Blueatt

Team 35 - Sprint 1 Retrospective

Justin Boudreau, Michael Enright, Moon Sun Hwang, Shulin Ye

What went well?

In Sprint 1, we finish 13 of 14 user stories. And within the finished ones, the issues we have are linking different parts and getting the bluetooth API to work. Also, we changed our approach in the middle in terms of the programming language and platform. Other than those, we manage to finish the individual functionalities from almost all user stories.

User Story #1

-As an instructor, I would like to store student's information in the software

We got this all of this to work, a UI was created to select a csv file can be selected and it will be parsed properly and then put into a database query to store the information.

User Story #2

-As an instructor, I would like to get a reading of a student's attendance on a specific date

We completed this user story, we had a proper UI working, although it was not connected to the database.

User Story #3

-As an instructor, I would like to select which class is in session

The UI was created and allows the instructor to select the class that is to be scanned for attendance. This gets put into a query to the database to pull the device IDs from the roster for the given class. These are then compared with the scanned device IDs and the ones that match up have their attendance recorded to the database.

User Story #4

-As an instructor, I would like to connect the scanner to a computer quickly

We have some issues with the java library for bluetooth and we finish the testing functions.

User Story #5

-As an instructor, I would like to select the intervals to periodically check for bluetooth signals

The instructor can select in the UI the intervals that would need to be set to scan for devices in the course selected. This will then recorded the present devices and save the attendance.

User Story #6
-As an instructor, I would like to view the percentage of attendance of each lecture

We get this user story to work in UI and backend separately. Our team is working on integrating UI and backend for this function.

User Story #7
-As an instructor, I would like to select the way to get the information, such as emails or instant report

We finish this user story successfully, the options are able to be selected in the UI.

User Story #8
-As an instructor, I would like to view the tendency of a students' attendance during the semester graphically

The instructor is able to select a course to see the attendance throughout the semester. The information is pulled from the database and displays the statistics to the UI.

User Story #9
-As an instructor, I would like to get notifications on if the scan is successful or not

After the instructor starts scanning for devices a timer is set and if it takes to long it times out and prompts an unsuccessful scan. If it is unsuccessful then a message is shown in the UI to allow the instructor to try and scan again.

User Story #10
-As an instructor, I would like to only receive information on students in the class

When scanning for devices starts the course's device ID list for the roster is pulled from the database. This list is then compared with the list of devices scanned to ensure that attendance is taken only for students in the course in session and not random devices. This is then recorded to the database

User Story #11
-As an instructor, I would like to manually add/delete/modify student's attendance

The instructor is able to select a class, day, and student in which to update the current attendance for them. This is then put into a query and stored into the database and a notification of success or failure is displayed in the UI.

User Story #12
-As an instructor, I would like to be able to add or remove a student from a class

Through the UI the instructor is able to select a student in the course to be removed from the roster. This prompts a message asking if they are sure as it will erase all records for the student in that course. This is then queried to the database where the changes take effect and a message is displayed to the UI of its status.

User Story #13
-As an instructor, I would like to be able to edit student's information if necessary

The instructor is able to select a specific student from UI and edit all his information, also he can do the same in the backend.

What did not go well?

First of all, we didn't plan this sprint that well. We have more user stories than we can handle considering we're a team of 4 members and also the tasks assigned to each person is not focused enough, so we have some redundant work and having trouble linking things. Moreover, we didn't have enough group working time to tackle some technical issues such as bluetooth API and the interface implementation. Completing tasks individually was not a problem but combining the parts was not able to be finished due to lack of time and communication.

User Story #14
If time permits, as an instructor, I would like to allow software to scan automatically at a given time everyday

We didn't have time to get to this story in Sprint 1, we planned too many user stories to implement and overestimated our time but it was presented as a "if time permits".

How should you improve?

1. Better planning

    a. Pick a more suitable workload and choose quality over quantity
    b. Give tasks to team members based on type of implementation (ex. front end)
2. More group working time
    a. We need to figure out some difficult technical issues together and keep things linked as we move forward
    b. Communicate how different parts of the project will interact with each other
3. Pay more attention to details
    a. making sure all parts are implemented correctly and to do the unit testing
4. Assign coherent and consistent work for individuals
    a. Previously, we separated each user story into steps and divided work within a single user story. However, merging all the work to make the user story work was complicated with this method.
    b. In the next sprint, we will put more effort in assigning the work so that one person is responsible for the certain relative parts of the program. This will allow us to easily complete user story one by one.