CS5500: NUSlack

Team 11: Mitali, Thomas, Tim, and Yanchi

SYSTEM FUNCTIONALITY

Overall

- Team consistently met/exceeded biweekly backlog
- Achieved nearly every priority 1 requirement on original SRS
- Departures from original SRS
  - Pivoted away from groups of groups, Subpoena, and some lower priority features like translation
    - Implemented some not originally present features like messages with an alias
  - Communicated with / got approval from product owner
  - Agile scrum method provided us the adaptability / transparency required to be successful in a dynamic environment

Backlog Overview - Sprint 2

- Write tests on existing code to be able to merge new pull requests ✔️
- Introduce a database connection ✔️
- Persist users (store users in the database) ✔️
- Support User-to-User messaging ✔️

Backlog Overview - Sprint 3

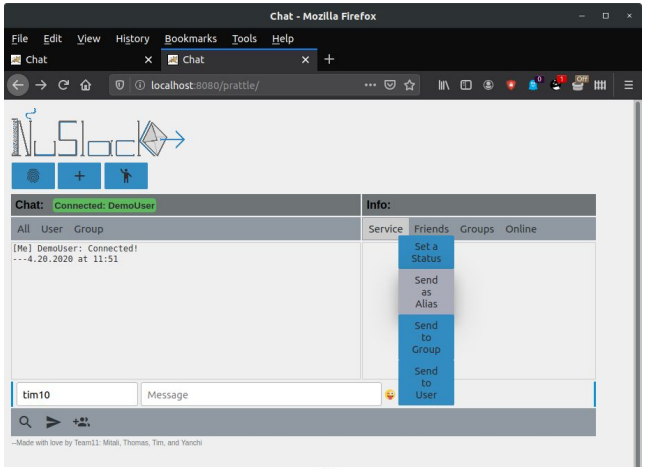
- ❏ Messages will have a timestamp. ✔️
- ❏ Messages will persist in the database. ✔️
- ❏ When a User logs in, they will see a list of all messages that were sent to them. ✔️
- ❏ A User will have the ability to search for another user and add that user as a friend. ✔️
- ❏ Users will be able to form groups. If a User forms a group, that User is automatically the moderator of that Group. ✔️
- ❏ Users will be able to join Groups that already exist. ✔️
- ❏ If a User is a member of a Group and sends a message to that Group, then all Users who are members of that Group, and no Users who are not members of that Group, will receive the message. ✔️

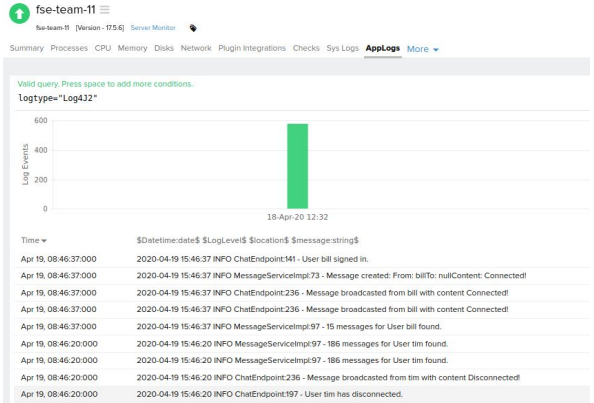
Backlog Overview - Sprint 4

- Users will be able to search for another user and send a friend request. ✔️
- Users will be able to accept friend requests. ✔️
- Users will be able to locate the users who have accepted a friend request. ✔️
- A user will be able to set a status and persist the status in the database. ✔️
- Users will see the status of another user. ✔️
- Users will be able to see who is currently online. ✔️
- Logging formats will be standardized. Logs will be viewable from the dashboard. ✔️
- Database will be moved. ✔️
- Moderators are able to add other users as Moderators to an existing group. ✔️
- Users will be able to send an anonymous message to a group. ✔️

Utility for client

- Resulted in full-featured messaging service
  - Includes User Interface
  - System implements detailed logging for runtime information/parameterization
  - Can be expanded or deployed at present state
    - Delivering ready-to-go product
    - Code is abstract, encapsulated, modular and extensible
    - Java documentation included





JOB QUALITY

Why focus on quality?

MODERN RESOLUTION FOR ALL PROJECTS

	2011	2012	2013	2014	2015
SUCCESSFUL	29%	27%	31%	28%	29%
CHALLENGED	49%	56%	50%	55%	52%
FAILED	22%	17%	19%	17%	19%

The Modern Resolution (OnTime, OnBudget, with a satisfactory result) of all software projects from FY2011–2015 within the new CHAOS database. Please note that for the rest of this report CHAOS Resolution will refer to the Modern Resolution definition not the Traditional Resolution definition.

