

Managing Software Design



Team 204 members

- Kasi Karuppiah Alagappan
- Neha Gundecha
- Ruturaj Nene
- Vishal Patel

System Functionalities



System Functionality and usability

Key notes:

1. Functionalities of the system - Users, Messages, Groups
2. Team achievement through the sprints
3. Client usability and usage

Functionalities of the system - Users

- User login and registration with a unique username
- Password encryption
- View and edit user profile
- Follow and unfollow a user to view activities of the users in the circle
- Help documented for user to view predefined intuitive commands for the command line interface

Functionalities of the system - Messages

- Private messages to others users
- Broadcast messages to all the users
- Group messages if part of the group
- Open chat when user is online
- Chat notifications when user is offline

Functionalities of the system - Groups

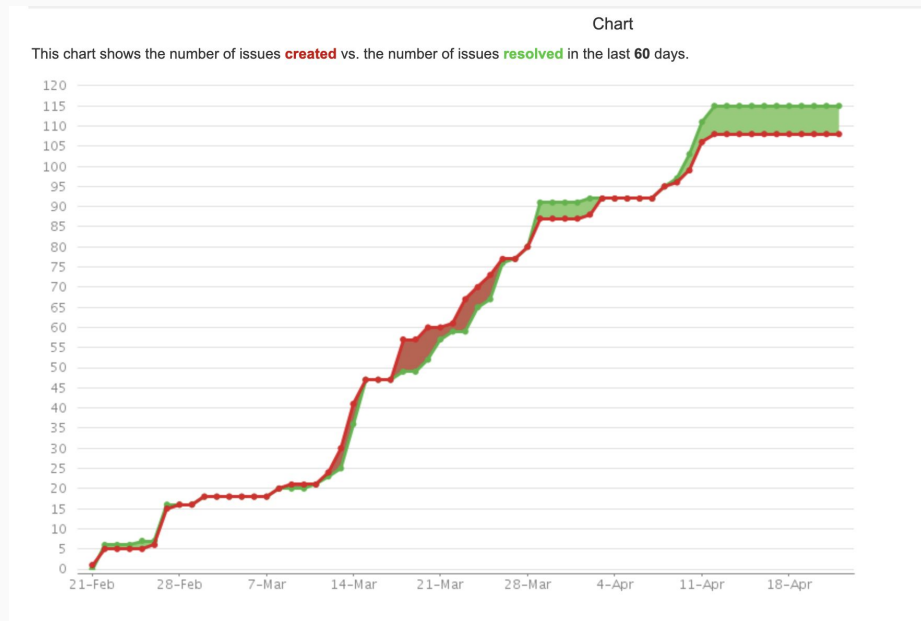
- Create and delete groups
- Add and delete user from a group
- Change moderator approval to add or delete user from a group
- Send group invites
- View and edit group profile

System functionalities

Did the team achieve a sufficient amount of functionality?

- Yes, all the system functionalities work properly and the product is stable.
- Implemented the must haves the client wanted in the product as stated in the backlog
- Both the frontend (CLI) and backend is deployed and the product is ready to use by the client

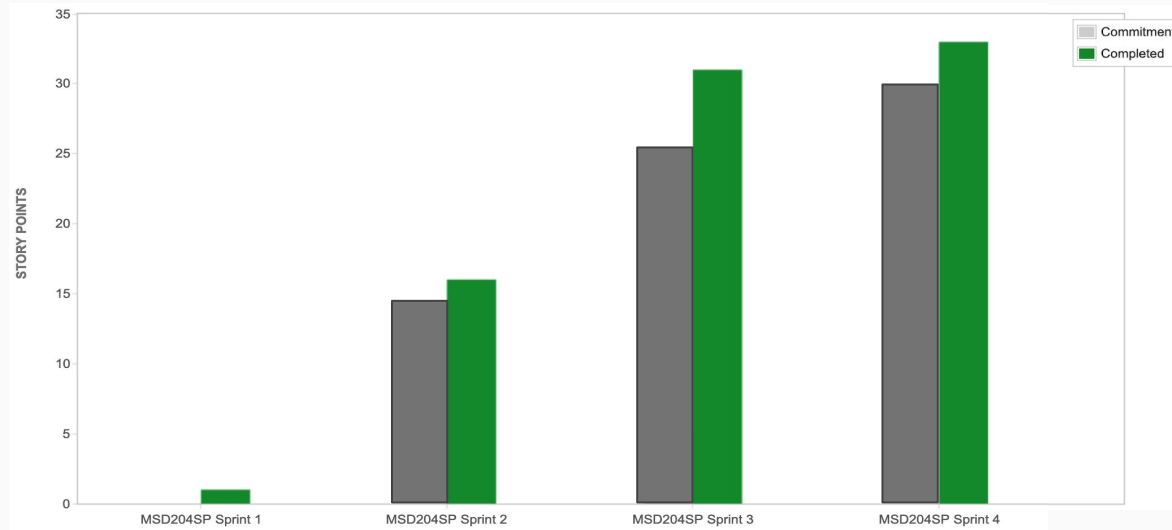
Issues created vs issues solved



Did the team achieve what it set out to do?

