



# COMP2043.GRP Interim Group Report

## Project 20: Animation of Sorting Algorithms and their Correctness

Submitted Dec 2020, in partial fulfillment of  
the conditions for the award of the degree **BSc Computer Science**.

**Team 10**

**Supervised by Dr. Heshan Du**

School of Computer Science  
University of Nottingham

Shiliang Chen (20125016), Ruizi Han (20125115), Yani Huang (20125915)  
Yijie Lu (20125931), Yuting Jiang (20126307), Yiming Tang (20126870)

We hereby declare that this report is all our own work, except as indicated in the text:

Signature \_\_\_\_\_

Date \_\_\_\_\_ / \_\_\_\_\_ / \_\_\_\_\_

## Abstract

Team 10 has worked on project 20, Animation of Sorting Algorithms and their Correctness, for approximately ten weeks. This project is to develop a software to animate the sorting process of sorting algorithms and help users learn them and their correctness. To achieve this, team 10 will conduct the whole software engineering process. Chapter 1 briefly introduces the entire project. The next chapter describes background and related research. In the third chapter, Requirement Specification, the information collected from the previous work is specified to context. Chapter 4 shows software requirements visually using graphic diagram with simple explanation. Team's initial design of the software is also included in this design chapter. After that, the implementation chapter concludes decisions the team has made when designing the software and presents the final prototype. Chapter 6 mainly describes encountered problems and preparatory work the team has done to prevent them. Finally, chapter 7 presents the project timeline and how schedule is managed.



# Contents

<b>Abstract</b>	<b>i</b>
<b>1 Introduction</b>	<b>1</b>
1.1 Motivation and Description of the Problem . . . . .	2
1.2 Aims and Objectives . . . . .	2
1.3 Description of the Work to Solve the Problem . . . . .	2
<b>2 Background Information and Research</b>	<b>3</b>
2.1 Background . . . . .	3
2.2 Survey of Existing Systems . . . . .	4
2.3 Market Research . . . . .	9
2.3.1 Focus Group . . . . .	9
2.3.2 Survey . . . . .	12
2.3.3 Interview . . . . .	14
2.4 Technical Research . . . . .	15
<b>3 Requirements Specification</b>	<b>21</b>
3.1 User Requirements . . . . .	21
3.2 System Specifications . . . . .	22
3.3 User Story . . . . .	24
<b>4 Design</b>	<b>26</b>
4.1 Use Case Diagram . . . . .	26
4.2 Sequence Diagram . . . . .	28

4.3	User Interface and Low-Fidelity Prototype . . . . .	31
<b>5</b>	<b>Implementation</b>	<b>35</b>
5.1	Key Implementation Decisions . . . . .	35
5.1.1	Platform . . . . .	35
5.1.2	Programming Language and Software Framework . . . . .	36
5.1.3	Development Tools and Methodology . . . . .	36
5.1.4	Integrated Development Environment . . . . .	37
5.1.5	Version Control and Team Management . . . . .	37
5.2	Results of Implementation . . . . .	39
5.2.1	High-Fidelity Prototype . . . . .	39
<b>6</b>	<b>Problems Encountered and Risk Management</b>	<b>49</b>
6.1	Problem Encountered . . . . .	49
6.1.1	Technique Issue . . . . .	49
6.1.2	Management Issue . . . . .	50
6.2	Risk Management . . . . .	52
<b>7</b>	<b>Time Plan</b>	<b>53</b>
	<b>Bibliography</b>	<b>53</b>
	<b>Appendices</b>	<b>58</b>
	<b>A Full Focus Group Report</b>	<b>58</b>
	<b>B Full Questionnaire Report</b>	<b>64</b>
	<b>C Questionnaire</b>	<b>72</b>
	<b>D Minutes</b>	<b>76</b>

# Chapter 1

## Introduction

Sorting is an operation that arranges a list of elements in order according to one or more standards. Sorting algorithms are systematic methods to achieve this. A set of chaotic numbers can become ordered and display in an ascending order after using a sorting algorithm. There are many sorting algorithms applying different strategies. They are valued in many areas, such as big data processing. A good sorting algorithm can save resources and time. In this case, sorting algorithms are essential when learning algorithms and programming. However, they are not easy to be fully understood and implemented by beginners. Animation is now widely used in educational area. Many researches have proved the effectiveness of algorithm animation in teaching and learning [23]. Though there are many existing similar software in the market, few of them are used among students, who are the main stakeholders for this type of software. Besides, many of them find difficulty when learning sorting algorithms. Therefore, developing a sorting algorithms animation software is necessary. Algorithms' correctness is an abstract concept, but useful for leaning, including this in the software could help advanced learners.

In this chapter, a brief introduction to this program will be presented in three parts. Section 1.1 introduces the motivation of doing this project and problem description. Team's aims and objectives are to develop an educational effective program and the animation process of sorting algorithms, more specific information is in [section 1.2](#), and section 1.3 describes work the team plan to do to solve the problem.

## 1.1 Motivation and Description of the Problem

Sorting algorithms, such as bubble sort, selection sort and merge sort, are basic algorithms for beginners. To better understand algorithms and related areas in computer science, learning sorting algorithms as the rudiments of algorithms is a great choice. Our project plan focuses on helping beginners learning sorting algorithms and their correctness. We have found that their correctness is not easy to understand for beginners because of how the algorithms work is not directly perceived. Research has been taken to show that visualising the sorting algorithm might help to solve this problem. Rudder et al. claimed that animations and visualisations could help present those abstract concepts in an easy-understanding way [15].

## 1.2 Aims and Objectives

The aim of this project is to build a game-like learning tool with the animation feature to achieve visualising sorting algorithms. The main picture is that the software will give a taste at the beginning by providing some easy tutorials. After that, it will provide several modules for users to learn different algorithm principles and their correctness in a simplified way. During the process, explanations and guides will be displayed with an animation. Finally, exercises will be provided for them to check whether they understand well enough.

## 1.3 Description of the Work to Solve the Problem

As planned, the Agile development methodology and **TDD** would be used to ensure everything is expected and under control. One month will be spent on the requirements and specifications. Interview, focus group and other requirements gathering activities will be taken. After that, four months would be spent on designing the software. Finally, two months will be spent to evaluate the quality of code, **UI** and functions by unit test and maintenance.

# Chapter 2

## Background Information and Research

To know more about situation in the existing market, background research has been done.

There are four sections in this chapter, 2.1 will present some collected background information. Analysis to existing similar software are included in section 2.2. Main stakeholders of this software have been confirmed as year 1 computer science students and beginners in learning programming. Other stakeholders are advanced learners who are interested in algorithm correctness and algorithms teachers. To gain requirements from them, focus group, a survey, and an interview to algorithm teacher have been accomplished. Brief reports are presented in 2.3 market research section. Furthermore, what is introduced in section 2.4, technical research was done to help the team make implementation decisions.

### 2.1 Background

Due to the abstract concept and theory, it is particularly difficult for beginners to learn and understand the algorithms. In order to solve this problem, the algorithm visualization technique was applied in the field of computer science teaching since the early 1980s [3]. According to Yu and Su et al., algorithm visualization technique explains how an algorithm is implemented through multimedia such as graphs, texts, colors, sounds, animations, and videos [21]. The main feature of algorithm visualization is that it can

dynamically demonstrate the data, operations, and semantics extracted from the program to learners. Therefore, the visualization technique is able to significantly improve the efficiency of the algorithm education process.

In 2012, Simoňák and his team released an algorithm visualization system named VizAlgo [17]. It is a portable platform based on a JAVA environment, and the plugin-based architecture of the system ensures its extensibility. Basically, it includes the visualization of sorting algorithms and some basic data structures. However, it also includes some limitations within this system, such as the limited support of advanced visualization features like dynamic changes in algorithm pseudocode reflected in visualization.

For this project, we focus on the sorting algorithm visualization. It seems that displaying the sorting process with the form of a tree like VizAlgo is not visual enough. In 2016, Simoňák proposed another algorithm visualization platform called Algomaster [18]. In this new system, back-stepping functionality for an easy algorithm stepping in both directions is developed, and a self-testing mode for automated testing of users' knowledge is available. That means the new platform increased interaction with users and partly improve the visualization features. However, according to the results provided by the experiment, there also exists some participants who were still not satisfied with the visualization part of Algomaster.

## 2.2 Survey of Existing Systems

The problem to be solved, which is demonstrated in the introduction, has several solutions to it already exist. However, different solutions are oriented to different groups of stakeholders. Research on existing systems mostly help to extract commonly achieved features and find aspects which can be improved. Key decisions of implementation can also be referenced from existing systems. Five existing systems on different platforms are introduced in this chapter.

### 2.2.1 VisualGo

#### Main Feature

VisualGo [22] is an online tool that can show visualizing data structures and algorithms through animation. In the sorting algorithm module, it provides animations, which shows how the disordered elements will be sorted using different sorting algorithms. After selecting random numbers for sorting, users can observe the sorting process and read the explanation using pseudo code at each step.

#### Advantage and Disadvantage Analysis

The advantage is that each sorting process could display by the animation. During the process, the explanation is showed at the bottom. Additionally, it gives the language preference option, which means learners from different countries can use it. However, this explanation is for users who learned the programming language before, but it is not suitable for someone who has no programming language experience. Additionally, the software lacks interaction with users because users can only observe the process but cannot have a chance to sort by themselves. Therefore, the knowledge could be difficult to absorb by users.

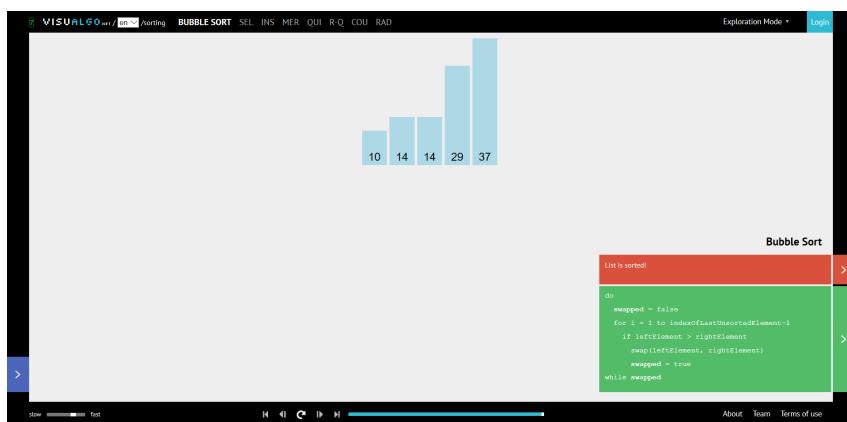


Figure 2.1: VISUALGO sorting module

## 2.2.2 Scratch

### Main Feature

Scratch [16] is a graphical programming tool developed by MIT's lifelong kindergarten team. It is mainly open to teenagers. The interface design mainly includes three parts: toolbar, operation platform, and output platform. In the operation platform, Users can create their program by dragging blocks. Each block contains a piece of pseudo-code that can make up for a completed algorithm.

### Advantage and Disadvantage Analysis

Under such a graphical programming environment, knowledge can be easy to learn. Children are happy to see the teaching combining with graphics. Additionally, interaction with users might strengthen because designers try to mix games and training. However, the guide needs to be obvious, which can refer to two issues. One is that the toolbar contains a series of blocks for choosing, but the guide of how each tool works is not provided amply. Meanwhile, creating an algorithm needs exploration, but it is a little hard without some tips. Probably adding some suggestions can be easy for promotion.

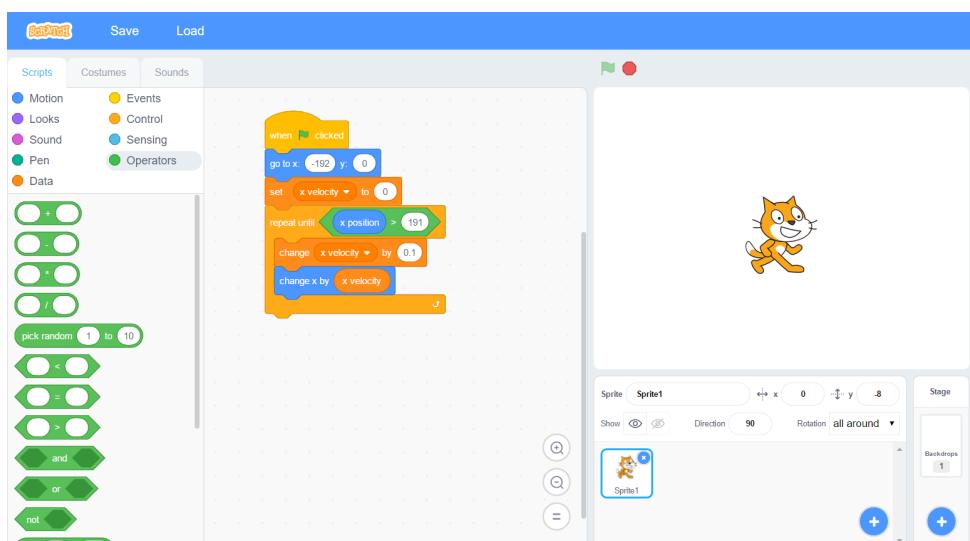


Figure 2.2: Scratch

### 2.2.3 WeChat MiniProgram

#### Main Feature

WeChat MiniProgram [24] can be accessed by scanning a QR code using WeChat. Since it is based on WeChat, it supports mobile devices well and no installation is required. As a light-weighted software, it only supports six basic sorting algorithms with a default input array of numbers. This software consists of a code part on the left and an animation part on the right. As for the left code part, it only supports code in JavaScript. For the animation part, a simple description of the current sorting algorithm is shown beneath the bars. Buttons are located at the top of the software. Choosing algorithms feature is hidden under the hamburger button in blue. The green play button will trigger the animation and the red one will reset the process.

#### Advantage and Disadvantage Analysis

Although this software contains limited features, they are all basic and practical. This also makes this software concise and easy to use. However, only providing real programming language is not friendly for newcomers. No English support would be another problem for users.

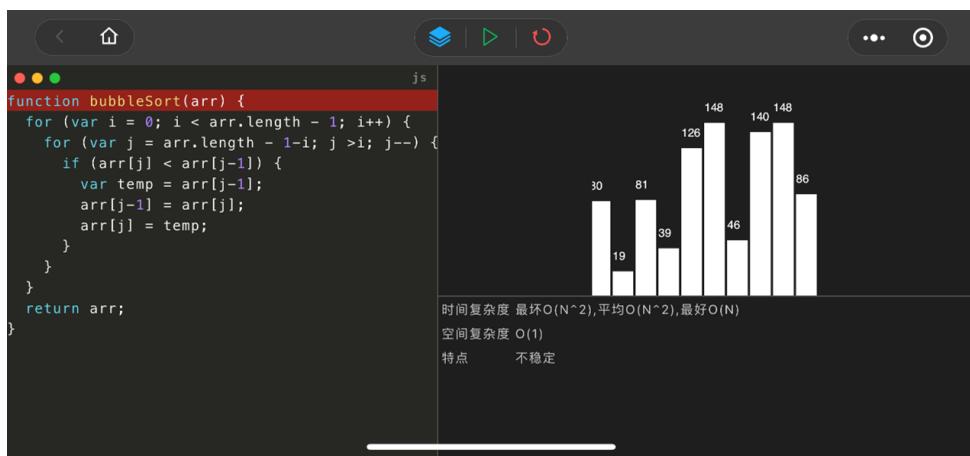


Figure 2.3: WeChat MiniProgram

## 2.2.4 Algorithms Explained and Animated

### Main Feature

Algorithms Explained and Animated [2] supports performance with iOS, so only iPhone and iPad can download and use this app. On the other hand, this provides the mobility of the learning process. In other words, users can learn or check algorithms anywhere and anytime without a computer.

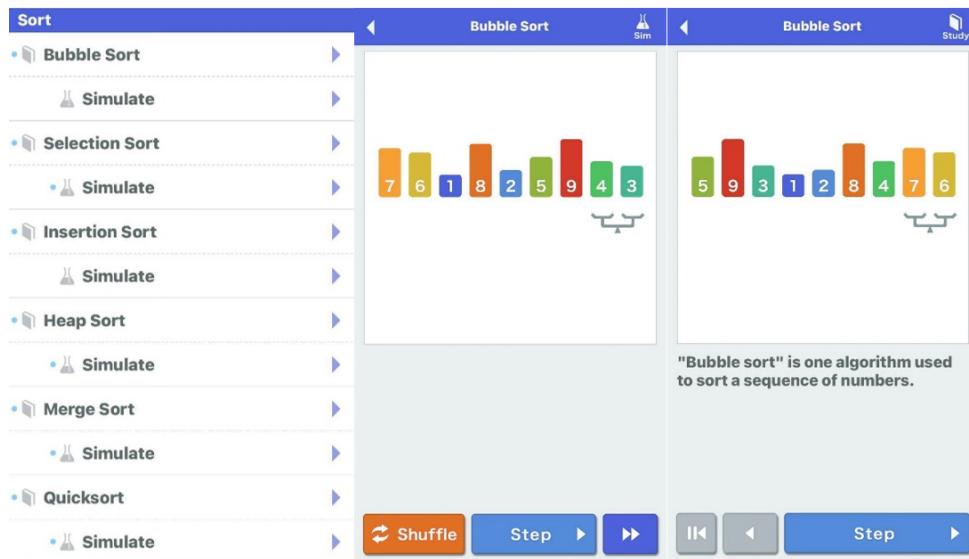


Figure 2.4: Algorithms Explained and Animated

### Advantage and Disadvantage Analysis

The structure of this app can be acceptable for newcomers. There are two parts specifically for two purposes. The one with the sorting algorithm name on it is for study purpose, i.e. the software introduces how a specific sorting algorithm works by both animation of bars sorting process and description below. Users cannot shuffle bars of numbers since it is a predefined animation. The other part is called simulate, which allows users to shuffle bars of number with three strategies and watch the animation either step by step or automatically. Moreover, a balance is under two bars being compared, which is straightforward for users to understand comparing strategies of different sorting algorithms.