Source:
https://www.standishgroup.com/sample_research_files/CHAOSReport2015-Final.pdf

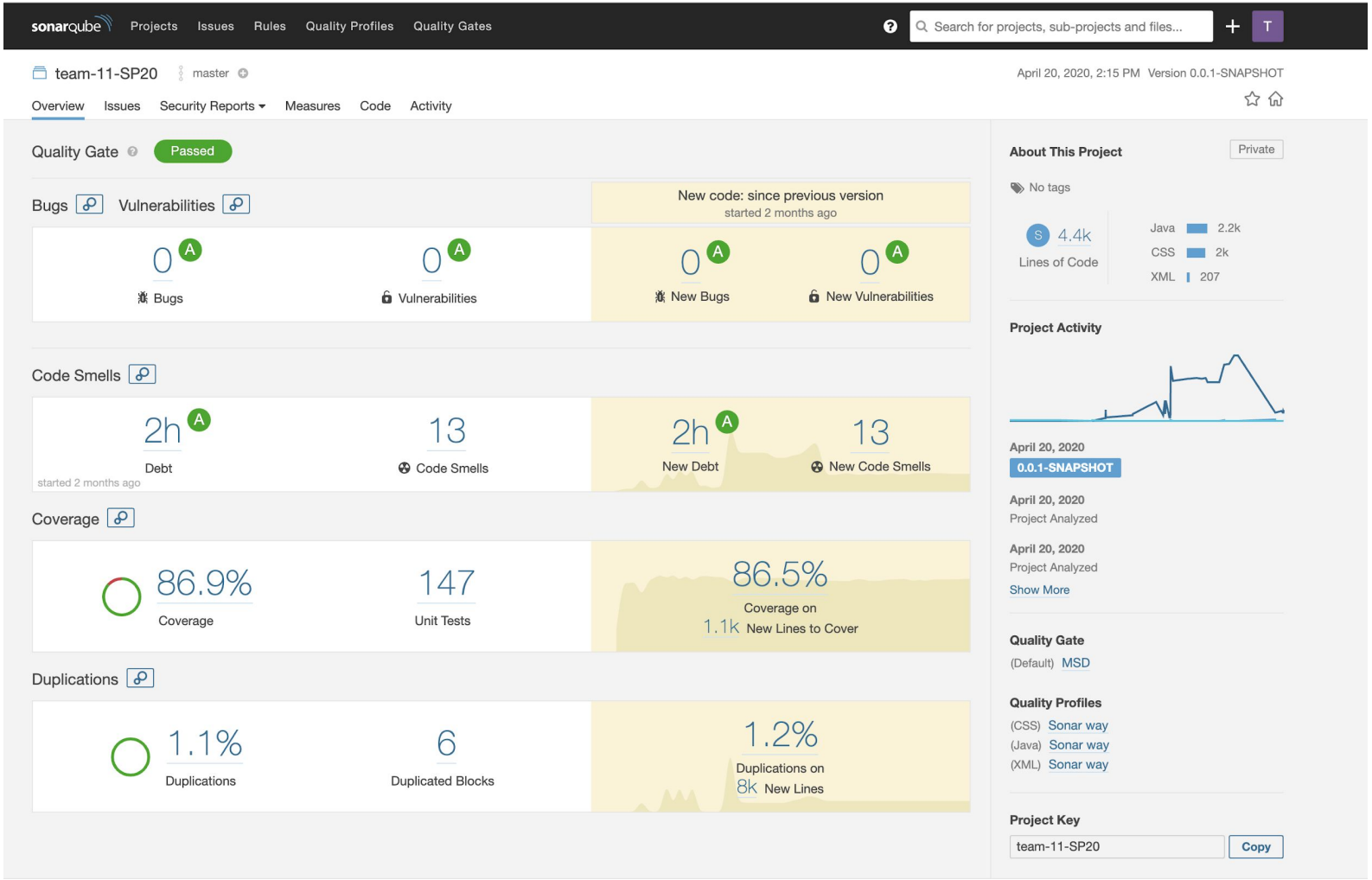
Why focus on quality?