- Yes, we completed all the goals for each sprint
- Completed a good portion of the backlog from the client
- We also completed some bonus tasks in mostly all of the sprints
- Delivered product with good code quality and and stability with all the edge cases handled

Amount of work completed from sprint to sprint

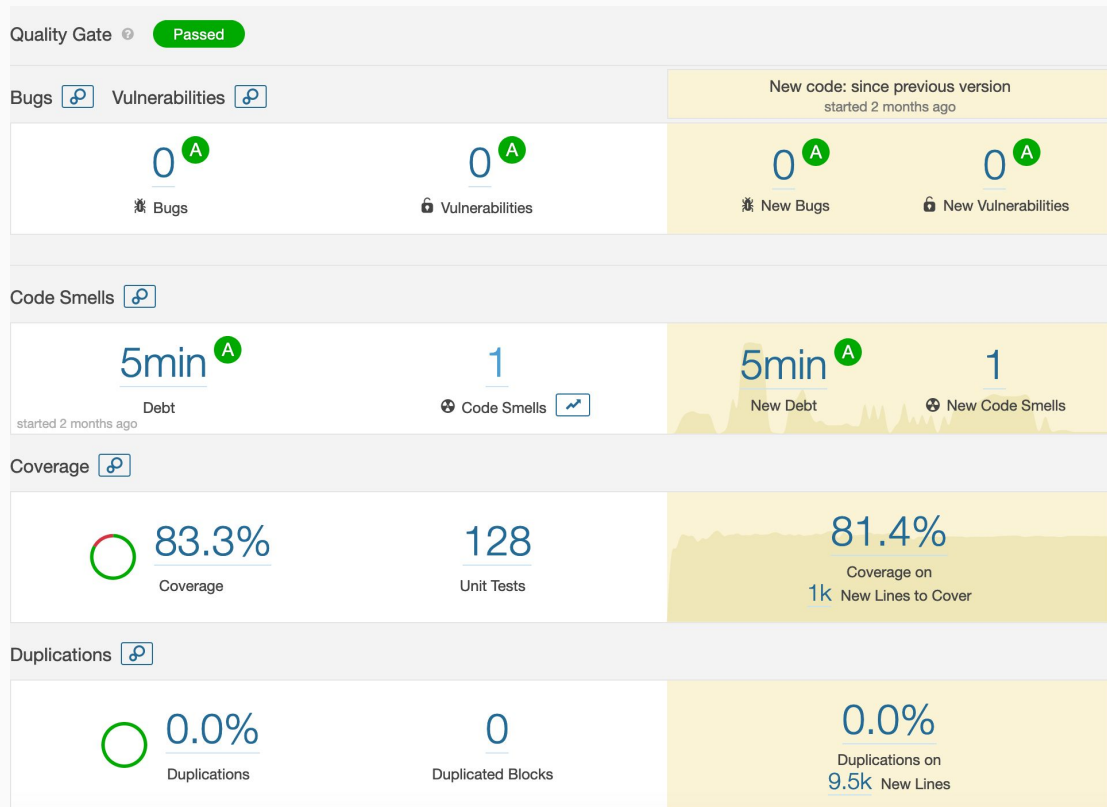


Is it useful to the client ?

- Yes, the product is deployed and ready to use for the client
- Though being command line interface, we have included intuitive commands for various tasks the user can perform on the system
- The product includes a help that guides through all the commands the user can use on the product for various tasks

Job Quality

SonarQube Report



- There are 128 unit tests that achieve 83.3% coverage.
- There is 0% Code Duplications
- There is only one minor code smell.
- There are no Bugs
- There are no Vulnerabilities.

Security Report
























OWASP

OWASP Top 10

Track Vulnerabilities and Security Hotspots conforming to OWASP Top 10 standard. [Learn More](#)

 Additional security-related rules are available but not active in your profiles.

☐ Show CWE distribution 

Categories	 Vulnerabilities	 Security Hotspots		
		Open	In Review	Won't Fix
A1 - Injection 	0 	2	0	0
A2 - Broken Authentication 	0 	0	0	0
A3 - Sensitive Data Exposure 	0 	4	0	0
A4 - XML External Entities (XXE) 	0 	0	0	0
A5 - Broken Access Control 	0 	0	0	0
A6 - Security Misconfiguration 	0 	0	0	0
A7 - Cross-Site Scripting (XSS) 	0 	0	0	0
A8 - Insecure Deserialization 	0 	0	0	0
A9 - Using Components with Known Vulnerabilities 	0 	0	0	0
A10 - Insufficient Logging & Monitoring 	0 	3	0	0
Not OWASP	0 	0	0	0









SANS

SANS Top 25

Track Vulnerabilities and Security Hotspots conforming to SANS Top 25 standard (25 CWE items in three categories). [Learn More](#)

 Additional security-related rules are available but not active in your profiles.