There are also both limitations and advantages in terms of mobility. Due to the limited

display size of mobile devices, this app does not support any programming language. Users cannot set an array of integers on their own. However, language support is appropriate for this app. The language this app uses alters following the iOS system language, but users cannot switch language directly in the app.

## 2.3 Market Research

The software being built will finally be deployed into the market, and much of business or technical requirements resides in the minds of stakeholders. Therefore, market research is conducted to gather requirements and suggestions from stakeholders. Team 10 used three professional approachesFocus group to hear from stakeholders inner voice in a group meeting, survey to gather data from a big group of people and interview to listen to an individual's opinion. The results are demonstrated in this section.

### 2.3.1 Focus Group

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 A.

### 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

This part is divided into three sections according to the grade of participants.

**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.

**1. 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.

**2. 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.

**1. 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.

**2. 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.

**3. 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.

#### 4. 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.

#### 5. Platform preference

Following options were provided to participants: 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 introduction and important points of 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.

**1. 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.

**2. 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.

**3. 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.

**2.3.2 Survey**

The software we plan to design is oriented to students who want to learn sorting algorithms. However, details of learning method that students prefer remain unknown. Questionnaire is an efficient tool to collect both ideas and suggestions from target users. To gather requirements for the project, Team 10 released a questionnaire with eleven multiple choice questions and one open question. The questionnaire remained open for

seven days. The full questionnaire report which includes data, analysis, and more detailed information is in appendix B.

### Objective

1. Understand how well users know about sorting algorithms and their learning habits.
2. Gather requirements from target users.
3. Ask for suggestions for the functions we plan to design.

### Time

From November 10th to November 17th.

### Respondent

Students from the University of Nottingham, Ningbo, China.

### Questionnaire

Please refer to the appendix C.

### Result and suggestion

The participants in this survey are students from the University of Nottingham, Ningbo, China. Among them, 55% are from year 1, 11% are from year 2, 29% are from year 3, and 5% are year 4 students. The proportion of males and females is relatively equal. About 75% of the participants have varying degrees of understanding of sorting algorithms, while some know nothing about sorting algorithms.

To summarize the survey result, some important points drawn from this questionnaire are as follows:

1. It is suggested to provide a basic conceptual introduction and outline of algorithms for users who do not have any knowledge of algorithms.

2. Compared with the entertainment functions, participants preferred practical ones.  
It is suggested to reduce game mechanics and focus on developing functions that are more helpful for learning algorithms.
3. Some participants proposed that providing a function, which can share and discuss their thoughts with others, like forum, may help learning.
4. To ensure users with different levels of algorithmic bases can all gain knowledge from this software, it is suggested to provide hierarchical teaching functions.
5. For those who have examination requirements, offering more practice questions will attract them to use this software.
6. Many participants mentioned that they preferred clean and delightful interfaces.  
Therefore, GUI design is one thing that needs to consider.

### **2.3.3 Interview**

To bring Team 10 a different view of visualising sorting algorithms in teaching perspective, rather than learner's view, an interview was appointed with the lecturer teaching "Introduction to Algorithms" course in Year 1 (Y1 for short). We tried to gain information about how Y1 students learn sorting algorithms and teacher's ideas of how they design the course and why they do so.

#### **Module**

Introduction to Algorithms teaches Y1 students algorithms through functional programming, that is, everything in recursion and no loops provided. The lecturer shows us the slides he used for teaching, which is mainly about recursion. To help student understand the process, trace, which would show changes after each step is done, is taking a main part of the slides. The lecturer also stressed that most of the students don't ask questions to him.

### Existing Similar Products

Having been showed some existing products regarding to sorting algorithms and used them for a while, the lecturer points out that pseudo-code is not suitable for Y1 students. Instead, detailed explanation, like what, why and how, is expected in such a software. We should also notice that users shall be able to decide the list length. We need to provide different shuffle scheme instead of only randomise the array.

### Hierarchy

The lecturer thinks bubble sort is basic. Insert and merge sort are basic as well but should be introduced later.

### Platform

The lecturer prefers Windows.

### Other

The lecturer would be happy to use our software for teaching sorting algorithms, and he thought animation is suitable for displaying sorting algorithms.

## 2.4 Technical Research

The project is to develop software which can visualise sorting algorithms.

In this case, methods of how to achieve visualization will be declared at first. The programming language to be used in the project shall be adaptable to a graphical user interface (GUI) and able to respond to user request through the GUI. Platforms need to be suitable for both users' usage and developers' development. As a significant coding tool, the choice of IDE should also be carefully considered. Analysis of these three parts is presented as follows.

### 2.4.1 Visualization and Animation

When it comes to visualizing sorting algorithms, many practical design ideas are learned from L. Végh and V. Stoffová's past research [20]. Pseudo-code is more recommended than a source code when teaching algorithms because it describes and presents algorithms at a higher and logical level [5]. Therefore, displaying pseudo code in this software can help users understand algorithms at the abstract level and not get stuck in understanding any programming language's syntax problem. The connection between animation and other modules is also very important. For example, when the animation is doing swapping, the corresponding code should be highlighted [11]. Difficulties in understanding each step's code are different. The animation of an important part of algorithms should be slowed down, or a changing animation speed function should be added [20]. In that case, users can learn and understand easier.

### 2.4.2 Platform

While discussing the platform for developing this software, four platforms were come up, which are PC (macOS/Windows), a mobile terminal (iOS or Android), web and WeChat mini-app.

- **WeChat mini program**

**Advantage:** WeChat is a widely used application, users can access this software quickly and conveniently.

**Disadvantage:** The screen size for a mobile device is too small, so users might not be able to see the animation.

- **Web**

**Advantage:** Users can use this software easily using the web without installing anything. Besides, it can be used on any devices that have a browser and link to the network.

**Disadvantage:** User privacy might be a problem. 'User guide' part for new user and 'history' part will be included in this software, so if this software is going to be

built on the web, cookies must be needed [9].

- **PC**

**Advantage:** Users can access software by the local computer. There is no need to link to the network. Also, installation and employment are easy.

**Disadvantage:** The computer is not that easy-carry as mobiles and other portable devices.

- **Mobile terminal**

**Advantage:** Same as PC, the software can be used without network.

**Disadvantage:** Mobile's screen size is small. According to the questionnaire, users seem to prefer the PC.

### 2.4.3 Programming Language and Software Framework Overview

Three main programming language will be introduced below with both advantages and disadvantages.

- **Java**

In JDK, Java provides two basic tools for building a graphical user interface, which is AWT and Swing. JavaFX was once a component of JDK but is a third-party tool now. AWT was introduced in JDK 1.0 and heavily depended on the Windows platform, while Swing is more flexible and can be executed in multiple platforms with Java [13] [14].

**Advantages:** Java is a cross-platform language based on Java Virtual Machine. Both Swing and JavaFX take advantage of this feature, which allows the software to be easily distributed in multiple platforms [12]. Basic functions to build software are fully supported in those Java tools. As a back-end language, Java takes advantage of object-oriented and it is good at express the business logic of the front end. Its property of strongly typed also improves the efficiency of the programming process.

**Disadvantages:** AWT was introduced in JDK 1.0 and Swing is released in JDK 2.0. Both are old and using a coding style which develops the user interface with

built-in graphic components. It is not straightforward to see and adjust the layout [13] [14]. Besides, relevant discussions and resources are not sufficient, even on the Internet. Lack of references would be an obstacle to the team's progress. As for Java itself, the efficiency of compiling would not be satisfying.

- **Web framework solution with JavaScript or Java**

JavaScript is a programming language which is used with the Web. A web browser is responsible for parsing the Web code along with JavaScript and displaying interactive content to users. Web browser is a basic software on Windows, Mac, iOS and Android.

**Advantages:** Web solution allows accessing a web page on multiple platforms with the same contents. Therefore, it is a fully cross-platform solution to build software once but run everywhere. The web consists of two parts, front end and back end. The front end provides a user interface only using markdown language HTML5 [6], and the back end works as a server which listens to the actions that users take and handle these actions with programming language JavaScript or Java. The field of web software has been developing fast since the mobile market expanded in recent years [1]. Many third-party tools and open-source softwares, such as Spring framework, take place [19]. Online resources such as plugins and existing web structures and plentiful discussions are also helpful for building such software. What's more, a web application can be easily converted into a desktop executable, which can run on multiple platforms even without a browser.

**Disadvantages:** The web was taught in the previous semester, and only basic knowledge of it is mentioned. The communication between the front end and back end may lead to relatively lower efficiency compared to a pure programming language like Java [6].

- **C#**

Using C#, developers can create secure and robust software that runs in the .NET ecosystem. It is an object-oriented language that is introduced by Microsoft [10].

**Advantages:** The syntax is elegant and expressive [10]. As a commercial prod-

uct, C# is well supported by Microsoft, there are plenty of relevant tools and a fully supported online community built by Microsoft. Therefore, learning resources would be sufficient, and troubleshooting might be easy with other developers' help. Graphics programming is also well supported in C#.

**Disadvantage:** C# is a product of Microsoft and can only work on Windows. Team 10 has not learnt anything related yet. It would be a brand-new field, exploring it could take much time.

#### 2.4.4 Development Tools

Two development software will be analyzed below with both advantages and disadvantages.

- **Eclipse**

Eclipse was originally developed by IBM as the next generation IDE development environment to replace commercial software visual age for Java. It was contributed to the open-source community in November 2001. It was managed by the Eclipse Foundation, a non-profit software supplier alliance [4].

**Advantage:** Eclipse is a light-weighted IDE, which takes less storage than IntelliJ IDEA.

**Disadvantage:** It has incompatibility issues between versions, especially for plugins. As for JDK, when a new version of JDK launches, eclipse generally lags for a long time before a new version supports the new JDK.

- **IntelliJ IDEA**

IntelliJ IDEA is a product of JetBrains. It is an integrated environment for Java programming language development. It mainly supports intelligent code assistant, code automatic prompt, refactoring, Java EE support, various versions of tools (GIT, SVN), JUnit, CVS integration, code analysis, innovative GUI design [7].

**Advantage:** It is equipped with strong integration capabilities such as git, maven, spring, etc. As for the prompt function, it is fast and convenient, and the scope of it

is relatively broad. Besides, it allows users to simplify operations by shortcut keys and useful templates. The precise search feature is also preferred.

**Disadvantage:** There is a lack of plug-in development. Only one project is supported at the same page, which brings some inconvenience to the development, especially for the programmers who like to build a test process to test some methods during development. It requires much storage and memory compared to Eclipse.

# Chapter 3

## Requirements Specification

Requirements specification is a process of how the team gain proper and reasonable requirements from stakeholders. It is the foundation of this project [8]. Requirements should be precise and complete for team to develop further steps based on them. This chapter consists of use requirements, system specification and user story. After analyzing and collating data gained from last stage, specific requirements and system specification have been confirmed. Section 3.1 and 3.2 will include this. Then, in section 3.3, user story is produced to further enhance the team's understanding of requirements.

### 3.1 User Requirements

#### Functional Requirements

1. Users want to learn sorting algorithms through the software.
2. Users want to watch animations of the process of sorting algorithms.
3. Users want to understand and prove the correctness of sorting algorithms.
4. Users want the software to support multiple languages

#### Non-functional Requirements

The software can only animate bubble sort, selection sort, insertion sort, merge sort, quick sort, heap sort.

## 3.2 System Specifications

### Functional Specification

The list is divided into two parts: core features and optional features. Core features are essential to the operation of the software and optional features are additional functionality.

- Core Features

1. Hierarchical learning feature

A user shall be able to choose a difficulty level according to individual mastery level of algorithms.

2. Sorting animation related features

- A user shall be able to select a sorting algorithm.
- A user shall be able to view the animation, which shows the sorting process.
- A user shall be able to view explanations of each step of sorting algorithms.
- A user shall be able to start from different time frames of an animation.
- A user shall be able to customize the array of numbers to be animated
- A user shall be able to ask the software to randomly generate input.

3. Correctness related features

- A user shall be able to test an algorithm through several sets of legal inputs to see if it can work correctly.
- A user shall be able to see explanations of sorting algorithms' correctness.
- A user shall be able to choose from several sets of illegal inputs.
- A user shall be able to test whether an algorithm can work out the sorting job or not.
- A user shall be able to view explanations of why a sorting job of a sorting algorithm fails.

4. Open source

A user shall be able to access the source code of this software.

- Additional Features
  1. A user shall be able to view a user guide of software operations.
  2. A user shall be able to take a tutorial of basic algorithms.
  3. A user shall be able to view the pseudo-code of sorting algorithms.
  4. A user shall be able to view the programming code of sorting algorithms.
  5. A user shall be able to create breakpoints to see outputs at that specific point.
  6. A user shall be able to compare different sorting algorithms' time complexity.
  7. A user shall be able to view history of the latest learned sorting algorithms.
  8. A user shall be able to view individual learning progress.
  9. A user shall be able to download learning notes provided by the software.
  10. A user shall be able to practice by sorting an array of numbers manually.
  11. A user shall be able to do after-class exercises.
  12. A user shall be able to switch languages.
  13. A user shall be able to share the information of the software.
  14. A user shall be able to write feedback to developers.

**Non-functional Specifications:**

1. Usability
  - a. The software supports English and Chinese.
  - b. The software provides programming code in C, Python, and Java language.
2. Capability

The hardware requirements for basic performance is 2GB RAM and 200MB storage.

### **3.3 User Story**

Using the user story is transparent for monitoring what users want the software to be. The role, the goal, and the motivation have been specified in the text to represent each requirement.

#### **Stories about Core Features**

1. As a year one student majored in computer science, I want to learn sorting algorithms with animation so that I could see the sorting process intuitively.
2. As a year two student majored in computer science, I want to learn as many kinds of sorting algorithms as possible so that I could use different algorithms for different requirements.
3. As a year three student majored in computer science, I want to know how to prove the correctness of sorting algorithms so that I can get a high mark in my ACE module.
4. As a lecturer teaching the sorting algorithm, I want to show students the sorting process step by step so that my student can understand the algorithms easily.
5. As a student without any foundation in sorting algorithms, I want to begin with the basic concept so that I can reduce learning difficulty.
6. As an explorer of sorting algorithms, I want to enter numbers by myself and let the software help me sort them so that I can know how to deal with exceptional cases, such as lots of the same numbers.
7. As a student who has no experience in using additional software, I want to watch a novice operation guidance so that I can quickly start to use the software.

8. As a year one student majored in computer science, I want to export a learning note for each algorithm so that I can review them conveniently.

9. As a student who is struggling in the coursework, I want to know how to implement the algorithms with a specific programming language, like C, JAVA, Python, so that I can use them for reference.

### **Stories about Additional Features**

1. As a year three student majored in computer science, I want to compare time complexity between different sorting algorithms so that I can know which algorithm is more efficient.

2. As a student who is preparing for the algorithm examination, I want to do more exercise about sorting algorithms so that I can pass the exam more easily.

3. As a business school student who is interested in sorting algorithms, I want the system to display and record my learning progress so that I can schedule my fragmented time to learn.

4. As a year two student majored in computer science, I want to see the visualization of each line of code so that I can fully understand the code.

5. As a leader of a study group, I want to share this software with my teammates so that we can learn together.

6. As a student who is not good at English, I want to change the software to the Chinese version so that I can study efficiently.

# **Chapter 4**

## **Design**

To design this software, its basic usage and appearance should be confirmed first to make next stage's implementation more efficient. UMLs and low-fidelity prototype for user interface (section 4.3) are helpful in establishing this part. Particularly, team 10 has drawn use case diagram and sequence diagram, they are in section 4.1 and 4.2. Low-fidelity is shown in section 4.3.

### **4.1 Use Case Diagram**

The use case diagram displays what a user can basically do with the software. This diagram is drawn based on the requirements gathered and to confirm the requirements with stakeholders.

User can choose to learn different types of sorting algorithm.

#### **Description:**

1. View user guide of the software:

The user guide will introduce functions of each component of the software, such as buttons and control bars. Description of different modules will be described as well.