Metric Type	Service Desk Cost Metrics	North American Statistics		
		Average	Min	Max
Cost	Cost per Ticket	$15.56	$2.93	$49.69
	Cost per Minute of Handle Time	$1.60	$0.76	$2.50

Source:
http://www.thinkhdi.com/library/supportworld/2017/metric-of-month-service-desk-cost-per-ticket.aspx,
Jeffrey Ruburg 2017

Standards of quality

1. Test coverage
    a. >= 50% condition coverage
    b. >= 80% line coverage
2. Code quality
    a. <= 3% duplicated lines of code
    b. Highest maintainability rating
    c. Highest reliability rating
    d. Highest security rating


sonarqube
Projects
Issues
Rules
Quality Profiles
Quality Gates
Search for projects, sub-projects and files...
T
team-11-SP20
master
April 20, 2020, 2:15 PM Version 0.0.1-SNAPSHOT
Overview
Issues
Security Reports
Measures
Code
Activity
Quality Gate
Passed
Bugs
Vulnerabilities
New code: since previous version
started 2 months ago
0 A
Bugs
0 A
Vulnerabilities
0 A
New Bugs
0 A
New Vulnerabilities
Code Smells
2h A
Debt
started 2 months ago
13
Code Smells
2h A
New Debt
13
New Code Smells
Coverage
86.9%
Coverage
147
Unit Tests
86.5%
Coverage on
1.1k New Lines to Cover
Duplications
1.1%
Duplications
6
Duplicated Blocks
1.2%
Duplications on
8k New Lines
About This Project
Private
No tags
4.4k
Lines of Code
Java 2.2k
CSS 2k
XML 207
Project Activity
April 20, 2020
0.0.1-SNAPSHOT
April 20, 2020
Project Analyzed
April 20, 2020
Project Analyzed
Show More
Quality Gate
(Default) MSD
Quality Profiles
(CSS) Sonar way
(Java) Sonar way
(XML) Sonar way
Project Key
team-11-SP20
Copy

src/main/webapp/javadoc/jquery/jquery-ui.css

Unexpected duplicate selector ".ui-icon", first used at line 70 ···

2 days ago L294

Code Smell Major Open TA Thomas A. Dohle 1min effort Comment

No tags

Unexpected duplicate selector ".ui-widget-overlay", first used at line 82 ···

2 days ago L532

Code Smell Major Open TA Thomas A. Dohle 1min effort Comment

No tags

src/main/webapp/javadoc/stylesheet.css

Unexpected duplicate selector "ul.blockList ul.blockList ul.blockList li.blockList h3", first used at line 298 ···

2 days ago L625

Code Smell Major Open TA Thomas A. Dohle 1min effort Comment

No tags

13 of 13 shown

Testing approach

1. Extend test coverage for the initial code the team was given
2. Test-driven development for new code
    a. Extensive coverage for all new classes/methods written
    b. Regular evaluation of code in development with jacoco and sonar lint
    c. High standards for code review
3. Create an environment that permits running tests quickly and often
    a. Configured a throwaway, in-memory database for use in testing
        i. Speeds up tests
        ii. Protects production database from errors being introduced during testing

Finding problems

- Implemented production logging using Log4J2 to identify problems when they happen
  - Rollover log files ensure that the production environment’s memory is not exceeded
- Added extensive logging coverage to relevant classes/methods
- Configured site24x7 to read production logs in real time and monitor site uptime

```
root@fse-team-11:/var/log# ls
alternatives.log        cloud-init.log  lastlog         syslog.4.gz
apt                     dist-upgrade    lxd             syslog.5.gz
auth.log                dpkg.log        nuslack-logs    syslog.6.gz
auth.log.1              journal         syslog          syslog.7.gz
auth.log.2.gz           kern.log        syslog.1        tallylog
btmp                    kern.log.1      syslog.2.gz     unattended-upgrades
cloud-init-output.log   landscape       syslog.3.gz     wtmp
root@fse-team-11:/var/log# cd nuslack-logs/
root@fse-team-11:/var/log/nuslack-logs# ls
current_log.log  previous
root@fse-team-11:/var/log/nuslack-logs# cd previous/
root@fse-team-11:/var/log/nuslack-logs/previous# ls
backup-04-09-20-20-53-36-1.log.gz  backup-04-09-20-21-51-17-1.log.gz
backup-04-09-20-21-46-46-1.log.gz  backup-04-09-20-21-51-18-1.log.gz
backup-04-09-20-21-46-47-1.log.gz  backup-04-09-20-21-51-19-1.log.gz
backup-04-09-20-21-46-48-1.log.gz  backup-04-09-20-21-51-20-1.log.gz
backup-04-09-20-21-46-49-1.log.gz  backup-04-09-20-21-51-59-1.log.gz
backup-04-09-20-21-46-50-1.log.gz  backup-04-09-20-21-52-58-1.log.gz
backup-04-09-20-21-46-51-1.log.gz  backup-04-09-20-21-52-59-1.log.gz
backup-04-09-20-21-46-52-1.log.gz  backup-04-09-20-21-53-25-1.log.gz
backup-04-09-20-21-46-53-1.log.gz  backup-04-09-20-21-54-03-1.log.gz
backup-04-09-20-21-51-12-1.log.gz  backup-04-09-20-21-54-13-1.log.gz
backup-04-09-20-21-51-13-1.log.gz  backup-04-09-20-21-55-17-1.log.gz
backup-04-09-20-21-51-14-1.log.gz  backup-04-09-20-22-13-28-1.log.gz
backup-04-09-20-21-51-15-1.log.gz  backup-04-09-20-22-13-51-1.log.gz
backup-04-09-20-21-51-16-1.log.gz  backup-04-09-20-22-13-58-1.log.gz
root@fse-team-11:/var/log/nuslack-logs/previous#
```

