

Retrospective Document

Sprint-<2>

Work & Test Progress

List the milestones planned for this sprint along with their completion percentages.

(Please do not list the details of workpackages/tasks.)

We had 3 main milestones to achieve for this sprint;

- a-) Implementing a messenger system to provide communication between users.
- b-) Designing and creating notification modules for messenger system (phase 1),
- c-) Creating a profile page interface (and some other integrated modules to enhance the visuals)

While main functionalities for the messenger system has been implemented, this milestone still lacks some minor functions and visual improvements.

Completion percentage: %90

After some study, phase 1 for notification milestone has been successfully implemented. It is important to note that phase 1 covers the notifications coming from instant messages between users. This will be made to involve all kinds of notifications (upon an item upload with desired interests/needs, upon an item upload from a followed user etc.)

Phase 1 completion percentage: %100

User profile page is successfully implemented as well (it involves a navigation drawer (sliding menu), a connection to the main profile etc.)

Completion percentage: %100

List the tests finalized as part of the milestones planned for this sprint.

For milestone a,

-We have first created the appropriate schemas for both users and messages at the server-side and made sure that these schemas are valid and work as intended.

-We have then proceeded with a single user sending messages to his/her corresponding message array shared with the person whom he/she sends the message to, and made sure that these messages arrive successfully, and in order.

-We have set up the table rows and list views at the front-end(basicly we have set up the UI for messenger system) and tested them by filling random strings which share the exact schema from the server-side.

- Finally, we tried sending messages from one client to another with Google Cloud Messenger with the information coming from server-side, and succeeded. (Incoming data was correct and was according to schema, and layouts behaved and expanded as intended)

For milestone b,

-We first tried sending data from server-side to front-end for to ensure notifications are processed accordingly whenever datum arrives from an external source.

-We have then started sending messages instead of random and plain data, and made sure that everything behave as intended.

-We have then ran the application in the background and tried sending messages from another client to ensure the application does not get notifications only when it's open.

-Our final test was to log out from the application and then run it in the background and check whether it still produces the necessary notifications upon message arrivals, although our

first attempt was a failure, we have gone to the bottom of the problem and made the necessary changes. It succeeded.

Team Progress

For this sprint, we have mainly worked together. We have held 8 meetings in total, both for development and producing new ideas regarding this sprint's milestones and routes to follow in order to achieve these milestones. In general, it would be fair to say that we have shared this sprint's workload equally among team members. Our milestone completion percentages and each individual's contribution to the respective milestones are as follows:

Milestones	Completion (%)	Süleyman Arıkan	Mehmet Emin Küçük	Efe Erdil	Osman Erdem Tülü
Implementing Messenger System	90	20	25	25	20
Implementing Notification System(Phase 1)	100	40	30	30	X
User Profile and its related tools	100	10	20	20	50

Left-overs (Backlog)

From the previous sprint, we had problems retrieving multiple images and their metadata from the server-side due to some variable scope-related problems at the back-end. We later have figured out that the problem arose not because from scope-related issues, but because of the working mechanism of NodeJS(we haven't written any callbacks so NodeJS was not waiting the related function to process I/O information and jumps right to return statement, hence, empty list. We have solved this problem by adding callbacks so that NodeJS waits for the

function to complete all its operations before returning the list). And for this sprint, we still have minor improvements to make for the user profile and lacking functionalities at the messenger system, all of which are surely taken care of by the end of the third sprint.

Next Sprint

Our milestones for the next sprint are as follows:

- Notifications Phase 2(whole notification system will be implemented.)
- Integration and Refactoring(All functionalities will be transported to the new server system)
- Recommendation(Items will be listed according to user's activities.)

No comments.

Assistant's Evaluation

Assistant's (Team Leader's) comments regarding to this completed sprint.

Supervisors' Evaluation

Supervisor's (Team Leader's) comments regarding to this completed sprint.