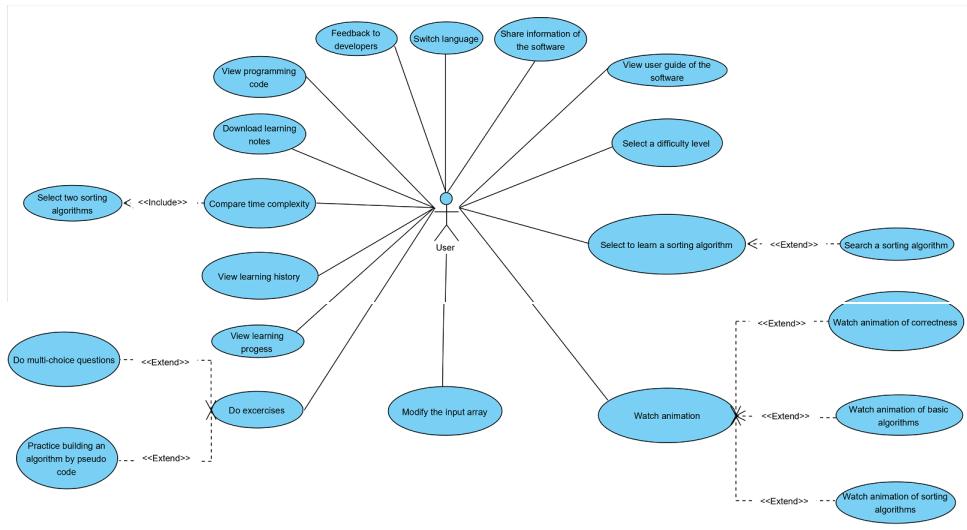


Figure 4.1: Use case Diagram

## 2. Share information about the software:

Information such as a link to the source code and contact info will be provided for sharing.

## 3. Feedback to developers:

Developers' email addresses will be provided.

## 4. Downloading the learning note:

The note is gathered and edited by developers and can be downloaded as images for learning offline.

## 5. Animation:

The software provides animation for each sorting algorithm. It will show the sorting process of its corresponding sorting algorithm. Users can adjust the animation to their requirements (Pause, step forward/backwards). Besides, users can modify the component of animation by changing the default input array. Users can choose to set the input array in two ways: select randomly or in decreasing order by the software or set it manually.

## 6. Exercise:

After learning, users can practice by doing exercises. There are two types of exercise,

multiple-choice questions and pseudo-code exercises.

For the pseudo-code exercises, users can deepen their understanding of sorting algorithms by writing their code. To do this, users need to drag the pseudo-code block provided by the software, then splicing the blocks. The software will check whether the blocks are in the correct order and tell users the result.

## 4.2 Sequence Diagram

This software provides user the most flexibility and the freedom of choice, so four small sequence diagrams are presented to show some beginning steps of each function part, instead of one diagram for the whole process.

### Get in

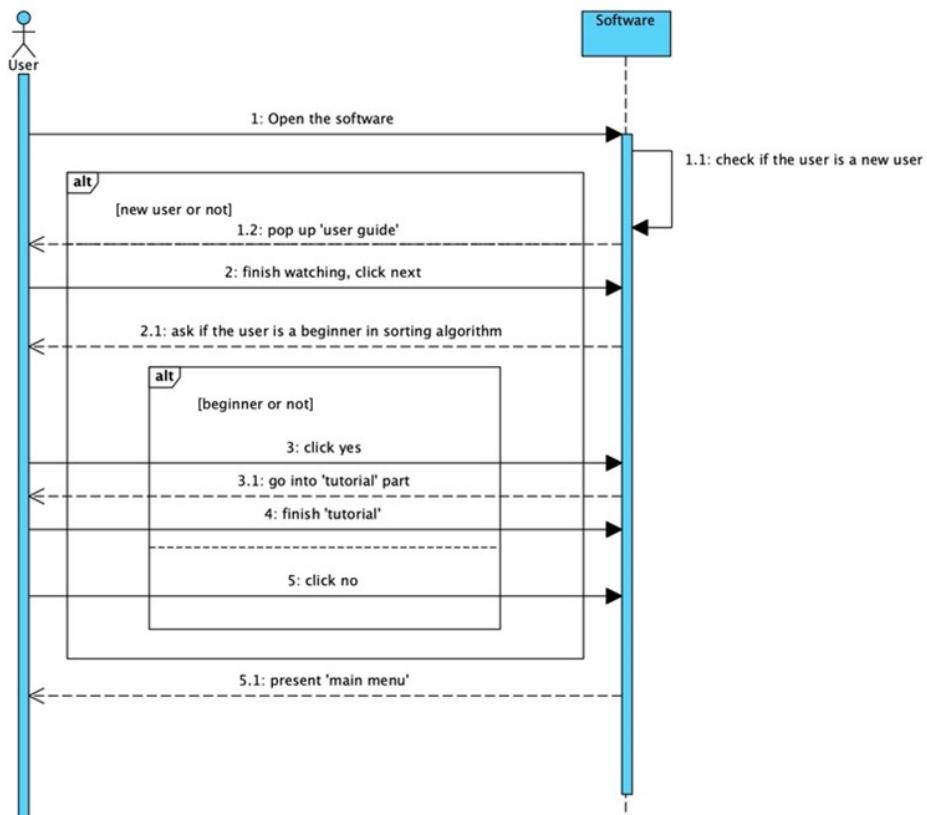


Figure 4.2: Get in

### Description:

Figure 4.2 shows the process when **user** open the software. According to **user's answer**, customized **page** will be presented.

### Animation

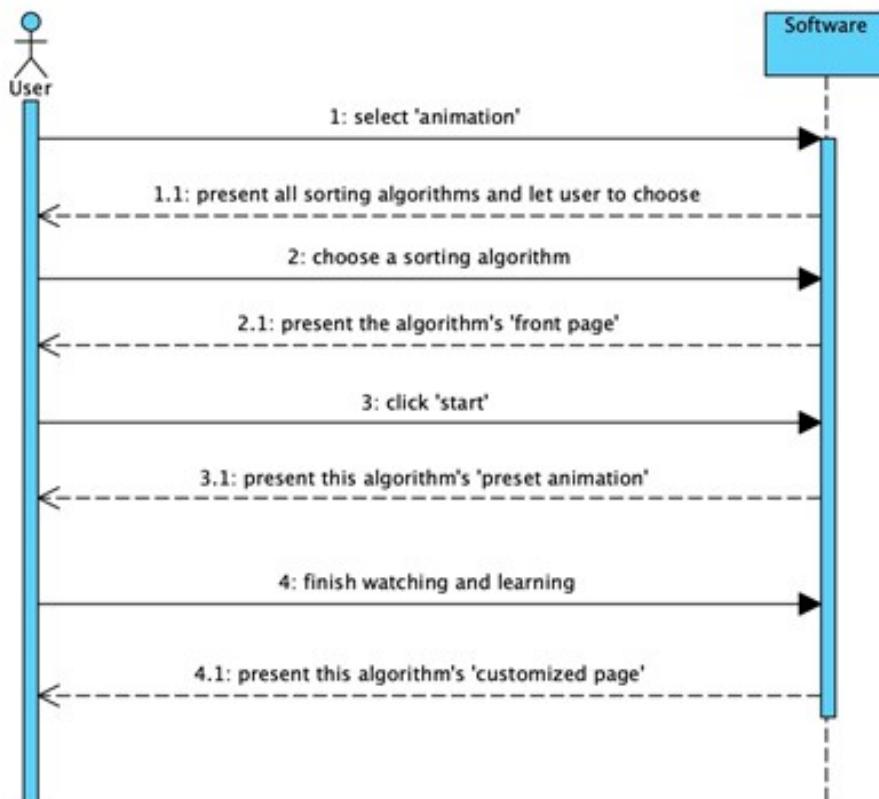


Figure 4.3: Animation

### Description:

Figure 4.3 shows the process when **user** enter “animation” page, and how they can use this function. After choosing a sorting algorithm, a simple front page will be presented with the algorithm’s brief introduction, then software will play the preset animation. After that, user can go to the “customized page”, where user can type in inputs they want.

## Efficiency

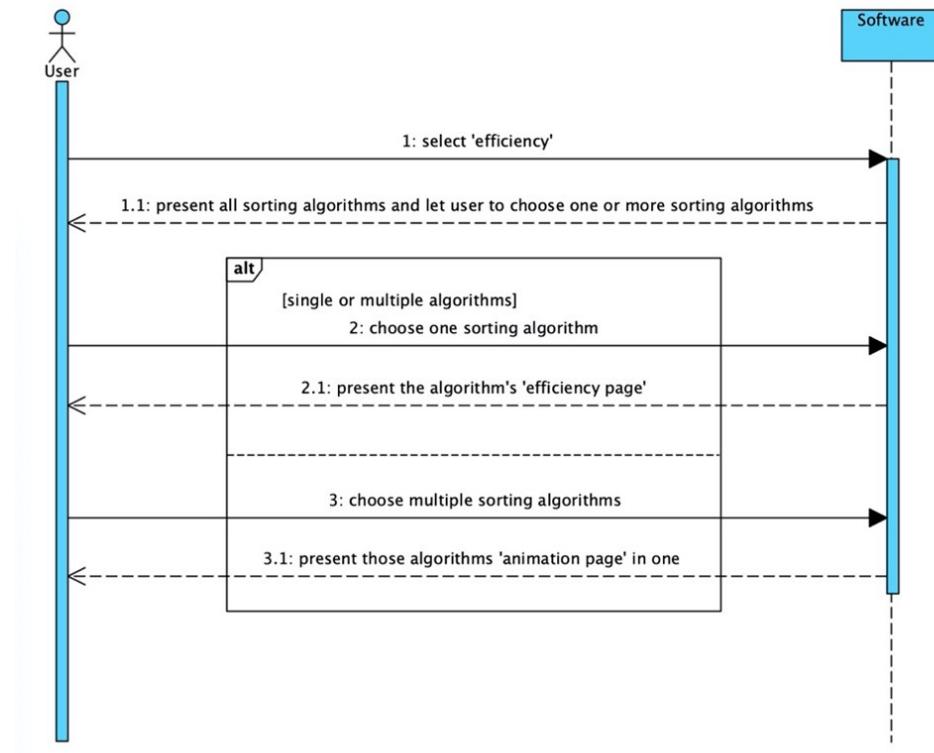


Figure 4.4: Efficiency

### Description:

Figure 4.4 shows the process when user enter “efficiency” page, and how they can use this function. User should first select one or more sorting algorithms they want to learn and compare. According to user’s selection, different interfaces will be presented. “Efficiency page” means an introduction page of this algorithm’s efficiency. “Animation page” refers to one algorithm’s sorting animation. All selected algorithms’ animation page will be presented in the same stage, and they will start at same time and play with same speed.

## Correctness

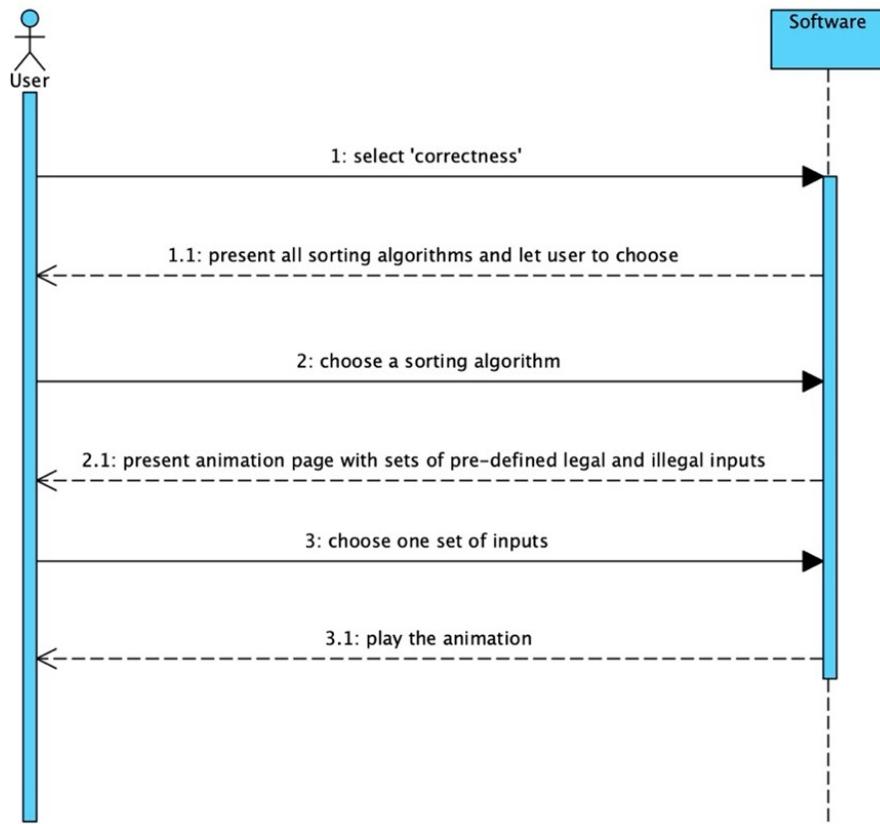


Figure 4.5: Correctness

### Description:

Figure 4.5 shows the process when user enter “correctness” page, and how they can use this function. After choosing a sorting algorithm, “animation page” will be presented. User can choose inputs set from pre-defined legal and illegal inputs. Then the software will play the animation of selected inputs.

## 4.3 User Interface and Low-Fidelity Prototype

The following pictures display a low-fidelity prototype after preliminary design. Pictures were drawn on an iPad.

## Level Choosing Page

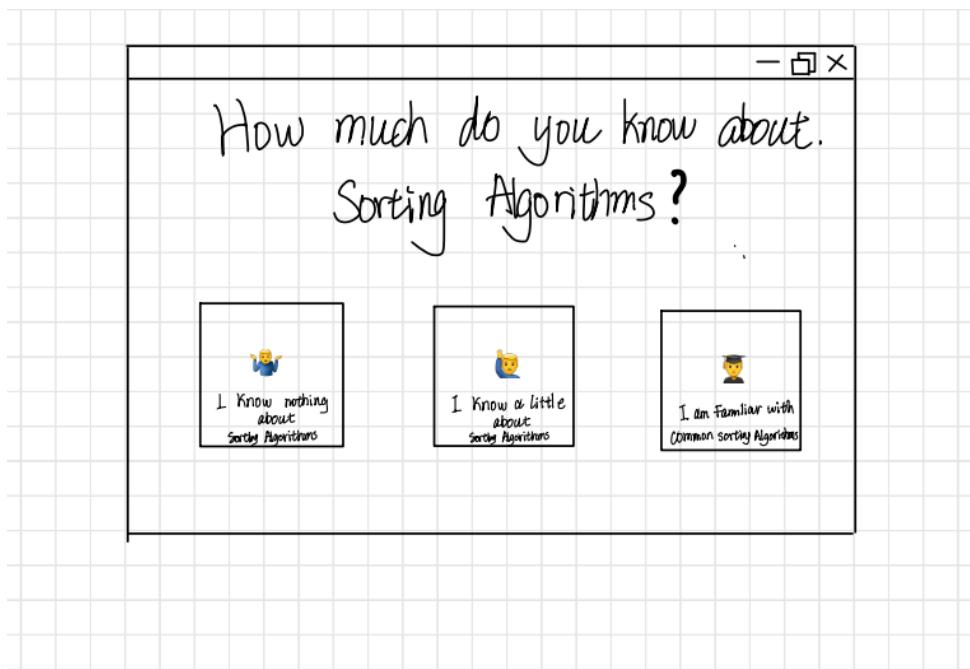


Figure 4.6: Low-Fidelity Prototype-Level Choosing Page

### Description:

Figure 4.6 displays the level choosing page. The software will jump to the corresponding tutorial according to the user's different choice.

## Main Page

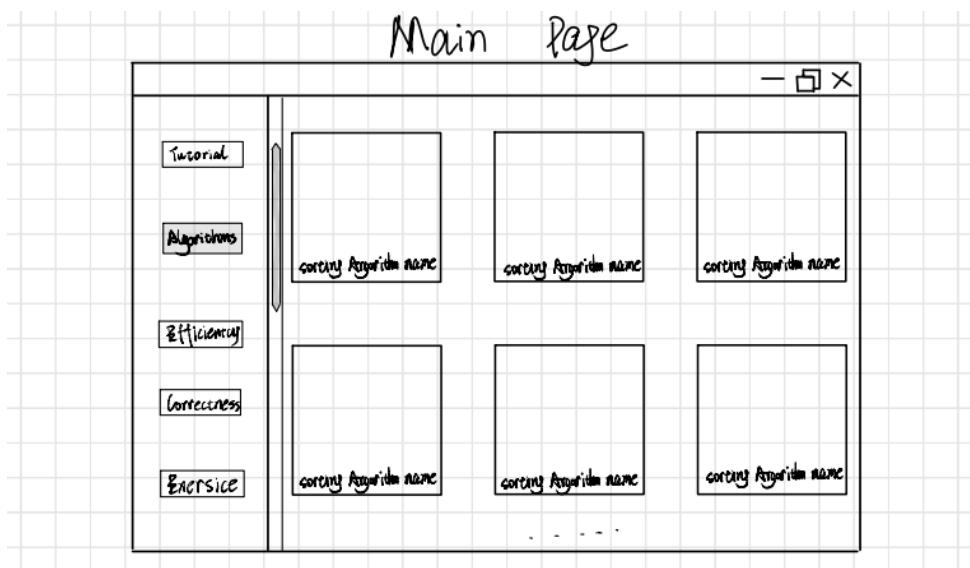


Figure 4.7: Low-Fidelity Prototype-Main Page

### Description:

Figure 4.7 is the Main Page of the software. Users can view different content by clicking buttons in the menu, which is at the left side of this page. The right side in this figure shows the 'Algorithm' module. Each rectangle presents a type of sorting algorithm. When a rectangle is clicked, the page in following will be displayed.