fse-team-11

fse-team-11 [Version - 17.5.6] Server Monitor

Summary Processes CPU Memory Disks Network Plugin Integrations Checks Sys Logs **AppLogs** More

Last 24 Hours

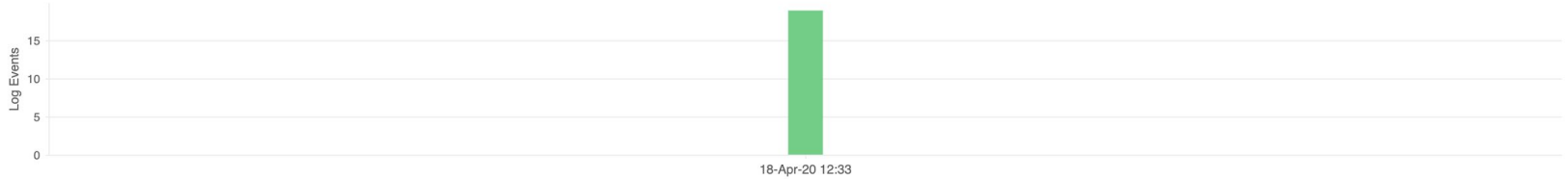
Add Custom Tab

Valid query. Press space to add more conditions.

146.99 MB/5 GB logs collected | Show Discovered Log Types | Disable AppLogs

```
logtype="Log4J2" and loglevel="ERROR"
```

Apr 18, 09:03 - Apr 19, 09:03
Last 24 Hours



Time	$Datetime:date$ $LogLevel$ $location$ $message:string$
Apr 18, 12:40:14:000	2020-04-18 19:40:14 ERROR UserServiceWithGroupsImpl:108 - Group already present with name: ThisIsANewGroup
Apr 18, 12:40:11:000	2020-04-18 19:40:11 ERROR UserServiceWithGroupsImpl:108 - Group already present with name: newGroupJustUsers
Apr 18, 12:40:11:000	2020-04-18 19:40:11 ERROR UserServiceWithGroupsImpl:108 - Group already present with name: newGroupMods
Apr 18, 12:40:11:000	2020-04-18 19:40:11 ERROR UserServiceWithGroupsImpl:100 - A Group called no users could not be created without any members
Apr 18, 12:40:08:000	2020-04-18 19:40:08 ERROR FriendController:49 - Tried to create a Friend that already exists from test1 to test2
Apr 18, 12:40:06:000	2020-04-18 19:40:06 ERROR ChatEndpoint:348 - null
Apr 18, 12:40:05:000	2020-04-18 19:40:05 ERROR UserServiceWithGroupsImpl:100 - A Group called ng2 could not be created without any members
Apr 18, 12:40:05:000	2020-04-18 19:40:05 ERROR ChatEndpoint:491 - The group must contain at least one member.
Apr 18, 12:40:05:000	2020-04-18 19:40:05 ERROR UserServiceWithGroupsImpl:158 - No entity found for query
Apr 18, 12:40:05:000	2020-04-18 19:40:05 ERROR UserServiceWithGroupsImpl:158 - No entity found for query
Apr 18, 12:40:05:000	2020-04-18 19:40:05 ERROR ChatEndpoint:491 - null
Apr 18, 12:40:05:000	2020-04-18 19:40:05 ERROR ChatEndpoint:272 - null
Apr 18, 12:40:05:000	2020-04-18 19:40:05 ERROR ChatEndpoint:272 - null
Apr 18, 12:40:05:000	2020-04-18 19:40:05 ERROR ChatEndpoint:334 - SecretMessage getting wrong amount of arguments
Apr 18, 12:40:05:000	2020-04-18 19:40:05 ERROR ChatEndpoint:348 - null
Apr 18, 12:40:05:000	2020-04-18 19:40:05 ERROR ChatEndpoint:348 - null
Apr 18, 12:40:04:000	2020-04-18 19:40:04 ERROR ChatEndpoint:259 - null
Apr 18, 12:40:04:000	2020-04-18 19:40:04 ERROR ChatEndpoint:272 - null
Apr 18, 12:40:04:000	2020-04-18 19:40:04 ERROR ChatEndpoint:234 - null

