Final-Report for Team-2-F19

Dec 8th 2019

-Section 1: Summary of project goals:

The goal of this document is to provide a detailed overview of Diglett Chat Application. This section will attempt to cover the developing goals and the ambition of the team. As described in the SRS document, the Diglett Chat serves the purpose to create the next big thing, a service where people can message each other or groups effectively and securely. Users should be able to retrieve or initiate conversations easily, have information about their communications partners at their fingertips, find people, and the like - all independent of any particular device. It doesn't matter if they are using a personal or public device, the system should work as if they are in their office or at home. So at the core, we are connecting users and transporting communications for them. We've put together our priority list around our users. The functionality goals to be accomplished are listed as follows:

- Users can:

1. Be individuals or join one or more groups
2. Invite other users to join their groups
3. Log into the system by creating a username and password, or using a husky account
4. Have personalized icons and change icons at any time
5. Know if their friends are in the same group
6. Set their status as online, invisible, or do not disturb
7. Search for other users/groups
8. Follow other users/groups
9. Send messages/emojis to other users and reply to messages
10. Send messages/emojis in a group and reply to messages
11. Purchase services where someone who is the originator of a message could find out where it goes
12. Search message history for a particular piece of message/emoji.
13. In a group, choose to reply to the message-sender or to a subset of the group
14. Keep their messages stored for a given period
15. Purchase services for keeping their messages stay longer
16. Type English as the default language, more languages such as Spanish, French, and Chinese will be added in the future
17. Attach video, image, and music links to messages
18. Pay for one-time pad service, meaning that all messages sent and received on this particular pad will be self-destructed after a given period
19. Recall/delete the message
20. See messages with time-stamps queued up in the correct order that other users send them

- Groups can:

1. Have one or more users to become moderators

2. Moderators can evict someone or block someone from sending messages. Moderators have the right to approve or disapprove the invitation group members send to other users/groups. Only moderators can accept the invitation to merge their group with another.
3. Set password for users to join
4. See replies from all their members

Besides these, several non-functional goals are stated as well:

1. Scrum Master assigns tasks to each team member according to their field of expertise on Jira.
2. There should be at least one team meeting per week.
3. Accomplish a total of 80% code coverage at the end of each sprint.
4. Allow users with a middle-school education level to handle most of Diglett chat functionalities after reading the user's guide.
5. Deliver a functional web chat application before the semester ends.

-Section 2: Overview of the result

- Functionalities that we achieved at the end of final sprint：

1. Sending message
2. Receiving message
3. Delete/recall message
4. Reply group message
5. Store message
6. Find other people or groups

- Performance promises that have been realized：

1. Support at least 500 simultaneous users
2. Send messages within 5 seconds

To conclude, we achieved most of the features listed in SRS, as one can tell from the attached form.

Specific requirements

- 3.1 External interfaces
  - 3.1.1 User Interfaces ---------- ✓
  - 3.1.2 Communication interface ---------- ✓
- 3.2 Functional Requirements
  - 3.2.1 Sending message ---------- ✓
  - 3.2.2 Receiving message ---------- ✓
  - 3.2.3 Delete/recall message ---------- ✓
  - 3.2.5 Support media in message ---------- ✗
  - 3.2.6 Message Storage ---------- ✓
  - 3.2.7 Expose message to the user ---------- ✓
  - 3.2.8 Support different character sets -------- ✓
  - 3.2.9 Translate message ---------- ✗
  - 3.2.10 Security ---------- ✓
  - 3.2.11 Find people in a user's circle ---------- ✗
  - 3.2.12 Find other people or groups ---------- ✓
  - 3.2.13 Follow groups ---------- ✓
  - 3.2.14 Forward a media message---------- ✗
- 3.3 Performance requirements ----------✓

- Additional Features/Functionalities that will be implemented in the future :

1. Support media message
2. Translate message
3. Find people in the user's circle
4. Forward a media message

-Section 3: Team's development process

We will discuss the team's development process sprint by sprint.

For Sprint 0, our focus was to get the SRS done and be as specific as we could about the product requirements and designing-details. The team delivered a well-crafted SRS which is available in our team repo. Since this was the first week of the team project, not too much coding was involved. Jiahao was elected as the scrum master for the next sprint to lead the team.