## Algorithm Study Page

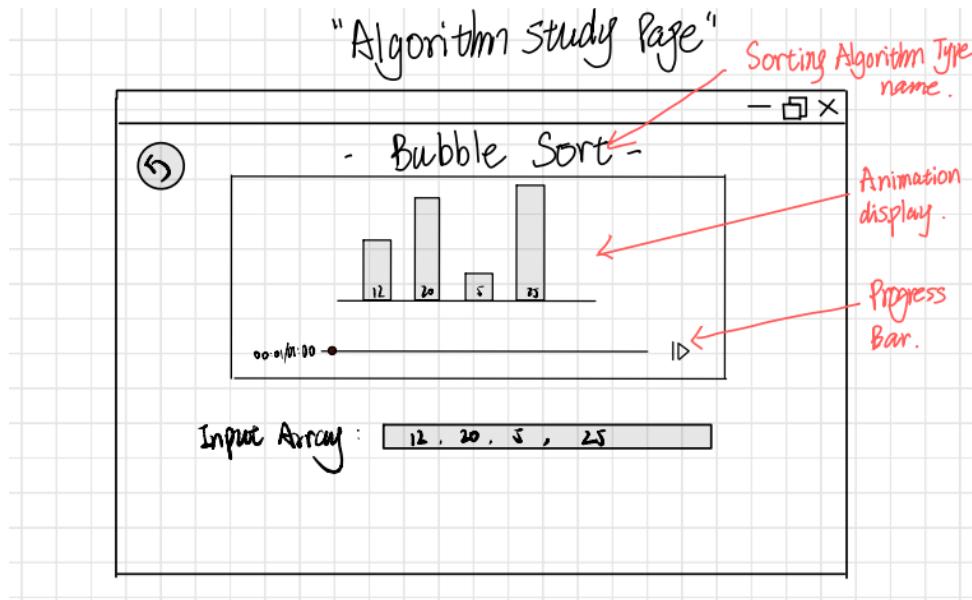


Figure 4.8: Low-Fidelity Prototype-Algorithm Study Page

### Description:

Figure 4.8 is the Algorithm study page. The name of each sorting algorithm will be displayed in the title position. This page will play the animation of the corresponding sorting algorithm.

# Chapter 5

## Implementation

This chapter claims implementations done by the team. It includes decisions the team made and high-fidelity prototypes. Team has decided the software's platform, programming language, software framework, development tools and **IDE** according to technical research and stakeholders' preference. Version control and teamwork tools are applied to help validate the workflow, file sharing and team management. Results and analysis are presented in section 5.1. Section 5.2 will show the prototypes.

### 5.1 Key Implementation Decisions

After technical research, Team 10 has gained some knowledge about the fundamental of building software with animation. Therefore, initial decisions about how to implement such software are divided into five parts. The platform is initially decided to be the Windows system. Web will be the programming base, and IntelliJ Idea will be used as the IDE. Git with GitHub is the version control tool and team management tool.

#### 5.1.1 Platform

Among 4 frequently-used platforms, PC (macOS/Windows), mobile terminals (iOS or Android), web and WeChat Mini Program, team 10 has made a choice. After analysing each platform, WeChat Mini Program and mobile terminal were excluded from the list due to the small size of their adaptive devices. The web is not considered because using

cookies may lead to serious privacy problems. Besides, according to the questionnaire results, PC is more welcomed among our stakeholders. After considering both platforms' features and stakeholders' opinions, the PC platform is chosen.

### **5.1.2 Programming Language and Software Framework**

#### **Front-end**

The project focuses on the animation, which requires a strong front-end tool. Although AWT, Swing and JavaFX can work with Java to provide a user interface, the restriction on UI design and functionalities make it less attractive. In comparison, HTML5 and CSS work well for designing UI freely. Existing learning resources are sufficient on the Internet as well. Moreover, it splits front and back ends, which further specifies the division of labour of the team and improve efficiency. Therefore, the team would choose HTML and CSS as the front-end languages for the development of UI.

#### **Back-end**

The project will contain several classes and objects for handling events and businesses which supports the service behind those events. JavaScript is supported well by most of the modern browsers but is hard to write code of business part. C# is fully functional and has plentiful resources to refer to. However, supporting Windows platform only makes this language less attractive. Java can be parsed and run on multiple platforms efficiently. It is also powerful in explaining animations and reacting to user actions. Being an object-oriented language also makes it easy to handle different events of objects. Therefore, the primary decision of the programming language of the back end would be Java.

### **5.1.3 Development Tools and Methodology**

#### **Build Project**

Gradle is used for building the software. As for the usability of Gradle, it is a software project management and comprehension tool. It can manage a project's build on any

platform, also deliver software automatically and quickly from end to end based on its rich API and mature ecosystem.

## Software Development Methodology

Since it is a small development team, and customer involvement is needed, the Agile project management approach will be used to embrace changes to requirements, delivers and frequent releases.

Specifically, **Scrum** will be used to contain a set of tasks to trace the work. During the development process, Sprints will be planned based on the backlogs which are made during the meeting. Informal meetings and daily stand-ups will be held to make sure the efficiency of the team.

### 5.1.4 Integrated Development Environment

After technical research of IntelliJ IDEA and Eclipse, IntelliJ IDEA is selected as the ultimate development management, such as convenient git project management. IntelliJ IDEA also supports automatic code generation and ZenCoding. Writing HTML would be convenient. IntelliJ IDEA also better supports for JS, CSS and plug-ins than Eclipse. Since this project is a Java Web project, the team would choose IntelliJ IDEA.

### 5.1.5 Version Control and Team Management

As for code quality, it is planned to utilise systematic methods to guarantee it during the whole process. Coding convention, including comment, naming, indentation, and changelog will be confirmed at the beginning.

Git will be used as a version control tool and Github will be the platform. In detail, the software will be developed using a test-driven development approach, and pair programming is deployed to avoid basic mistakes. As for management tools, the team is managed by several useful GitHub features such as “Project”. Issues with labels are used to raise questions, distribute tasks, alert bugs. Kanban is a clear and visible feature for managing tasks showing the whole process at the same time. Milestones will be used for making

stage-based objectives with due time, to improve productivity.

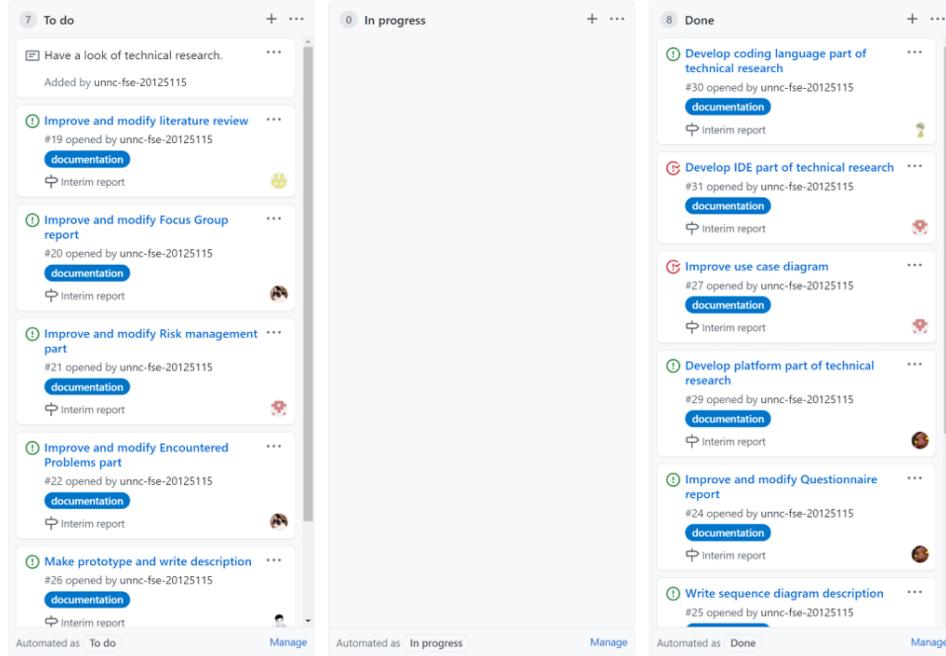


Figure 5.1: Kanban

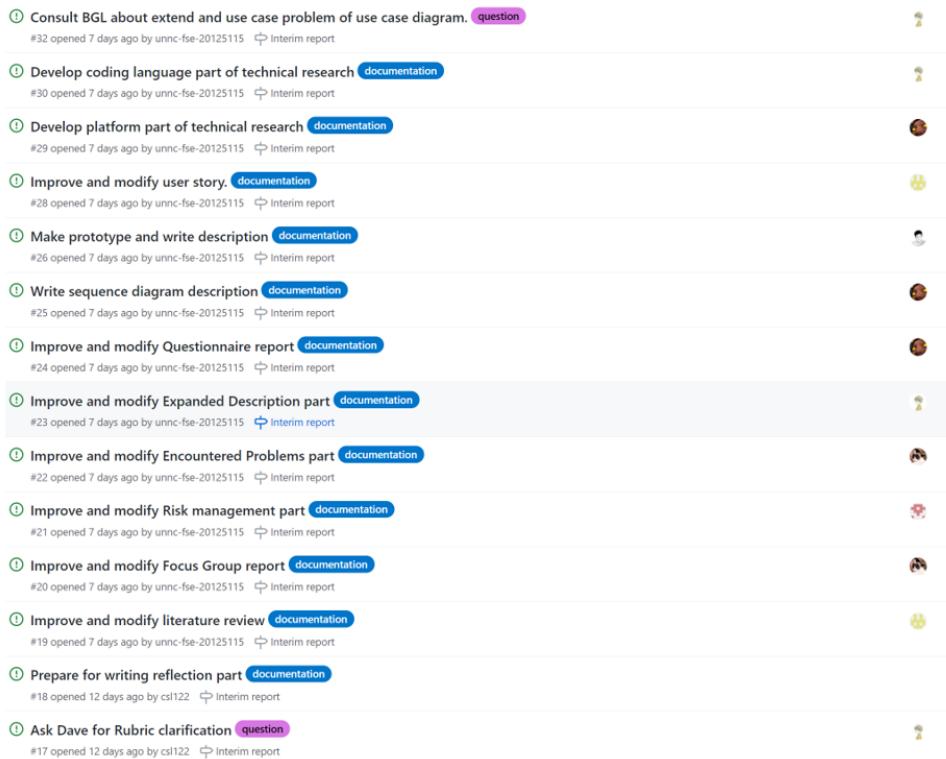


Figure 5.2: Issue sheet

## 5.2 Results of Implementation

Team 10 has been improving prototype since requirement engineering was started. It helps to determine and modify the requirements, facilitate discussions and interviews, and make the final product closer to the requirements.

### 5.2.1 High-Fidelity Prototype

In the following figures, the grey modules are expected to be images or represent the objects inside it is an entirety. The sky blue round with numbers is a mark to the object for the main text.

The learning process is divided into five parts: Tutorial, Procedure, Efficiency, Correctness and Exercise. Due to lack of space, mainly focusing on Procedure.

#### Tutorial Page

The Tutorial is designed for complete beginners. Basic algorithms and other simple topics are prepared for them.



Figure 5.3: Tutorial page

## Main Pages

Below are the main pages of Procedure, Efficiency and Correctness parts. Take Procedure part as an example: the block in Figure 5.3 is an algorithm block to learn. Every bar in Figure 5.4 is a progress bar of this algorithm, showing the percentage user have learnt in this module. Highlighted algorithm block as Figure 5.5 is recently accessed. Two buttons at Figure 5.6, the user may set language and export notes etc. in Setting, or get usage in Help. Menu Figure 5.7 is to switch parts.

Each part is distinguished by a different colour to help users identifying them. Tutorial pink, Procedure blue, Efficiency green, Correctness red and Exercise orange.

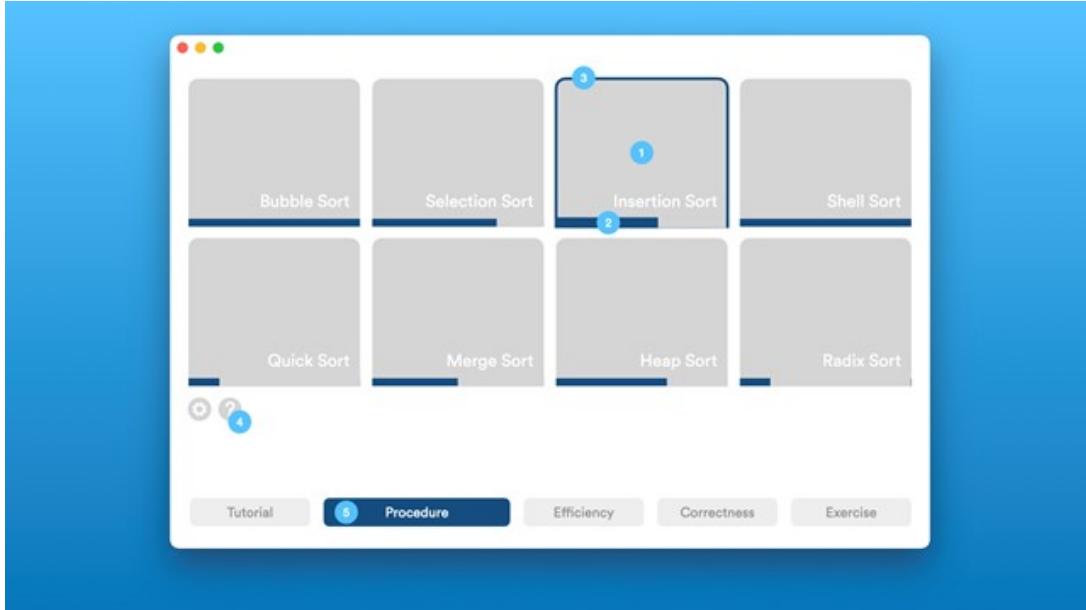


Figure 5.4: Procedure main page

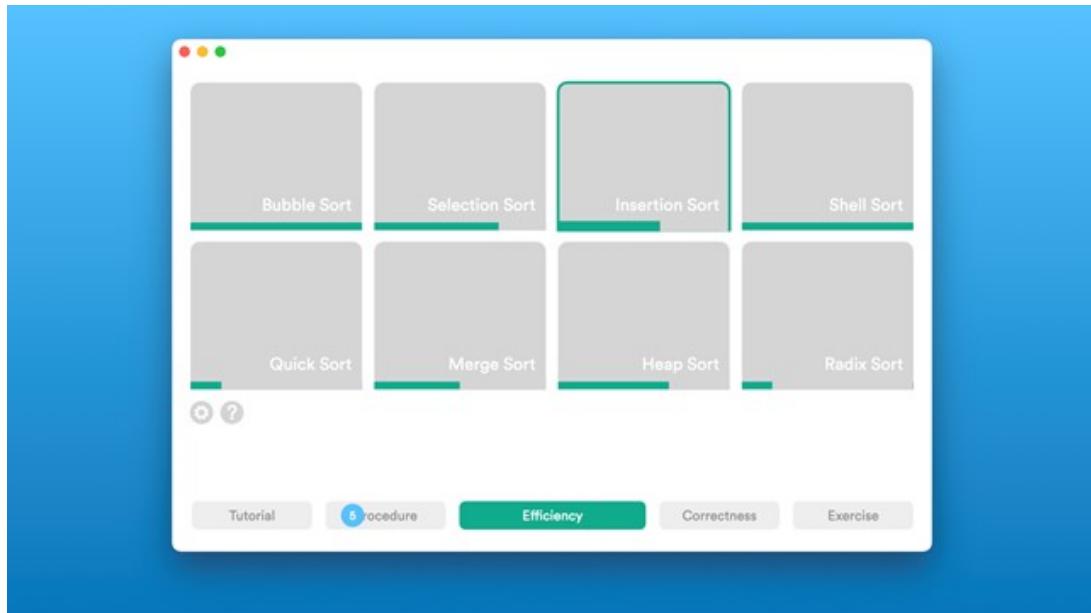


Figure 5.5: Efficiency main page

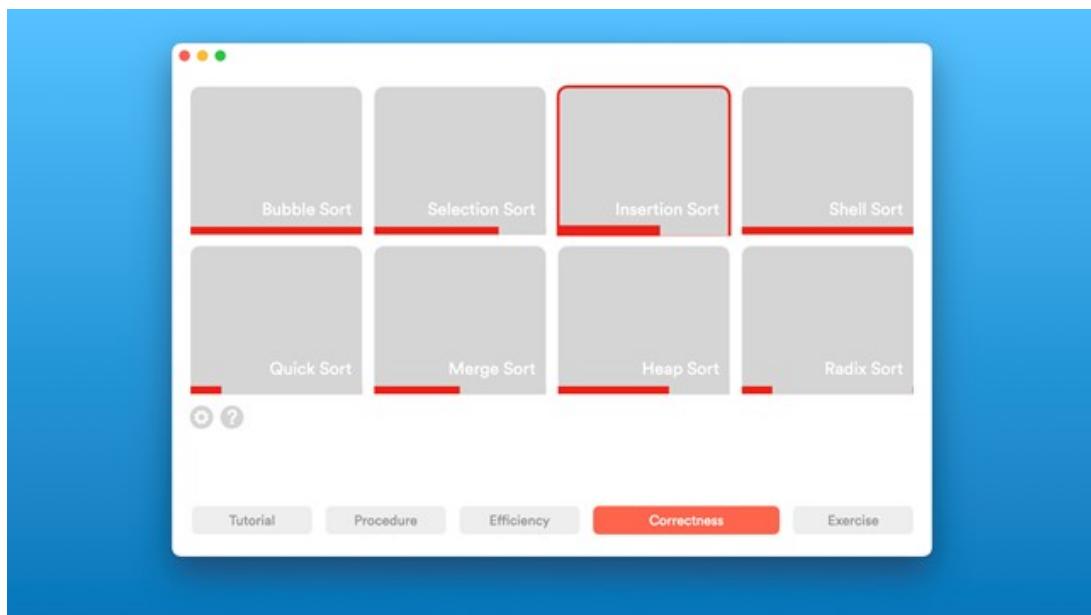


Figure 5.6: Correctness main page

## Procedure Pages

When the user clicks an algorithm block in Procedure part, Procedure part of this algorithm will be entered. The algorithm is divided into 4 subparts: Introduction, Operation, Implementation and Exercise. Take bubble sort as an example.