PROCESS AND TEAMWORK

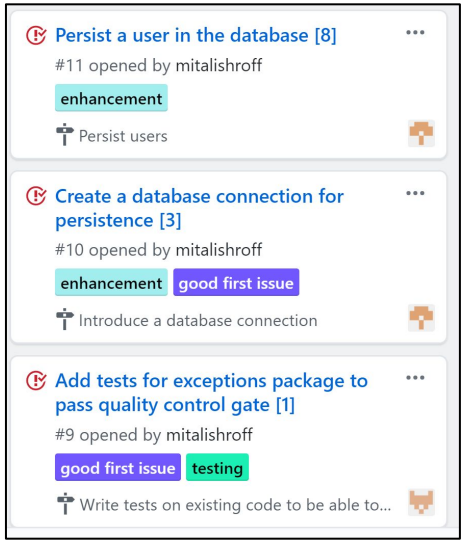
Our process - Scrum

1. Create Product Backlog and System Design

2. For each sprint

    a. Story Grooming and Sprint Planning

    Project Board



Our process - Scrum

2. For each sprint (continued)
    b. Sprint
        i. Working on deliverables
        ii. Holding update meetings /posted standups to slack
        iii. Test features

Our process - Scrum

2. For each sprint (continued)

    c. Sprint Retrospective

    d. Sprint Review

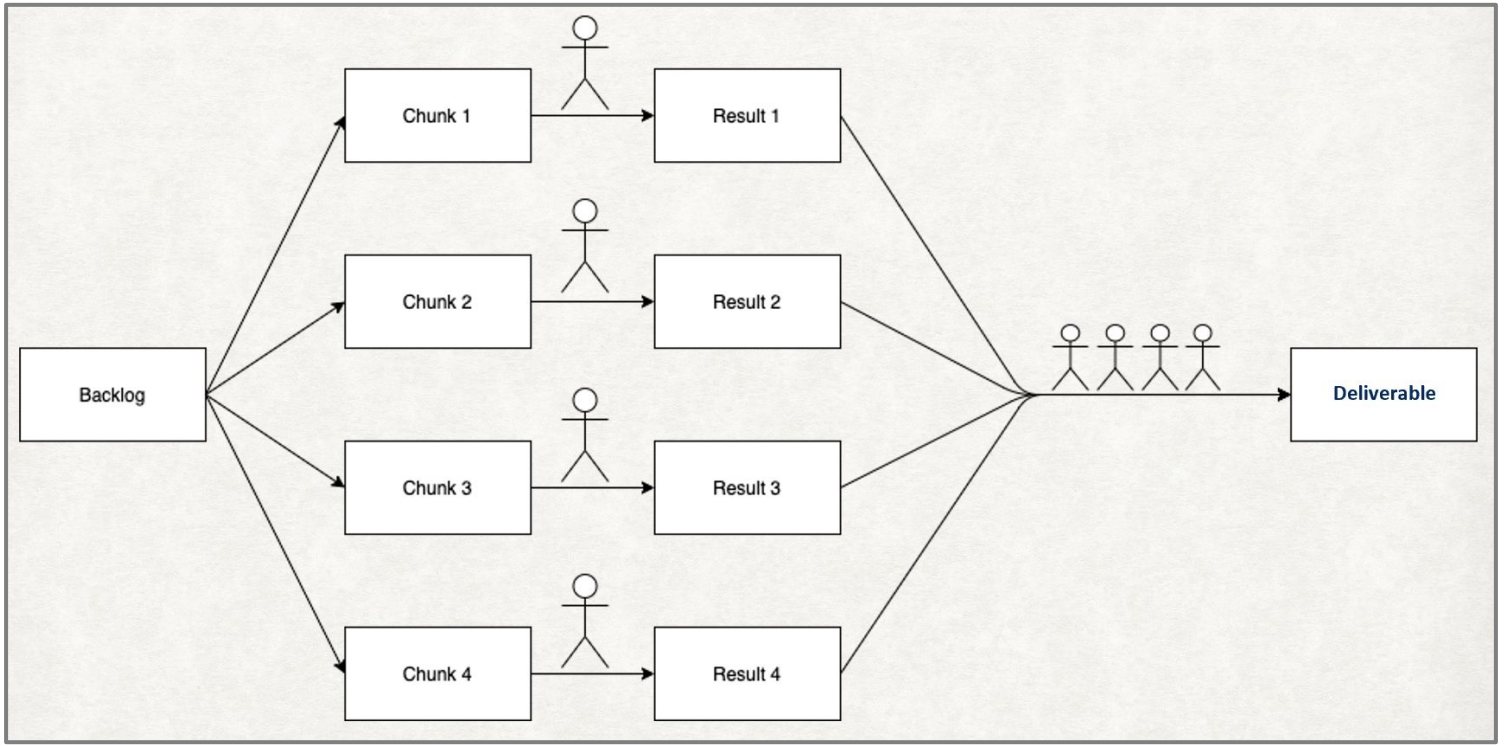
3. Refining end product

    a. Improving test coverage

    b. Removing code smells

    c. UI update

Our workflow



Created by Yanchi Li

Working as a team

- Collaboration
- Availability
- Self-organizing
- Providing good feedback
- Knowledge-sharing
- Dedication

Working as a team

Collaboration	Availability	Self-organizing
Providing good feedback	Knowledge-sharing	Dedication

Working as a team

Collaboration	Availability	Self-organizing
Providing good feedback	Knowledge-sharing	Dedication

Working as a team

Collaboration	Availability	Self-organizing
Providing good feedback	Knowledge-sharing	Dedication

Working as a team

Collaboration	Availability	Self-organizing
Providing good feedback	Knowledge-sharing	Dedication

Working as a team

Collaboration	Availability	Self-organizing
