Using focus groups allows team 10 to develop a product that meet customer requirements. It helps team 10 to stay on top of understanding the customer perspective.

Participant background, brief meeting process and discussion results are as follows. The full Focus Group Report which includes focus group process introduction and more detailed information is in appendix[NUM].

**Participant Background**

Computer science students of Year 1, year2 and year 3 were invited to the focus group meetings respectively. To guarantee the objectivity of the outcomes of the discussion, the meetings were carefully arranged that participants in the same group were unfamiliar with each other and team member.

**Discussions**

**Year 1 group**

Due to year 1 participants’ lack of relevant background knowledge, the discussion results of this focus group are less than expected, but there are still some valuable advice.

**Pseudo-code game**

Participants mentioned that writing pseudo-code was hard for beginners, a game such as dragging pseudo-code blocks may help them to learn the ideas of writing sorting algorithms.

**Time complexity**

Participants suggested that the software can provide a time complexity comparison of different sorting algorithms. This idea is presented in the requirements.

**Year 2 group**

Year 2 students’ discussion was out of our expectation. Many innovative ideas were raised in this focus group.

**Fun and game likely VS functionality**

Participants showed a strong preference for functional software compared to the game likely one. They would like the software to have a single purpose, without those distracting things. This result helped the team decide the requirements.

**Method of selecting modules**

Two methods of selecting modules were introduced to participants. The first method was that users could not access the next sorting algorithm’s learning module unless they finished the one before. The other one was free choosing. Participants preferred to choose an algorithm module freely instead of “unlocking” the next module by finishing the current one. This result overturned team members’ initial idea, and the first method is abandoned.

**Displaying code**

Participants thought pseudo-code is enough for learning since it gives a logical idea of sorting algorithms. Besides, pseudo-code is more friendly to beginners. Source code can also be presented, they claimed, and for language, C, Java and Python are preferred.

**Progress bar**

After presenting and explaining our preliminary prototype, participants showed great interest in the progress bar we designed. They thought that a display of where they were in the learning program was clear. Besides, they suggested us to design history and reset features.

**Platform preference**

Following options were provided to participates: PC, mobile app, WeChat mini program and website. The result is that the PC was more popular than others. Participants also proposed that PC would be a more suitable platform for notes’ quick export function.

Besides, participants also responded positively to the following functions:

1.Multiple language support.

2.Quick export of brief notes, i.e., a file includes an introduction and important points of a sorting algorithm.

3.Show efficiency and time complexity of algorithms.

4.Apply breakpoints.

5.A user guide to give users a quick look at how to use this software was highly recognized by participants.

6.Exercises.

**Year 3 group**

Year 3 students’ discussion focuses more on correctness.

**Difficulties in learning correctness**

Participants explained that they did not expect that correctness would have a connection to mathematics and predicate logic. It would be helpful to provide math related information. They hoped to see animations with logical explanations and sufficient examples.

**How to help understand correctness**

Participants suggested that a hierarchical learning method by introducing concepts step by step would help. For instance, concepts of termination and correct output can be illustrated respectively. In addition to normal examples, counter ones shall be provided for better understanding.

**How to make learning interesting**

One of the participants raised a proposal that the software could be a listener who listens to and records the user’s own explanations of algorithms. Users can review their previous ideas by replaying these recordings.

**Result**

Focus groups help to acquire users’ feedback about the product being built. Many requirements related problems were confirmed. It also helps confirm the validity of questionnaire.