Below figure is the first subpart of Procedure, Introduction. Figure 5.3 is Home, back to the menu. Figure 5.4, Figure 5.5 and Figure 5.6 are used to switch between subparts. When Figure 5.5 is clicked, a drop-down box will show all subparts, allow jumping to them. Figure 5.7 is description text, which would often appear in this software. Figure 5.6 is the preset autoplay animation. Each bar is an element, and the whole bunch of elements (for short, call it visualiser in the following) is the array to be sorted. The animation will show how this algorithm works by selecting and moving the elements. The user is not allowed to operate it currently but is expected to watch it, together with the Figure 5.7 description on the left, to obtain a brief idea about this algorithm.



Figure 5.7: Procedure - Introduction

Operation subpart will be reached next. User is can play with the visualiser, that is, he/she can now edit the value of each element, add or delete elements, play or pause the process, go to next step or previous step.

Below figure is the second subpart, Operation. The visualiser is now editable. Element's value can be edited when it is clicked at the top in Figure 5.3; can also be deleted when the bottom in Figure 5.4 is clicked, a “-” will show to delete it. If the user puts the cursor at the very right of the last element, a “+” will show to add a new element. Below this visualiser, an explanation text Figure 5.5 describes what this step is doing. Bar Figure 5.6 is a progress bar of the sorting process; the dot is draggable to move forward and backwards. Buttons are able to play the animation, move forward or backwards one step.



Figure 5.8: Procedure - Operation

Implementation subpart is reached if the Next button clicked. Implement the algorithm using our building blocks, a more user-friendly method, to some extent.

Below figure is the third subpart, Implementation. In building blocks box in Figure 5.3, there are building blocks which can be dragged to the Figure 5.4 coding module. The user would operate like this to implement algorithm Figure 5.5 is a running state. Buttons in Figure 5.6 are used to see changes. The software will check if the user's algorithm is correct.



Figure 5.9: Procedure - Implementation

A congratulation message box will pop up when users successfully finished a certain part.

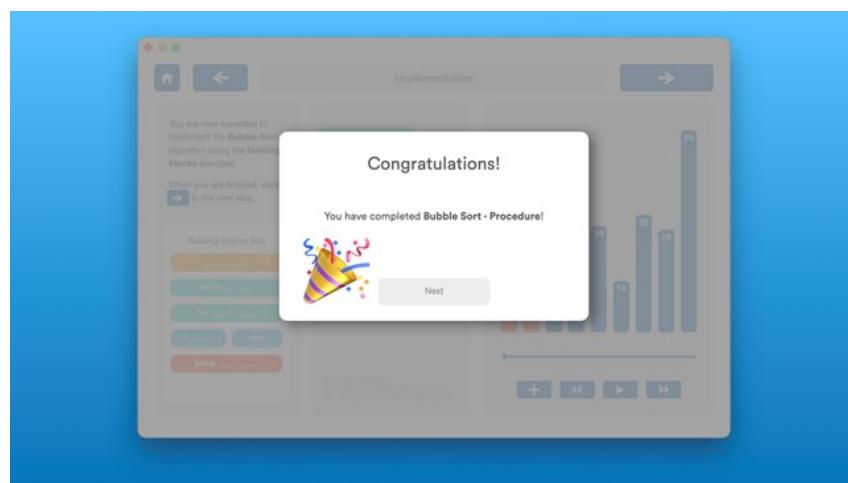


Figure 5.10: Congratulations

## Efficiency

For Efficiency part, the user will firstly be given an introduction to efficiency, and then choose multiple different algorithms to run together as array length go to infinity. Compare them to recognise basic knowledge about it.



Figure 5.11: Efficiency - Introduction



Figure 5.12: Efficiency - Comparison in small array size

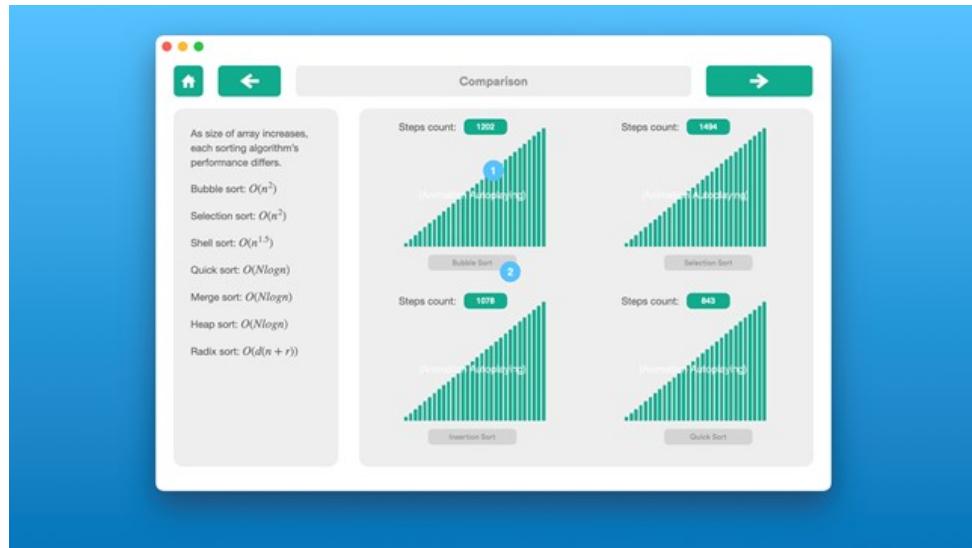


Figure 5.13: Efficiency - Comparison in high array size

## Correctness

For Correctness part, first, introduce related terms of correctness, then provide a simplified correctness definition. The user would then select an algorithm to learn. He/she would follow our learning procedure to examine this algorithm in a correctness view after then.

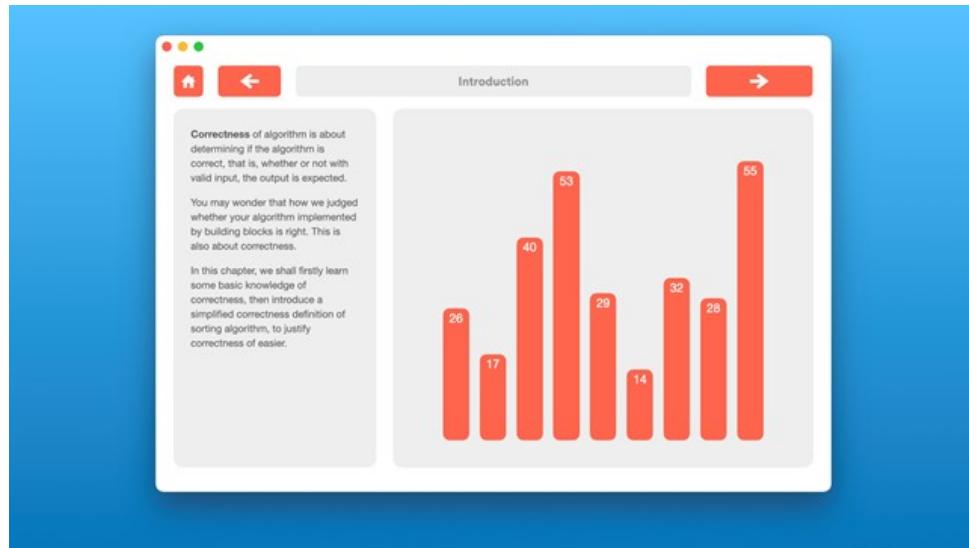


Figure 5.14: Correctness - Introduction

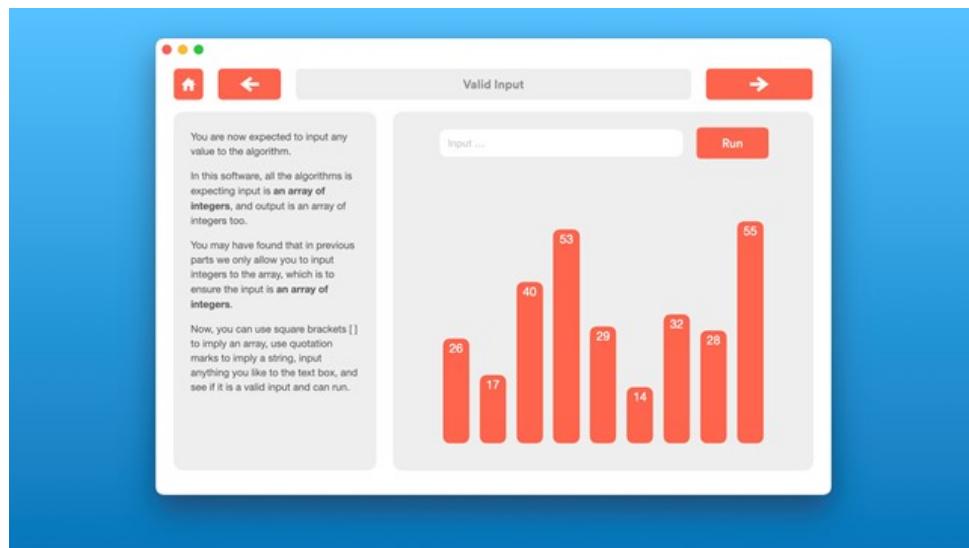


Figure 5.15: Correctness - Valid Input (other subparts are similar to this figure)

## Exercise

For Exercise part, it allows user to choose the aspects of questions. It will then generate a set of questions, let the user do in a limited time, and give feedback.

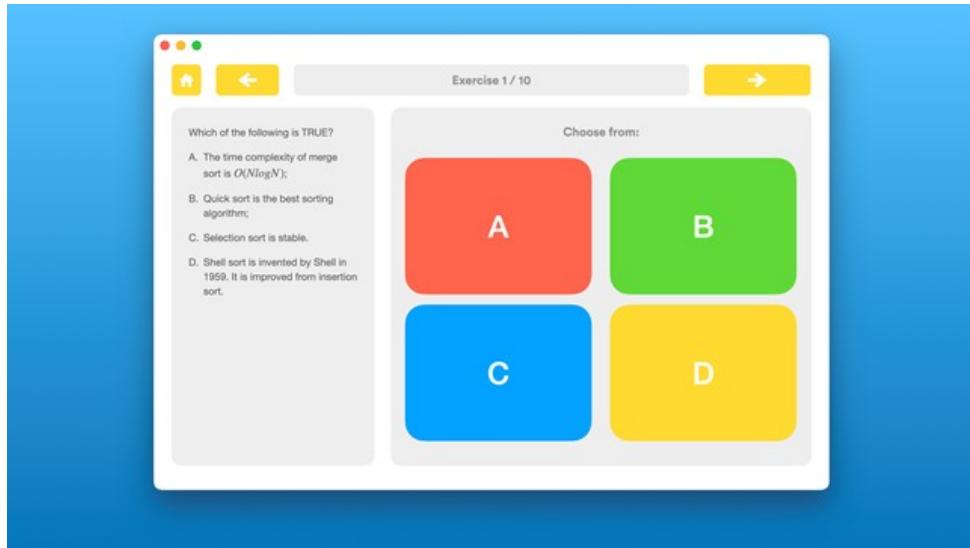


Figure 5.16: Exercise - Choice Questions

# **Chapter 6**

## **Problems Encountered and Risk Management**

As a team, conflicts cannot be avoided during teamwork. Section 6.1 includes problems that team 10 has already encountered. Team 10 also did risk management before the project started to effectively solve these problems, it is claimed in section 6.2.

### **6.1 Problem Encountered**

This section demonstrates some problems that team 10 has encountered. The causes of problems and solutions are also included.

#### **6.1.1 Technique Issue**

##### **Requirements collection**

In the beginning, there was no stakeholder except the supervisor in this project. Moreover, there were only a few basic requirements provided. Therefore, how to collect requirements was the first problem. After discussion, the teammates firstly decided the target users of the software and then agreed with the supervisor. Further, survey and focus group approaches were used to obtain requirements from target users. The results of both methods were quite useful and helped to solve this problem.

## **Software engineering method choice**

For teammates who are inexperienced in software engineering, it is not easy to decide which software engineering method to use, traditional or agile. Firstly, we reviewed the related lectures of the Software Engineering module and listed the advantages and disadvantages of both methods. Also, we referred to the advice of the supervisor. Finally, we decided to combine two methods to develop this software. The traditional development method was used to detailed record the requirement documents, while the agile development method would be used in the design and development stages.

## **Technical research**

For the technical research of development platform and development language, we were confused about research directions. That is perhaps because in this team nobody has relevant experience. Therefore, we referred to the suggestion of the supervisor. Further, we analyzed the report rubric and determined the general directions.

## **Grammar problems**

The supervisor pointed out that there were several grammar errors in documentation files. Because all members of team 10 are Chinese, there indeed exists difficulties in writing formal reports in English. To deal with this problem, we decided to double-check all documentation and pay more attention to vocabulary usage and grammar errors when writing reports.

### **6.1.2 Management Issue**

#### **Document management**

For this one-year project, there are quite lots of files need to be stored. Further, it is necessary that all members need to be able to check files at any time. In the beginning, we sent files to the WeChat group. However, because of the massive amounts of messages, members always missed the essential files. Therefore, we decided to use GitHub repository

to store all files.

## Team management

### A. Disagreement about the prototype

A severe disagreement occurred during the prototype design stage. Different members had different opinions about the process of learning sorting algorithms. This kind of situation is not a surprise. We have predicted it in the risk management section. To deal with this problem, all members must keep calm. Then, we decide to finish the sequence diagram together, and the disagreement above solved finally.

### B. The chairperson and secretary assignment

Since the chairperson of each meeting needs to prepare in advance, while the secretary is responsible for meeting minutes and tasks assignment. It is not reasonable for one person to do the work in every meeting. Therefore, we decided to take turns to be the chairperson and secretary of each meeting. Make sure that everyone had a relatively equal workload.

### C. Task assignment

Different members are responsible for different work in this project. However, it is difficult for members with cooperation to keep track of progress of each other. To solve that problem, we decided to use the Kanban function in GitHub. At the same time, we assign issues in GitHub to ensure that each member know about tasks.

## Time management

### A. Formal meeting time longer than 30min

The supervisor emphasized that each formal meeting must be controlled within 30 minutes. However, sometimes we may have lots of content expected to discuss with the supervisor. That sometimes led to the meeting longer than 30 minutes. To solve this problem, one member pays attention to the time in each formal meeting. The member will remind everyone if time is exceeded. Further, we carefully choose more important

content to discuss in formal meetings.

### **B. The overall schedule**

Compared to other groups, our overall progress seems a bit slow. The reason may be that we need to do more work to collect user requirements. Some members have raised concerns about whether we could finish the project on time. Then, we showed module convenor our time plan and asked him if it was a reasonable arrangement. The result is that he reminded us to focus on ourselves because it is meaningless to compare with different groups. We should follow our time plan.

## **6.2 Risk Management**

This section shows the possible problems team 10 may encounter in this project and the expected solutions.

Risk	Description	Strategies
Team member turnover.	A Team member leaves before project ends.	Distribute the work evenly and encourage the team members.
Disagreement among members.	There is disagreement among the group.	Analyse the reason and find common agreement among the issue. Discuss about the disagree part.
Requirement Changes.	Requirement changes due to issues.	Modify the requirements specification and verify it with the supervisor.
Process is behind the plan.	The task is too large to complete in the scheduled time.	Replan and enhance time management skills.
No experience in developing the required tools.	A Team member does not have experience with tools or language that the project requires.	Arrange learning conference to share knowledge among team members.
Submit documents over deadline.	Missing documents or submitting documents lately due to personal mistakes	Make plan of an early deadline to prevent late submission.