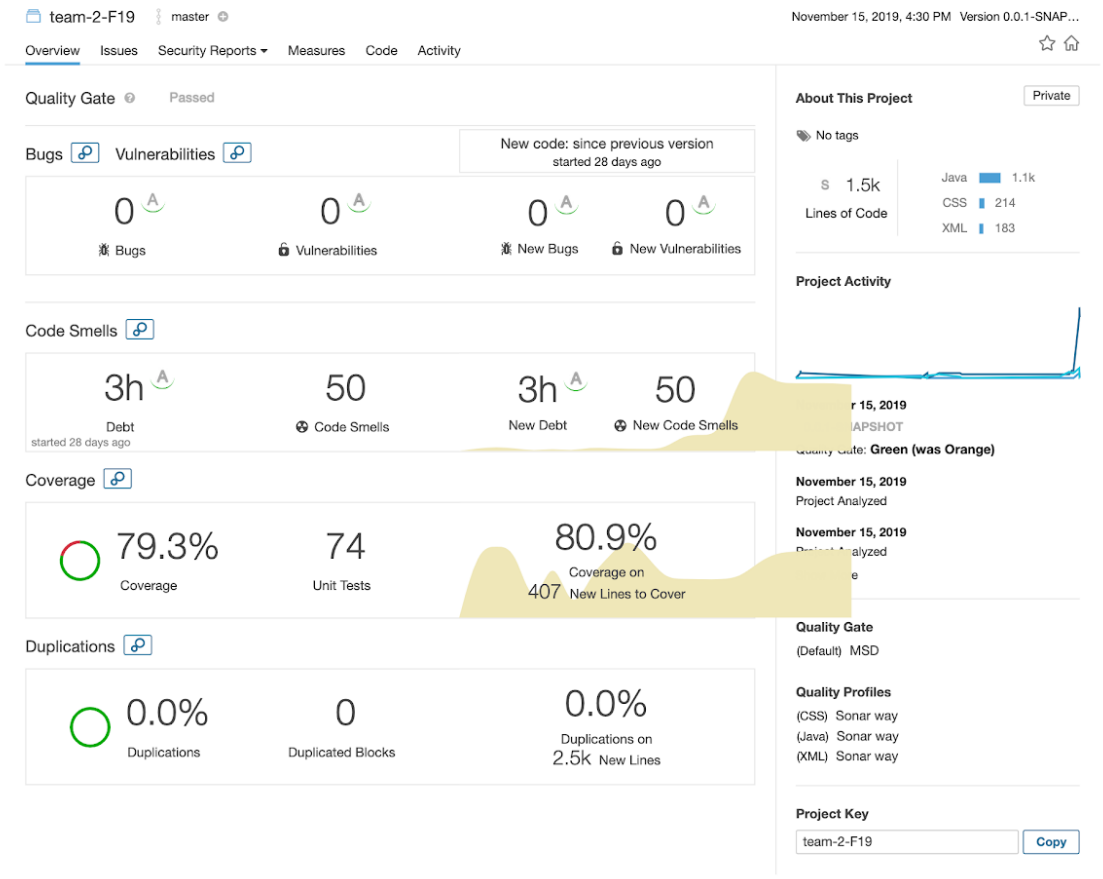
For Sprint 1, the team faced some issues when trying to get code coverage beyond 80%. What we did was to set up Tomcat, Prattle and deploy the website on the AWS server. Weihan deployed our chat app on AWS and drew the system architecture. Chihao contributed a lot to test writing and he helped us pass the 80% code coverage requirement. He also drew the UML diagram of user-cases. Chen drew the UML diagrams defining classes and methods. We also added test cases to

the source code to pass the quality gate on SonarQube. Each member is now more familiar with how Jenkins works. We did not modify the source code a lot because we had plenty of discussions on how to design the UML diagram. The team would like to ensure a solid system architecture is built before implementing random code. We missed the deadline to deliver the report because of insufficient test coverage. We felt a little frustrated but we encouraged each other to do a better job for the next Sprint. Chen was elected as the scrum master for the next sprint to lead the team.