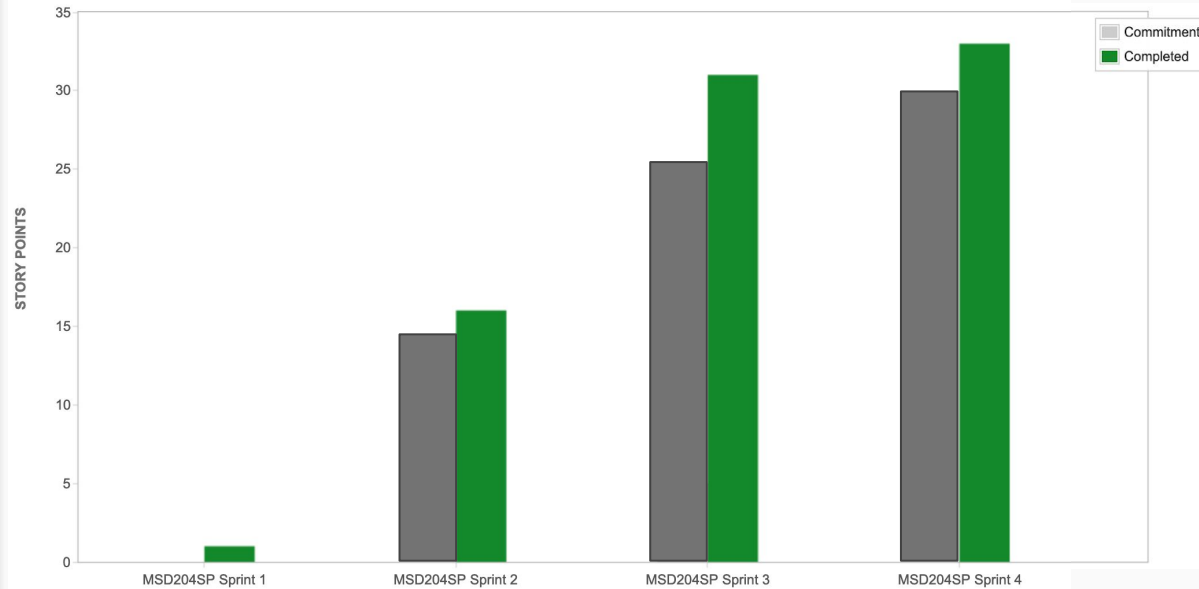
☐ Show CWE distribution 

Categories	 Vulnerabilities	 Security Hotspots		
		Open	In Review	Won't Fix
Porous Defenses 	0 	0	0	0
Risky Resource Management 	0 	0	0	0
Insecure Interaction Between Components 	0 	0	0	0

- In all the measure for the OWASP and SANS security report the projects has A rating.

Team Progress

- Sprint 1 was the planning phase.
- Through Sprint 2 through 3 the amount of work measure in terms of story points increased.
- Sprint 4 was more focused towards delivering a robust product than adding more features.



Process and Teamwork



Did the team work as a team?

- The team functioned quite efficiently and in an orderly manner
- The communication between the team was quite remarkable
- Everyone on the team had a fair share of contribution towards the project
- The team was collectively focused on the goals
- The team helped each other to overcome hurdles
- Everyone had great understanding and chemistry
- The team was quite flexible in terms of everyone's schedules
- We had team lunches to facilitate team bonding

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

Yes, the team used the process well and was highly organized which is highlighted by the points below.

- The team conducted regular standups
- JIRA board was greatly used to organize issues and workflow of tasks
- JIRA-GitHub Integration: Facilitated Smart commits

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

The team automated the build, test and promote processes using the following:

- Jenkins: Facilitated the automated building and testing of the code
- GitHub: Facilitated version control and source code management
- JIRA: Facilitated the task of tracking issues/tasks
- SonarQube: Maintained Code quality with reports covering test coverages, code smells, bugs, etc.
- Slack-GitHub-Jenkins Integration: Notified about commits to GitHub and jobs on Jenkins

Technology Transfer

Handoff to Client

- The code is stable and free of any bugs/ vulnerabilities
- The server is live and readily accepting connections from clients
- The server load can be monitored from the Stat website

Next Steps

- The code is well documented
- The server side follows a factory design pattern and the client side follows a command design pattern so any new feature can be added with minimal changes to existing code
- The app uses singleton implementation for DB handle which can be easily modified to create additional DB handle in case of high server loads
- The Stat page can be extended to log and view additional information as required

Thank you