Figure 6.1: Risk Management

# **Chapter 7**

## **Time Plan**

The timeline of the project is illustrated in a Gantt chart. The Gantt chart is made with Microsoft Excel. By typing in the start period and duration, the chart will show the bar of corresponding length automatically. The bars coloured with different colours denote different status. Team 10 utilises this tool to trace the current stage of the plan.

The plan is basically divided into six stages, research, design, implementation, reports, testing and evaluation. Except for reports stage, each of the stages is depended on the previous one. These stages are also presented in the Git project as milestones.

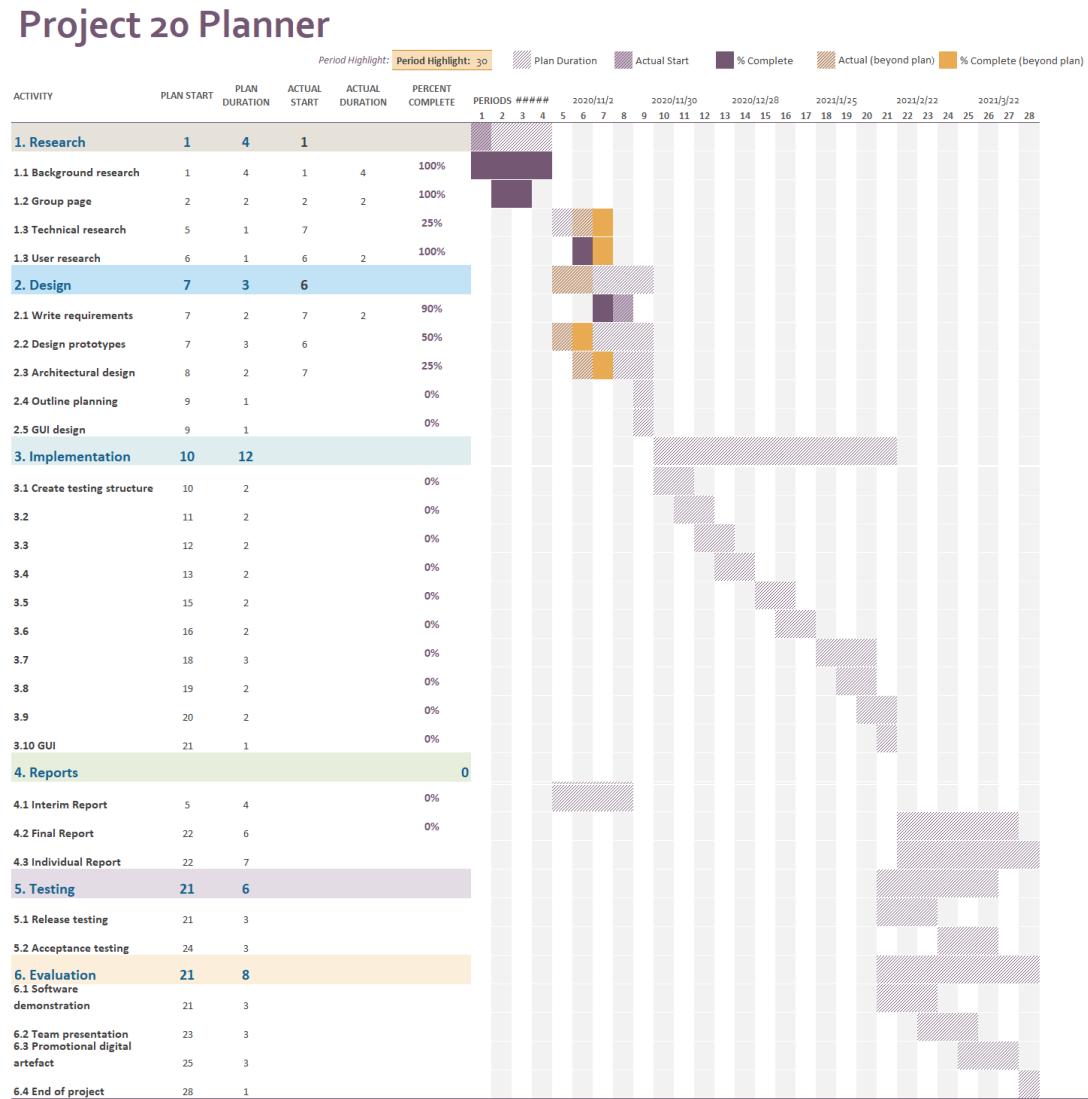


Figure 7.1: Time Plan

# Bibliography

- [1] AHMAD, T. *Performance Exploration and Testing of Web-based Software Systems*. 2020.
- [2] APPLE. Algorithm explained and animated. <https://apps.apple.com/us/app/algorithms-explained-animated/id1047532631>, 2020. Accessed: 2020-11-22.
- [3] BAECKER, R., AND PRICE, B. *The early history of software visualization* *Software Visualization*. MIT Press, Cambridge, Massachusetts, USA, 1998.
- [4] ECLIPSE. Eclipse foundation. <https://www.eclipse.org/eclipseide>, 2020. Accessed: 2020-11-27.
- [5] FLEISCHER, R. KUCERA, L. Algorithm animation for teaching. software visualization. *Software Visualization* 2269 (2002), 113–128.
- [6] HUNT, L. Html5 reference. <https://dev.w3.org/html5/html-author/>, 2010. Accessed: 2020-11-29.
- [7] JETBRAINS. Jetbrains. <https://www.jetbrains.com/idea/features/features>, 2020. Accessed: 2020-11-23.
- [8] LAWRENCE, B. *Top Risks of Requirements Engineering*.
- [9] MENDO, J. C. S. B. T. W. R. A. *Online Privacy Concerns Associated with Cookies, Flash Cookies, and Web Beacons*.

- [10] MICROSOFT. Introduction to the c language and .net. <https://docs.microsoft.com/en-us/dotnet/csharp/getting-started/>, 2020. Accessed: 2020-12-01.
- [11] NAPS, T. GRISSOM, S. The effective use of quicksort visualizations in the classroom. *J. Comput. Sci* 18, 1 (2002), 88–96.
- [12] ORACLE. The java® virtual machine specification: Java se 15 edition. <https://docs.oracle.com/javase/specs/jvms/se15/html/index.html>, 2020. Accessed: 2020-11-24.
- [13] ORACLE. Package `java.awt`. <https://docs.oracle.com/en/java/javase/15/docs/api/java.desktop/java.awt.summary.html>, 2020. Accessed: 2020-11-15.
- [14] ORACLE. Package `java.swing`. <https://docs.oracle.com/en/java/javase/15/docs/api/java.desktop/java.swing.summary.html>, 2020. Accessed: 2020-11-30.
- [15] RUDDER, A. BERNARD, M. M. S. Teaching programming using visualization. *Proceedings of the Sixth IASTED International Conference on Web-Based Education* (2007), 487–492.
- [16] SCRATCH. Scratch. <https://www.scratch-edu.com>, 2020. Accessed: 2020-11-15.
- [17] SIMOŇÁK, S. Using algorithm visualizations in computer science education. *centr.eur.j.comp.sci* 4 (2014), 183–190.
- [18] SIMOŇÁK, S. Algorithm visualizations as a way of increasing the quality in computer science education. *IEEE 14th International Symposium on Applied Machine Intelligence and Informatics (SAMI)* (2016), 153–157.
- [19] SPRING. Spring framework 5.3.1. <https://spring.io/projects/spring-framework>, 2020. Accessed: 2020-12-06.
- [20] STOFFOVÁ, L. V. V. Algorithm animations for teaching and learning the main ideas of basic sortings.

- [21] SU, W. Y. F. On application of algorithm visualization in teaching software of algorithm. *2010 International Conference on Electrical and Control Engineering* (2010), 2876–2879.
- [22] VISUALGO. Visualgo. <https://visualgo.net/en>, 2020. Accessed: 2020-11-20.
- [23] VÉGH, L., S. V. Algorithm animations for teaching and learning the main ideas of basic sortings. *Informatics in Education* 16, 1 (2017), 121–140.
- [24] WECHAT. Wechat mini-program. <https://mp.weixin.qq.com>., 2020. Accessed: 2020-11-29.

## **Appendix A**

### **Full Focus Group Report**

## **Focus Group for Requirement Gathering of Sorting Algorithms Animation Software**

### **Position:**

Project Room 07, Library (Year 1 and Year 2);  
Project Room 14, Library (Year 3)

### **Time:**

November 7, 2020 (Year 1 and Year 2);  
November 12, 2020 (Year 3)

### **Introduction**

To help prepare the questionnaire and acquire basic requirements in users' aspects, we did a focus group research.

Computer science students from three grades, year 1 to year 3, were invited to our focus group meetings. We held three focus groups respectively. Each focus group had six students. The purpose of the focus groups was to receive feedback for our questionnaire and get inspiration for some new ideas about our software.

Brief participants information and an overview of those focus groups are presented in the following. Meeting process and discussion results will be included.

### **Participants Background**

As mentioned above, there are three focus groups divided by participants' grades (from year 1 to year 3). It is ensured that participants in the same group were unfamiliar with each other. Also, they did not know our team members which guarantees the objectivity of the outcome of the discussion. Year 3 group participants are equipped with basic correctness knowledge. All participants knew the purpose of the focus group and volunteered to attend.

All participants had signed relevant information documents at the beginning of the whole process.

### **Process**

In the beginning, we introduced participants our software's purpose and basic functions at a theoretical level. Then, for participants' further understanding of our product, we presented some similar existing software and prototypes and showed functions which we may also include. After giving them a general idea of our software, here comes to the free discussion part, the discussion based on some questions we prepared in advance.

## **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 and creative ideas were raised in this focus group.

### **User guide**

We planned to have a user guide to give users a quick look at how to use this software, explaining each the usage of buttons. When talked about this feature, participants quickly responded that the user guide is necessary. One participant added that user guide should not contain too much text, or users may not want to read them.

### **Fun and game likely VS functionality**

When asked which type of software they would like to use, more interesting or more functionality, participants showed a strong preference for functional software. They would like the software to have a single purpose, without those distracting things.

### **Method of selecting modules**

We introduced two methods of selecting modules to participants. The first method was that users cannot access the next sorting algorithm's learning module unless they finished the one before. The order of modules is designed by us, from easy to difficult. The other method was free choosing. It seems that participants would like to freely choose an algorithm module instead of "unlocking" the next module by finishing the current one.

### **Displaying code**

We asked participants' attitude about pseudo-code and source code. Their responses were helpful. They thought pseudo-code is enough since it gives a logical idea of sorting algorithms. Besides, pseudo-code is more friendly to beginners, which is our target user group. Source code can be presented, they claimed, and for language, C, Java and Python are preferred.

### **Multiple language support**

When asked if the multi-language function would be helpful, the participants responded positively. They indicated that there might be some Y1 students who are not that good at English, a teaching software which only uses English may decrease its educational effectiveness.

### **Quick export of brief notes**

Students responded positively to the idea of exporting a brief note. They thought this function is useful, but not that essential.

### **Platform preference**

The participants were asked which platform they prefer. Following options were provided: PC, mobile app, WeChat mini program and website. The result is that the PC was more popular than others. Participants also said that PC would be a more suitable platform for notes' quick export function. Participants showed great interest in a mobile app at first for its portability, but they overturned this idea in a short time since they realized that a mobile screen is too small to see the animation model.

## **Additional ideas**

### **Showing algorithms' complexity**

One participant suggested that we could add a function showing each sorting algorithm's efficiency, i.e. its time complexity.

### **Breakpoints**

Participants mentioned that using breakpoints in pseudo-code or example code might be helpful for understanding code. With breakpoints, user can skip those code they already understand and quicken the learning process.

### **Using sound**

One of the participants showed us a video – many blocks with specific sound were sorted after sorting algorithm and made a fluency sound. He also recommended us to use sound to help make the learning process easier for users. He reckoned this way of presenting is quite perceptual.

## **Modules division**

One request was raised by our participants that is adding an exercise module. For the exercise module, it could contain some exercises like some multiple-choice. They also described software in their mind: there were two modules, one for learning and one for practice. The former part is just sorting algorithm's animation, like what we designed. The latter part is consolidation and practice.

## **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 program would be helpful. Besides, they suggested us to design history and reset function.

## **Year 3 group**

Year 3 students' discussion focuses more on correctness and provides other suggestions as well.

## **Difficult parts of learning sorting algorithms**

Our participants were confused about the recursive steps of some sorting algorithms. Loops were also difficult to understand. One of the solutions is that demonstrate each step. Participants all agreed that animation would be an intuitive way of understanding the sorting process.

## **Problems in learning correctness**

Correctness is hard to understand at the beginning. Participants explained that they did not expect that correctness would have a connection to mathematics and predicate logic. They hoped us to show animations of correctness logically and provide sufficient examples. However, it is difficult for year one students to understand predicate logic.

## **Way to understand correctness easier**

Participants suggested that providing stages letting users get used to the concept step by step would help. For example, termination and correct output for any legal input are two fundamental ideas of correctness. These two steps can be animated separately to illustrate fundamental ideas. One of the participants raised an idea that was using truth table to show the mapping relationship between outputs and legal inputs. Moreover, people can understand a concept easier by counterexample of incorrect algorithms.

## **How to make it interesting**

The last topic was a module that allows users to build algorithms by dragging pseudo-code blocks provided, and the software will generate corresponding code later. One of the participants suggested that using cute and unexpected

things to attract users. He suggested that an animated yellow duck character can be displayed at a corner of the software to denote whether the pseudo-code users build is correct. If it is incorrect, the duck will die. This duck can also be used as a listener to record users' understanding of algorithms when users explain their code to it.

## **Appendix B**

### **Full Questionnaire Report**

# **Questionnaire report**

## **Motivation**

The software we plan to design is oriented to students who want to learn sorting algorithms. However, details of learning method that students prefer remain unknown. Questionnaire is an efficient tool to collect both ideas and suggestions from target users. To gather requirements for the project, team 10 released a questionnaire with eleven multiple choice questions and one open question. The questionnaire remained open for seven days.

## **Objective**

1. Understand how well target users know about sorting algorithms and their learning habits.
2. Gather requirements from target users.
3. Ask for suggestions for the functions we plan to design.

## **Time**

From November 10<sup>th</sup> to November 17<sup>th</sup>.

## **Respondent**

Students from the University of Nottingham, Ningbo, China.

## **Questionnaire**

Please refer to the appendix C.

## **Result**

This survey is for collecting user requirements of animated learning software for sorting algorithms. It was released on the Wenjuanxing, an online survey platform. Overall, 207 valid questionnaires were collected. The results are as follows:

### **Question1. What's your gender?**

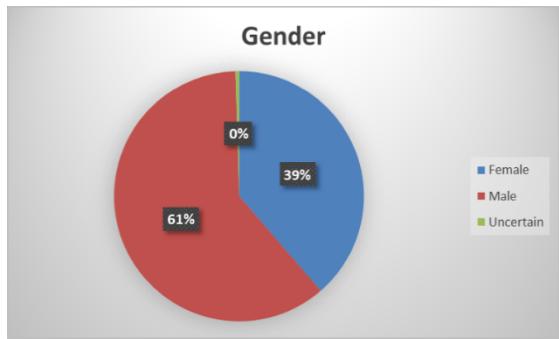


Fig1. The pie chart of results of question 1.

Overall, 207 students from the University of Nottingham, Ningbo, China were investigated, among which 61% were male, and 39% were female. The ratio of male and female is about 3:2.

### **Question2. Which year are you in university?**

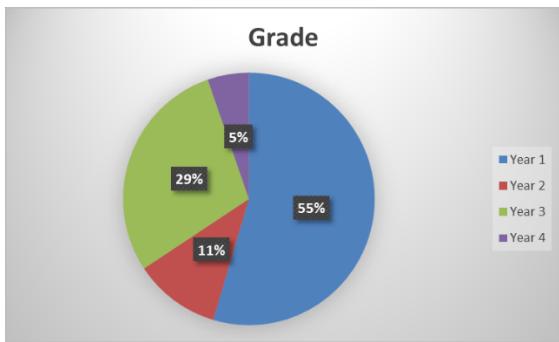


Fig2. The pie chart of results of question 2.

More than half of the participants were year 1 students, which takes 55 percent. The second is from year 3 and year 2, accounting for 29% and 11% respectively. 5% of those who filled in the questionnaire were year 4 students.

### **Question3. I learn algorithms because of:**

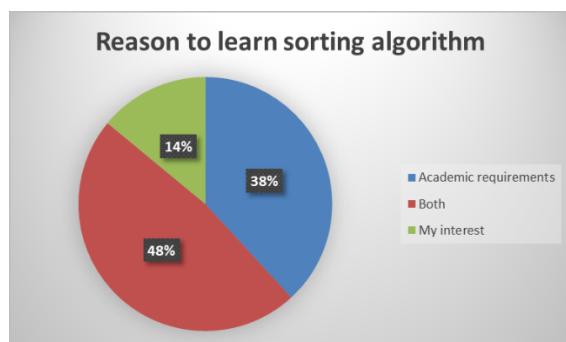


Fig3. The pie chart of results of question 3.