For Sprint 2, we implemented the user login via username and password functionality. We achieved the individual to individual communication. We decorated our web application with an abundance of CSS and JS to make it look more attractive. We attempt to deploy our SQL database to store user information so that most of our functionalities could be achieved. We plan to implement the function where an individual will be able to communicate with the whole group. And the group should have moderators who have the privilege to ban and invite people. Weihan did a great job implementing the individual to individual communication functionality and adding a timestamp so messages show with a correct timestamp. She found many useful tools to help the rest of the team understanding web-sockets. She talked a lot about the concepts during the sprint review. This helps a lot because it builds a solid foundation for our future development and we could extend her contribution to other functionalities such as individual to group communication. Chihao was elected as the scrum master for the next sprint to lead the team.

For Sprint 3, we achieved many goals both on the front-end and back-end. For the front-end, which Chihao and Jiahao are responsible for, we implemented the UI of chat-room, friend-list, and login-form. We wrote some test cases for the JavaScript file. For the back-end, my teammates successfully connect our application to an online database, thus allowing user's information to be stored. I designed and drew the ER-diagram for our database. For now, the database is connected to our back-end application, but to directly talk to it via the UI which is the front-end, some problems persist. The friend status("online", "offline"), chat history, a timestamp will be added into our UI features. Weihan and Hao did a great job implementing the individual to individual communication functionality and adding a timestamp so messages show with a correct timestamp. Chen was helping Chihao to establish a connection between the database and our application. Chen also learned some Jest to help us write tests for the Javascript file. Hao was elected as the scrum master for the next sprint to lead the team.

For the last sprint which is happening right now, Sprint 4, we delivered our final presentation on Dec. 4th. Each member had something to say about the project and it was a successful presentation. The team continuously works on testing some of the previously-mentioned functionalities as well as adding some trivial minor functionalities to our final product. The purpose is to deliver a user-friendly and bug-free application. As one could tell from the attached images, the test coverage was constantly improving as well as the code quality.



The development process wasn't easy. We here would like to show our appreciation to the responsible and helpful teaching assistants, Sibendu and Vaibhav, for any assistance they gave us either during sprint review or during office hours. Not to mention that Professor Weintraub gave amazing lectures throughout the whole semester which helped us understand the core of Agile Software Development and the significance of the Scrum process. I believe this is an influential class which more or less benefits the student's career path as they enter this industry soon or later.

To summarize, our team has regular meetings twice a week. During the meetings, we:

1. Evaluated workload and estimated deliverables
2. Discussed and categorized tasks
3. Addressing problems that were posted on Jira or via email and message

We have two front-end developers and two back-end developers. The Scrum master carried the same responsibility each sprint which was:

1. Coordinate meetings
2. Distribute tasks based on each individual's field of expertise
3. Encourage teammates
4. Help team members as needed

To build, test, and promote processes, we use Jenkins and Maven to

automatically build, test and run the code. We did have shortcomings and faced some challenges:

1. No web developing experience
2. Steep learning curve
3. Spend plenty of time learning tech-skills (WebSocket, AJAX, JSON, JEST, etc)

-Section 4: Retrospective of the project

Looking back to the beginning of the development, five individuals with different sets of knowledge gathered together to accomplish the same goal. We learned that Agile/Scrum is generally preferred to Waterfall process model because it is adaptable to changing needs and priorities. The sprint time-box model defines short development iterations after which priorities and tasks may be added, adjusted, or removed before the next sprint. Each sprint produces something that can be deployed. This creates an opportunity to gather feedback, which can be fed into future sprints. It also creates value for the client because the time spent waiting for something useful is kept reasonably short.

Get to know the difference between functional testing and structural testing. Functional tests identify missing functionality at an input-output level. Structural testing emphasis whether the elements within a program are covered during the execution of the system to illuminate which parts contributed to the output. In other words, functional testing is done without knowing the internals of the system under testing and structural testing is done knowing these internals.

We have never been involved in an object-oriented-design team-project that lasts for the entire semester. Not only does it take time, but also requires consistent team-communication. We encountered some problems such as two people doing the same function due to lack of communication; Scrum master not able to attend the meeting due to interview-preparations, etc. However, we delivered a functional application eventually. “More hands make for lighter work.” “Two heads are better than one.” “The more the merrier.” These adages speak to the potential group projects have, and we do experience an overall more productive, creative, and motivated development process.