Providing good feedback	Knowledge-sharing	Dedication

Challenges faced (and overcome)

- Switching from in-person to online communication was a challenge
  - Everyone was flexible and invested in the project so we could transition seamlessly
- Pushing code towards the end of the sprint.
  - We decided to have a hard deadline a few days prior, for merging code into the master branch

Challenges faced (and overcome)

- Merge conflicts due to changes on the same files
    - We found that the best way was to resolve them together with input from the concerned parties
- Could have kept up with the rigorous testing we started out with
    - We brought our code coverage back up, and removed code smells, improved documentation before the final submission

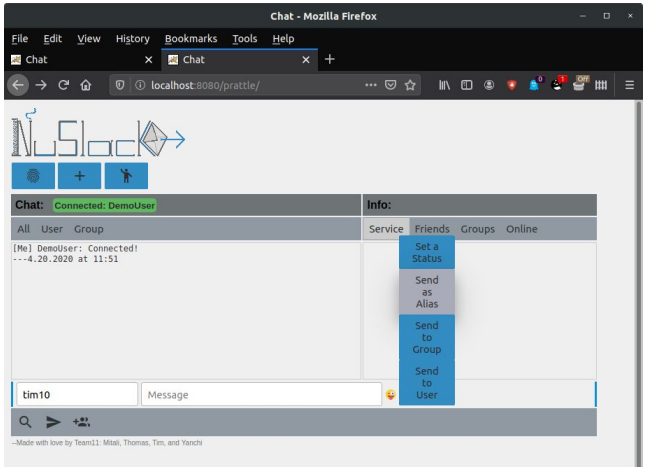
Things done well

- Using Scrum processes well
- Setting up good practices from the start
- Scrum masters keeping track of progress, but trusting the team
- Team members volunteering to take on stories, helping each other
- Going above and beyond

TECHNOLOGY TRANSFER

TECHNOLOGY TRANSFER

- Core functionality
- System on remote server
- Easy-to-use user interface
- System demo video
- System logging & Javadoc



NEXT STEPS

- Message translation
- Send files/pictures
- Message forwarding/recalling
- Hashtag
- Polls in a group

Thanks!