For the learning motivation, the proportions of 'both' and 'preparing for examination' are both high. By contrast, students who learn sorting algorithms because of interest are relatively small.

**Question4. How well do you know about sorting algorithms? (The content of following questionnaire will be adjusted according to your answer)**

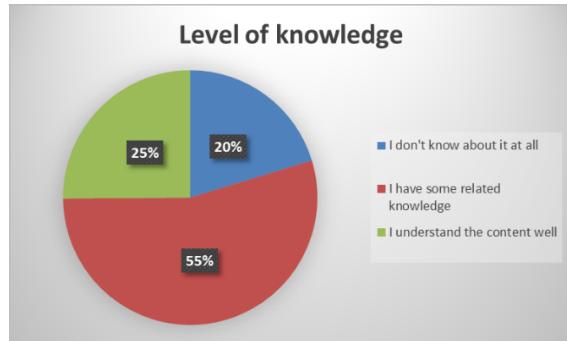


Fig4. The pie chart of results of question 4.

According to the investigation result, most of the participants have some knowledge of sorting algorithms. However, 20% of participants choose that they do not know what is sorting algorithms at all. This shows that providing a module for teaching basic concepts of algorithms for beginners might be necessary.

**Question5. In your opinion, a software which helps learn algorithms should more focus on:**

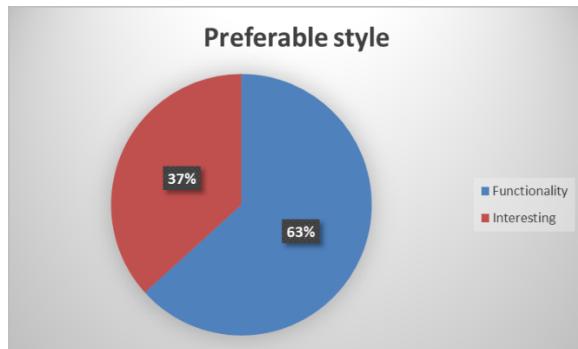


Fig5. The pie chart of results of question 5.

According to the investigation result, 63% preferred useful learning functions rather than entertaining interactive functions. This means Team 10 may reduce some of the interactive game design and pay more attention to the functions, which could provide more help for learning and understanding the algorithms.

**Question6. How much time are you willing to spend each time using our software?**

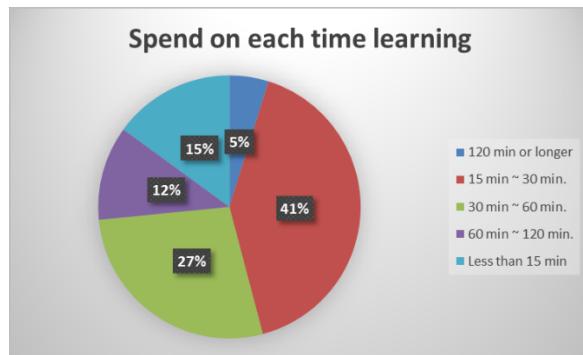


Fig6. The pie chart of results of question 6.

In general, the participants thought that within 60 minutes was acceptable for them to learn each time. More than half of the participants chose to spend 15-30 minutes each time. This shows that it is better to control one single algorithm's learning time of within about 30 minutes.

**Question7. You learn algorithms or programming mainly by:**

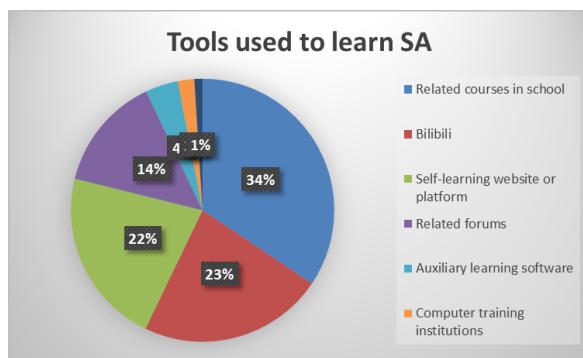


Fig7. The pie chart of results of question 7.

In addition to the university's courses, there are a large number of students learning sorting algorithms on website platforms. Video teaching software is top-rated, which was chosen by 34% of students. Other participants preferred to discuss with others in forums. It suggests that the market for our products is vast. Moreover, Team 10 may provide a function that allows users to share their thoughts with others.

**Question8. Which platform do you prefer to use this software (e.g. likes to use mobile terminal most, and computer terminal second, then 1 iOS/Android, 2 PC/Mac...)**

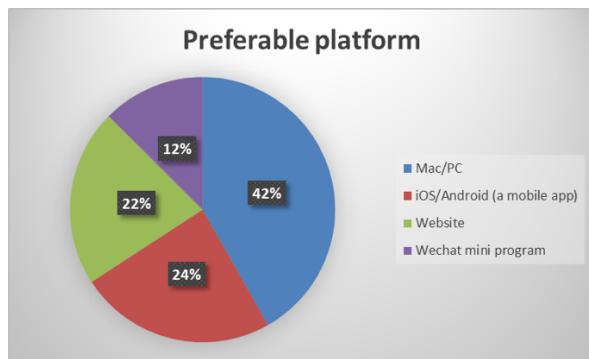


Fig8. The pie chart of results of question 8.

The investigation result demonstrates that most participants are used to use similar learning software on computers. Secondly popular ones are mobile apps and websites. This result may help decide the development platform.

#### **Question9. What goal you want to achieve through learning:**

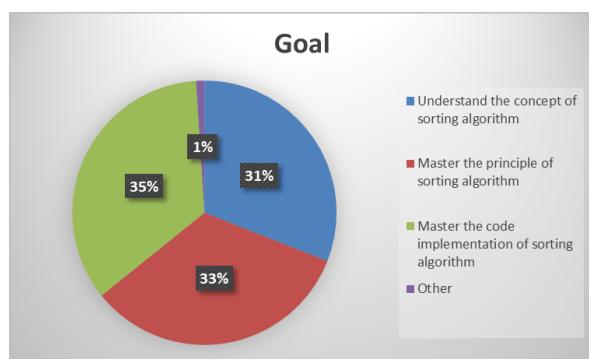


Fig9. The pie chart of results of question 9.

For these three levels of learning output provided, the choices of participants were approximately equal. That means we may need to provide a hierarchical teaching method so that users of different levels can all gain some knowledge they require.

#### **Question10. What language would you like to present the code?**

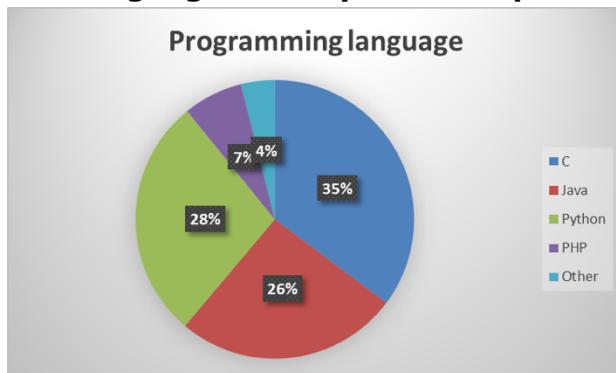


Fig10. The pie chart of results of question 10.

This question aims to collect the programming language preference of our target users. The result shows that users' demand for C, Java, and Python is relatively high.

### **Question11. Which following function do you prefer?**

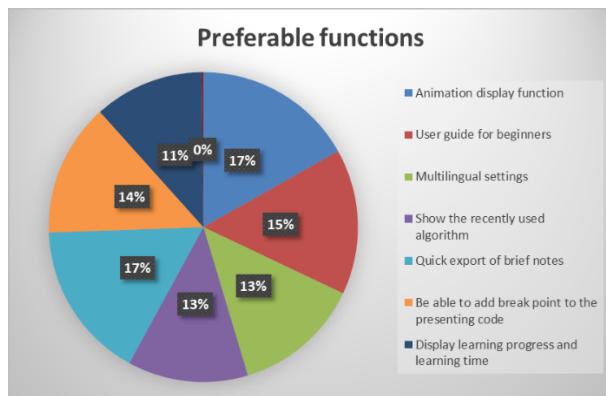


Fig11. The bar chart of results of question 11.

For the envisioned functions, what the participants like best are the animation display function and exporting brief notes function. The second is the user guide and adding breakpoints. Lots of participants select multilingual settings function and recently used function, both are approximately 13%. The learning progress and learning time displaying function are also popular. The result demonstrates that our envisioned functions can be acceptable to the users basically. Further, one participant proposed that some practice questions after learning would be helpful.

### **Question12. What other functions do you think the software should have? Are there anything we should pay attention to?**

This is an open question, and the participants can choose to answer it or not. Some valuable answers are as follows:

1. Maybe provide contact information of the developers and provide access to view the source code.
2. Share functions! If it is a good piece of software, I want to be able to share it with others.
3. The interface must be clean and delightful.
4. It should be easy to use.
5. Maybe provide a forum, and users can discuss multiple implementations on it.
6. Enable users to implement their own code.

## **Analysis and suggestion**

The participants in this survey are students from the University of Nottingham, Ningbo, China. Among them, 55% are from year 1, 11% are from year 2, 29% are from year 3, and 5% are year 4 students. The proportion of males and females is relatively equal. They learn sorting algorithms for interest or examination requirements. About 75% of the participants have varying degrees of understanding of sorting algorithms, while some know nothing about sorting algorithms.

To summarize the survey result, some important points drawn from this questionnaire are as follows:

1. It is suggested to provide a basic conceptual introduction and basic learning thought guidance for users who do not have any knowledge of algorithms.
2. Compared with the entertainment functions, participants preferred practical ones. It is suggested to reduce game mechanics and focus on developing functions that are more helpful for learning algorithms.
3. Some participants proposed that providing a function, which can share and discuss their thoughts with others, like forum, may help learning.
4. To ensure users with different levels of algorithmic bases can all gain knowledge from this software, it is suggested to provide hierarchical teaching functions.
5. For those who have examination requirements, offering more practice questions will attract them to use this software.
6. Many participants mentioned that they preferred clean and delightful interfaces. Therefore, GUI design is one thing that needs to consider.

## **Appendix C**

### **Questionnaire**

## **User demand questionnaire on the development of animation teaching software for sorting algorithm**

Our project aims to design a software which can help users who either have no experience in coding or know a little about sorting algorithms and their correctness. The basic feature of this software is clearly showing the swapping animation of sorting elements step by step. Users can also interact with the software to control the sorting process. Additionally, there are different kinds of sorting algorithms provided, such as bubble sort, quick sort, merge sort, etc. As for more advanced modules we designed, we plan to focus on the correctness of sorting algorithms, demonstrating why a specific sorting algorithm is correct. Moreover, we plan to add some additional functions, which needs your help!

All information in this questionnaire is used for research and project development, and will not be disclosed, transmitted or used for other purposes. The filling time of the questionnaire would be no more than 2 minutes. Please answer those questions according to your actual ideas. Thank you!

**(1) What is your gender?**

- Female
- Male
- Uncertain

**(2) What year are you in?**

- Year 1
- Year 2
- Year 3
- Year 4

**(3) I learn algorithms because of:**

- My interest (not examination-oriented)
- Academic requirements (examination-oriented)
- Both

**(4) How well do you know about sorting algorithms?** (The content of following questionnaire will be adjusted according to your answer)

- I don't know about it at all.
- I have some relevant knowledge.
- I understand the content well.

**(5) In your opinion, a software which helps learn algorithms should more focus on:**

- Interestingness (have some characters of game, which is entertaining, with lively and attractive interface)
- Functionality (concise interface, with learning function only)

**(6) How much time are you willing to spend in learning with our software?**

- Less than 15 min.
- 15 min ~ 30 min
- 30 min ~ 60 min
- 60 min ~ 120 min
- 120 min or longer.

**(7) You learn algorithms or programming mainly through:**

(multiple-choice)

- Related courses in school
- Self-learning website or platform (e.g. rookie course, MOOC, Tencent class)
- Bilibili
- Computer training institutions
- Related forums (e.g. CSDN, Baidu Post Bar, etc.)
- Auxiliary learning software

Other: \_\_\_\_\_

**(8) Which platform do you prefer to use this software?**

(Please sort the items, e.g. like to use mobile terminal most, and computer terminal second, then: 1 mobile, 2 computer...)

- PC/Mac (need to download and install)
- IOS/Android (a mobile app)
- WeChat miniApp
- Website

**(9) What goal you want to achieve through learning? (multiple-choice)**

- Understand the concept of sorting algorithms.
- Master the principle of sorting algorithms.
- Master the code implementation of sorting algorithms.

Other: \_\_\_\_\_

**(10) What language would you like to present the code?**

(multiple-choice)

- C
- Java
- Python
- PHP

Other: \_\_\_\_\_

**(11) Which following function do you prefer?** (multiple-choice)

- Display animation (speed control, go to previous step or next step, pause button).
- A user guide for beginners.
- Multilingual settings.
- Show the recently used or learned algorithm
- Quick export of brief notes
- None of above: \_\_\_\_\_

**(12) (optional) What other functions you think the software should have?** You are welcome to talk about your expectations of our software.

## **Appendix D**

## **Minutes**

# Team 10 - 1st Meeting

## Information

**Time:** 2020.9.30 15:00~15:45

**Location:** Online

**Chairperson:** Shiliang Chen

**Secretary:** Ruizi Han

**Translator:** Ruizi Han

**Attendance:** 6/6

<b>Members</b>	<b>Yiming Tang, Shiliang Chen, Yani Huang, Ruizi Han, Yijie Lu, Yuting Jiang</b>
Late	0
Absent	0

## Agenda

*The whole meeting is expected to take 40 minutes.*

1. Ask members to submit their CVs and give some comments. (About 10 minutes)
2. Discuss which 3 projects to choose (About 20 minutes)
  - Choose a project between 15 and 16
  - Decide which project should be put in the first place
  - Select our 3 projects
  - Decide the order of these 3 projects
  - Alternatively tell other teams our decision
3. Discuss the completion of bidding document (About 10 minutes)
  - Share eoi format file
  - Share framework of bidding document
  - Check each group's recent research result
4. Raising questions and allocate tasks of next stage (About 10 minutes)
  - Decide the chairperson and secretary of next meeting
  - Decide the specific time of next meeting
  - Confirm and go through next stage's tasks

---

# Minute

## Outcomes

1. CVs should be spell checked. Some details will be checked after the meeting (Time spent: 5 minutes)
2. Selection and order arrangement of 3 projects (Time spent: 30 minutes)
  - Project 16 is chosen from 15 and 16, because there are fewer competing groups.
  - Project 16 is in the first place.
  - Final decision: 16, 20, 12
  - We could tell other teams our decision
3. The process of bid writing and plan for the next stage (Time spent: 10 minutes)
  - Share and explain the framework of bidding document: fill in the framework first, then deal with the format
  - Share useful bidding document website
  - Share and explain the template of minute
  - All three groups did some research on their project
  - Decide to upload all useful files (all meeting minutes and template) on GitHub
  - Chairperson should share the meeting agenda with members before each meeting
4. Preparation for next meeting (Time spent: x minutes)
  - **Chairperson:** Yijie Lu
  - **Secretary:** Yuting Jiang
  - **Time:** 3 p.m. 4/10/2020
  - Acceptance and discussion of bidding document. Plan what to do in the next stage.

---

## Work summary for last stage

Task	Members	Report	Question	Completeness
Writing CV	All		/	100%
Work on the draft of bidding document	All		We changed our projects in this meeting.	Every group did some research on their project.

## Tasks

Task	Members	DDL
Spell check your CV	All	9.30
Content check of CVs	Shiliang Chen	9.30
Finish the draft of bidding document	All	10.3
Share your github account	All	9.30
Put every member in a github organization	Yiming Tang	9.30

## Problems

Priority(0 for highest, 5 for lowest)	Problem Description	Possible consequence	Proposer	Tentative Solution	Expected completion time
3	Should we elect a leader?			We will do that after we know our project.	10.7
3	How to share minutes and other important files among members conveniently?			We decided to use GitHub.	9.30

## Comments

Whole meeting process has finished successfully. Agenda are basically completed.

