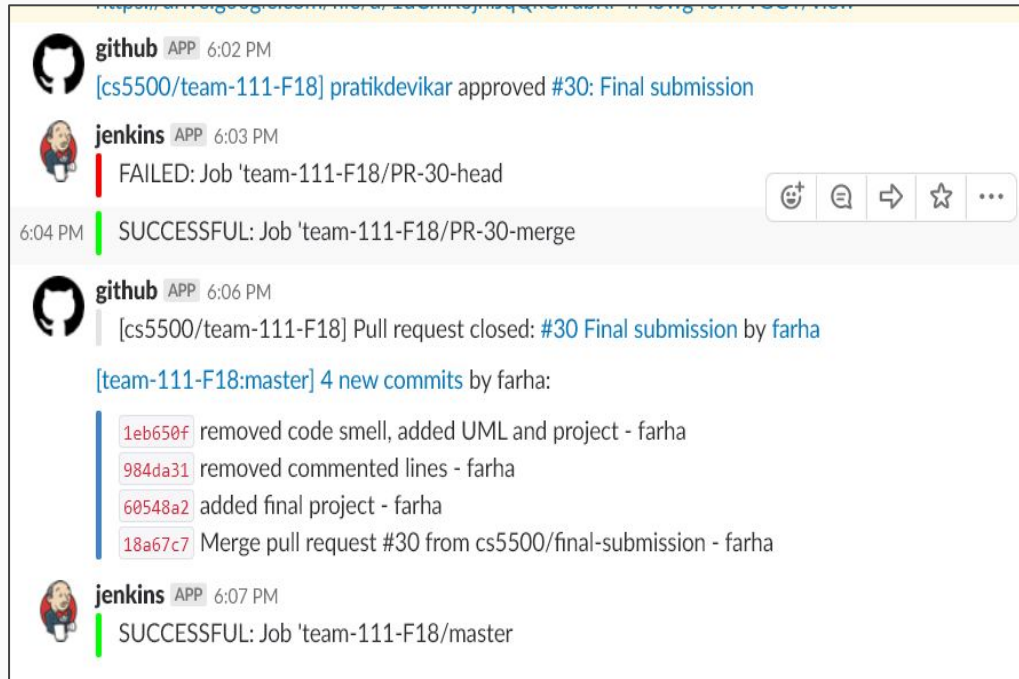
The screenshot displays the Jira interface for the project 'CS5500-Team-111-F18'. The top navigation bar includes 'Dashboards', 'Projects', 'Issues', 'Boards', and a 'Create' button. A search bar and user profile icon are on the right. The left sidebar contains navigation links: 'Backlog', 'Active sprints', 'Releases', 'Reports', 'Issues' (selected), and 'Components'. Below these are 'PROJECT SHORTCUTS' with a link to 'Add a link to useful information for your whole team to see.' and a '+ Add link' button. The main content area is titled 'All issues' with a 'Switch filter' dropdown. It lists several issues, including 'MSD111-40' (Stretch: Github should inform the teams...), 'MSD111-39' (Stretch: Jenkins should inform teams of ...), 'MSD111-38' (Group partition in LDAP), 'MSD111-37' (Refactor User to follow design pattern), 'MSD111-34' (Directing messages to groups), 'MSD111-33' (Directing messages to individuals), 'MSD111-32' (Group CRUD), 'MSD111-31' (User CRUD), 'MSD111-30' (Remove code smells), and 'MSD111-29' (Remove code vulnerabilities). A '+ Create issue' link is at the bottom. The 'Attachments' section shows a 'Drop files to attach, or browse.' prompt. The 'Activity' section has tabs for 'All', 'Comments', 'Work Log', 'History', and 'Activity'. It lists several comments by 'Farha Binte Jawed' dated 09/Nov/18, including 'User model and service', 'UserService wrong attr fix', 'Get original password text', 'Change in user repo methods', 'method for login check', and 'updates entries'. The right sidebar shows 'Development' with '1 branch', '8 commits' (Latest 09/Nov/18 7:32 PM), and '1 pull request' (MERGED, Updated 15/Nov/18 2:55 PM). It also shows 'Agile' with 'Future Sprint: MSD111 Sprint 2' and a 'View on Board' link.

## Performance over time

### Sprint 2:

Started using smart commits on Git

# Job Quality Contd.



The screenshot shows a Slack conversation with the following messages:

- github APP** 6:02 PM  
[cs5500/team-111-F18] pratikdevikar approved #30: Final submission
- jenkins APP** 6:03 PM  
FAILED: Job 'team-111-F18/PR-30-head'
- 6:04 PM  
SUCCESSFUL: Job 'team-111-F18/PR-30-merge'
- github APP** 6:06 PM  
[cs5500/team-111-F18] Pull request closed: #30 Final submission by farha  
[team-111-F18:master] 4 new commits by farha:
  - 1eb650f removed code smell, added UML and project - farha
  - 984da31 removed commented lines - farha
  - 60548a2 added final project - farha
  - 18a67c7 Merge pull request #30 from cs5500/final-submission - farha
- jenkins APP** 6:07 PM  
SUCCESSFUL: Job 'team-111-F18/master'

## Performance over time

### Sprint 2:

Integrated Jenkins and Github with Slack

# Job Quality Contd.

## **Performance over time**

- Team met more frequently
- Incorporated feedback from sprint reviews

# Process and Team Work

Did the team work as a team?

Yes!!!!!!!

The fact that we as a 3 member team is able to finish the project is the best proof.

# Process and Team Work

Did the team use process well or was the project mis/mal-organized?

Show Jira.

Assigned tickets, update stories, and smart commits.

Slack webhook

Email notifications



# Process and Team Work

Was the team able to automate the build, test, and promote process?

Jenkins/SonarQube.

# Technology Transfer

Is is the system in shape to be handed over to the client?

1. Client side needs more work
2. Still had some issues with LDAP server
3. Need to modify the way we persist messages

Future works:

1. Fixed the issues that mentioned above
2. Writing test cases using a real server in addition to Mockito
3. Provide user interface