Main problems found here are:

1. Submission and checking of works should be done before the meeting. Common problems shall be mentioned during the meeting, otherwise after the meeting privately
2. agenda shall be released at least half an hour before start to get everybody prepared.
3. Template for bid is kinda rough. I will reorganise it after meeting
4. checking for CVs will be done after meeting

# Team 10 - 2nd Meeting

---

## Information

---

**Time:** 2020.10.04 10:00a.m.~11:00a.m.

**Location:** Online

**Chairperson:** Yijie LU

**Secretary:** Yuting JIANG

**Translator:** Yuting JIANG

**Attendance:** 6/6

<b>Members</b>	<b>Yiming Tang, Shiliang Chen, Yani Huang, Ruiz Han, Yijie Lu, Yuting Jiang</b>
Late	0
Absent	0

---

## Agenda

---

*The whole meeting is expected to take 60 minutes.*

1. discussion of form (About 20 minutes)
  - the form might contain the contents, reference, and prototype (up to the bids' content)
2. every team comes up questions when they wrote the bid, other partners give some suggestions. (About 15 minutes)
  - difficulty and confusion about the project of 16, 20, 12
3. every team illustrates their idea of the bid (About 20 minutes)
4. Allocating tasks of next stage (About 5 minutes)
  - Decide the chairperson and secretary of next meeting
  - Decide the specific time of next meeting
  - Confirm and go through next stage's tasks

# Minute

## Outcomes

1. Bidding form discussion (Time spent: 10 minutes)
2. Go through the three draft biddings (Time spent: 5 minutes)
3. Talk about the problem of three draft biddings (Time spent: 20 minutes)
4. Talk about the comprehension of three projects (Time spent: 20 minutes)
5. Preparation for next meeting (Time spent: 5 minutes)

- **Chairperson: Yani HUANG**
- **Secretary: Ruizi HAN**
- **Time: 6/10/2020**
- Confirm three biddings and submit

## Work summary for last stage

Task	Members	Report	Question	Completeness
The 1st version of biddings	All		Contents, reference, and prototype.	100%

## Tasks

Task	Members	DDL
Finish 2nd version of biddings	All	10.6
Check for each other	All	10.7

## Problems

Priority(0 for highest, 5 for lowest)	Problem Description	Possible consequence	Proposer	Tentative Solution	Expected completion time
4	How to guarantee the final quality of three biddings?		All	Discussion and check for each bidding together.	10.7

## **Comments**

---

Whole meeting process has finished successfully. Agenda are basically completed.

Main problems found here are:

1. Deadline for tasks should be more reasonable, it is better to make sure that there is enough time to submit the bids
  
2. To make more efficiency, if the leader needs to collect the information from partners, asking by order forwardly instead of waiting, but if someone wants to speak at first, the priority should be given.

# Team 10 - 3rd Meeting

---

## Information

---

**Time:** 2020.10.6 21:00 ~ 22:15

**Location:** online

**Chairperson:** Yani Huang

**Secretary:** Ruizi Han

**Translator:** Ruizi Han

**Attendance:** 6/6

<b>Members</b>	<b>Yiming Tang, Shiliang Chen, Yani Huang, Ruizi Han, Yijie Lu, Yuting Jiang</b>
Late	0
Absent	0

---

## Agenda

---

*The whole meeting is expected to take 70 minutes.*

1. Discuss and examine bid of 3 groups (About 60 minutes)
  - Members give some advice
2. Prepare for bid submission (About 5 minutes)
  - Final version of CV
  - Final version of bid documents
3. Rasing questions and allocate tasks of next stage (About 5 minutes)
  - Decide the chairperson and secretary of next meeting
  - Decide the specific time of next meeting

---

## Minute

---

# Outcomes

1. Discuss and examine bid of 3 groups (Time spent: 60 minutes)
  - o Go through each group's bidding documents and members give some advice
2. Prepare for the submission (Time spent: 10 minutes)
  - o Collect each one's pdf format CV
  - o Decide which member will submit final file
  - o Set ddl for the final version of bid
3. Preparation for next meeting (Time spent: 5 minutes)
  - o **Chairperson:** Yuting Jiang
  - o **Secretary:** Yani Huang
  - o **Time:** not decided yet

---

## Work summary for last stage

Task	Members	Report	Question	Completeness
Finish 2nd version of biddings	All		Format, content	100%
Check for each other	All		Grammer, format	100%

## Tasks

Task	Members	DDL
Modify each group's bid according to members advice	All	10.6 23:00
Send CV to group chat	All	10.6
Collect CVs and bids, submit final file	Yiming Tang	10.7

## Problems

Priority(0 for highest, 5 for lowest)	Problem Description	Possible consequence	Proposer	Tentative Solution	Expected completion time
/	/	/	/	/	/

---

## Comments

---

The meeting went well. But still exist few problem that could be improved:

- Spend too much time on reading documents, this could be done before the meeting for efficiency.
- The meeting was held late (21:00 - 22:15), the meeting may be more efficient if it's not held on-line.

# Team 10 - 4th Meeting

## Information

**Time:** 2020.10.10 10:15 - 11:10 (55min)

**Location:** Library

**Chairperson:** Yuting JIANG

**Secretary:** Yani HUANG

**Translator:** Yani HUANG

**Attendance:** 6/6

<b>Members</b>	<b>Yiming Tang, Shiliang Chen, Yani Huang, Ruizi Han, Yijie Lu, Yuting Jiang</b>
Late	/
Absent	/

## Agenda

*The whole meeting is expected to take 50 minutes.*

1. Decide roles of our team (About 15 minutes)
  - contain the Leader, UI Designer, Technical Leader, Quality Assurance Leader, Editor, Repository Master.
1. Discuss basic points of our project [P20] (About 20 minutes)
  - contain the understanding of project, the bid, and questions.
  - Confirm the points that should be discuss with the supervisor in the formal meeting. (About 10 minutes)
  - write an email maybe.
4. Allocating tasks of next stage (About 5 minutes)
  - Decide the chairperson and secretary of next meeting
  - Decide the specific time of next meeting
  - Confirm and go through next stage's tasks

---

# Minute

---

## Outcomes

### 1. Division of labour:

- Team Leader: Shiliang Chen
- UI Designer: Yiming Tang, Yani Huang
- Technical Leader: Shiliang Chen
- Quality Assurance Leader: Yijie Lu, Yuting Jiang
- Report Editor: Yani Huang, Yuting Jiang ,Ruizi Han
- Repository Master: Ruizi Han, Yiming Tang
- Monitor (Tester): Yani Huang, Shiliang Chen
- Mediator

*This is a Temporary arrangement, which will be adjusted according to the needs in the future.*

### 2. Questions in the 1st Formal meeting:

- Suggestions on our project concept and bid.
- How to realize "correction" in software function?
- Are the expenses incurred in the R & D process reimbursed?
- Requirements for requirements (the team intends to check more information, whether other methods are needed)
- Selection of SE development process.
- Idea about the ethics form

### 3. Preparation for next meeting (Time spent: 5 minutes)

- **Chairperson:** Yiming Tang
- **Secretary:** Shiliang Chen, Yijie Lu
- **Time:** Unknown
- **Remark:** Remember to Bring a recorder.(Shiliang Chen)

---

## Work summary for last stage

Task	Members	Report	Question	Completeness
Writing Bids	All Members	/	/	All Done

## Tasks for next stage

Task	Members	DDL
Prepare for the Formal Meeting.	All Members	/
To explain idea of the project to the supervisor	Yijie Lu	Before Next Meeting
Determine which SE process to use		

## Problems

Priority(0 for highest, 5 for lowest)	Problem Description	Possible consequence	Proposer	Tentative Solution	Expected completion time
/	/	/	/	/	/

## Comments

The meeting proceeded smoothly and was completed within the expected time.

One problem is that if a member is required to make a specific statement in a meeting, the member should be informed in advance of what he or she intends to present.

# Team 10 - 5th Meeting

## Information

**Time:** 2020.10.15 10:00~11:00

**Location:** PMB449

**Chairperson:** Yiming Tang

**Secretary:** Shiliang Chen, Yijie Lu

**Translator:** Shiliang Chen, Yijie Lu

**Attendance:** 7/7

<b>Members</b>	<b>Heshan Lu, Yiming Tang, Shiliang Chen, Yani Huang, Ruizi Han, Yijie Lu, Yuting Jiang</b>
Late	0
Absent	0

## Agenda

The whole meeting is expected to take 30 minutes.

### 1. Opening (About 1 minutes)

- Many thanks on awarding us this project, and we are very honoured to be supervised by you. We are Team 10, and we treat this as a formal meeting.
- This meeting is arranged to 30min long

So we divide this meeting into 2 parts,

- self introduction, then we will talk about the project,
- discussions

### 2. Self-introduction(About 5 minutes)

- Let Heshan know us.

### 3. Discussion (About 16 minutes)

### 4. Our questions (About 6 minutes)

- We have created a GitHub repository, to manage our files. Would you have any interest to join us the private team, to take a look at it?
- Do you have any suggestions on our project concept and bid?

- How to understand the "correction" in software function?
- How to gather more requirements?
- Expenses reimbursed?
- Need to fill the ethics form? We thought in the Alpha testing or Beta testing, we would like to invite some other students to test the procedure,
- Suggestions on our procedure?
- Suggestions on formal meeting?

5. Heshan's questions (About 10 minutes)

6. Rasing questions and allocate tasks of next stage (About 3 minutes)

- Raise questions
- Decide the chairperson and secretary of next meeting
- Decide the specific time of next meeting
- Review all new action points, confirm and go through next stage's tasks

7. Others (About 6 minutes)

- Other issues to be raised

## Minute

### Outcomes

1. Opening (Time spent: 2 minutes)

- Completed

2. Self-introduction (Time spent: 3 minutes)

- Completed

3. Discussion (Time spent: 50 minutes)

4. General

- English speaking in formal meetings (compulsory)
- Heshan will join our informal repo

5. The solution of proving the sorting algorithm's correctness.

- Guide user to understand the sorting algorithm
- Allow user to create an algorithm in their way.
- Additional question: confirm that the game can help prove the correctness

6. About Requirement

- Basically, first we come up with our own ideas then check existing implementation. Analyse existing ones' pros and cons. After this, build a prototype and do market research on what our user will like.
- Functional requirement should **think about what we can operate in the software as a user**, it more focuses on the function the user needs to achieve; non-functional requirement more focuses on the constraints on what the user needs to do such as the language preference, opening source for user to download

- **Functions we build must be able to improve the ability to help users.** Don't come up with new idea that's only for differentiating our product with others. Main task for requirement is to let user to understand the algorithm. Designing different shape on disorder elements for making some differences from other software might not exactly achieve the goal. Considering change the integers to letters.
- Deciding the stakeholder. Focusing children group might be a little difficult, because we do not have enough experience for contacting with kids. Additionally, if we need to interview groups in a formal way, we should prepare for the ethic form.
- Functional requirements should be more specific and precise but not vague

## 7. Other issues

- Ethic forms are on the air
- 1617 project

## 8. Software engineering method

- Depend on team members
- Heshan prefers Agile

## 9. Experience from earlier team

- **Cooperation and teamwork**
- **Innovation** is very important

## 10. Preparation for next meeting (Time spent: x minutes)

- **Chairperson:** Ruizi Han
- **Secretary:** Yiming Tang
- **Time:** 10/20 Tuesday Night (specific time will be confirmed on 10/18)

What is going to be discussed in the next meeting:

- Discuss the market analysis
- Confirm the target market
- Discuss what we will do in requirement analysis
- Discuss about SE method we are going to use
- First version of requirement analysis
- Decide the order of chairperson and secretary
- Discuss about group website
- Prepare for the formal meeting

## Last stage's action points review

Task	Members	Report	Question	Completeness
Determine which SE process to use	All	/	TBA	50%

## Action points

Task	Members	DDL
Email Heshan for 1617 project, invite her for github, ask for agenda template	Yiming Tang	10/16
Write sth about requirements (functional especially)	All	10/20
Ethic form	/	10/29
Group website design	/	10/29

## Problems

Priority (0 for highest, 5 for lowest)	Problem Description	Possible consequence	Proposer	Tentative Solution	Expected completion time
1	Pay attention to innovation	more impressive product	Heshan	mind storm	/
1	Functional requirement is not specific	serious problems	Heshan	assign task	10/20
3	SE method is not determined	/	Shiliang	discuss in the next meeting	10/20

## Comments

I didn't control the time well, because I didn't expect that a question could develop into a discussion of nearly ten minutes. I had planned to ask a few questions in about ten minutes, but I didn't expect that the 30 minute meeting would last nearly an hour.

Next time, we can omit some questions in a planned way, and then the chairperson may have to rehearse in advance. If necessary, he can interrupt the discussion and say "we are running out of time, get into the next part.

In addition, the chairperson can ask the team members in the wechat group whether we can go on to the next topic.

Every time the chairperson enter the next part, he or she must ask everyone about "any questions from others".

# Team 10 - 6th Meeting

## Information

**Time:** 2020.10.20 20:00~22:00

**Location:** Library 4F Project Room 30, 32

**Chairperson:** Ruizi Han

**Secretary:** Yiming Tang

**Translator:** Yiming Tang

**Attendance:** 6/6

<b>Attendance</b>	<b>Yiming Tang, Shiliang Chen, Yani Huang, Ruizi Han, Yijie Lu, Yuting Jiang</b>
Late	0
Absent	0

## Agenda

*The whole meeting is expected to take 30 minutes.*

1. Last weeks' tasks (Expected 3m)
  - Mail to Dr. Heshan completed?
  - Dr. Heshan's documents (Agenda template & 1617 GRP works)
2. Requirements (Expected 40m)
  - Share requirements with team members, and they may mention measure
  - Target users (where it can be used, if users will use it often, analyze competitive products, advantage and limitations, which required by Dr. Heshan to be written in the report)
  - What we should do during analyzing requirements (survey and interview for gathering info, functional & non-functional requirements, UML etc.)
  - integrate all requirements and have a general requirements analysis document
3. Choose SE method, discuss the arrangement of the later stage and preliminarily determine the process (about 15m)
  - Which SE method best fit us? (maybe combine two methods for a period of time)

- Discuss our plans for the future, and roughly determine what to do in the future
4. Ethics form (about 10m)
    - briefly introduce the filling process and several documents
    - Assign tasks
  5. Website (about 15m)
    - Huang and Tang briefly introduce their ideas
    - Discuss
  6. Definition of correctness (about 15m)
    - Give ideas about correctness
    - Possible ways to achieve proving it
  7. (optional) Characteristics of each member (about 10m)
    - Share the test result or suitable Belbin team role for them
    - What each member is able to do
  8. Questions and next meeting arrangement (about 10m)
    - Other questions
    - Next meeting time, chairperson and secretary, how to rotate
    - Next stage tasks

---

## Minute

---

### Outcomes

(summary of all discussed main points, all decisions, all action points)

1. Last week's task
  - Mail to Heshan sent
  - Heshan's documents pushed to GitHub
2. Requirements Analysis
  - Share
    1. Tang: Encourage
    2. Jiang: Game. First open simple algorithms to him, such as bubble first, and then open others to him. But in this case, suppose what he wants to learn is difficult ones, we cannot achieve this. May integral. Bubble, integral, unlock (recharge); token, upgrade, reward mechanism; within a single algorithm or all algorithms? Milepost.
    3. Huang: In the two modes, the algorithm distinguishes the difficulty, what can be exchanged for reward token, skin and interface color?
    4. Han: There may not be long-term users who will be lost after learning; therefore, I don't want to have login. Target users: self-taught programmers, people who don't understand

5. yn: May add efficiency part, but we pay more attention to correction. Let him finish the evaluation on his own or when he finishes it?

6. Chen:

- Prefabricated animation
- Novice tutorial
- Test the demo module, let the user input the number by himself, the previous step, the next step, and the automatic playback, so that the user can know what the algorithm is doing
- User drag module, (we'll discuss a few more at that time) because he probably understands that he imitates the way the algorithm drags, which is more interactive and user friendly. It can detect whether he is dragging the right one in real time, which is more interactive than the one dragging on the right, and can also prompt
- Unlock the final module: scratch pseudo code, there may be several steps wrong, let him correct
- If the user wants to see the code, show him the code in various languages we provide

7. When showing the code to him, show the language he wants? Heshan is not Party

A. We may evaluate what percentage of users want to see the code and give Heshan the things they have investigated for confirmation.

8. Help other users understand and master the significance of sorting algorithm?

9. Once again, we can't understand what correction is and how to prove it.

- determine the target audience and market analysis of our products (where can they be used? Will you use them more? Competition comparison, analysis, advantages and limitations, which Heshan requires to be written in the report)
  - Don't think about kids, because it's hard to deal with.
  - College students who are interested in computers but basic.
  - Host group: Freshman CS major, teachers
  - market analysis
  - Do you want to send out questionnaires or ethic?
  - Competitive products

- literature review

1. Animation demonstration <https://www.cs.usfca.edu/~galles/visualization/ComparisonSort.html>, no code, only animation sort
2. Code puzzle <http://snapapps.github.io/edgy/app/edgy.html> It looks like pseudo code, but it's not for sorting algorithms
3. Galant diagram demonstration <https://github.com/mfms-ncsu/galant> Although the form of bar may be monotonous, they use Galant graph, which may replace this thing
4. Sortko: using mobile devices to learn sorting algorithm <https://ieeexplore.ieee.org/document/6185079> This paper is a mobile phone software, this article reference value is relatively high, has the collection data.
5. Visa: visualization of sorting algorithms <https://ieeexplore.ieee.org/document/6240816> Visualization of sorting algorithm,
6. All of these can be used in the literature review. The professor's words can be shown to

him.

7. Algorithm animation diagram

-Discuss what we need to do in the requirement analysis phase (survey and interview to collect information, determine functional and non functional, draw UML, etc.)  
-I didn't have time to talk about it this time

8. Select SE method, discuss the arrangement of the later stage, and preliminarily determine the process (estimated to be 15 minutes)

-Which SE method is more suitable for us? (maybe try the combination for a while, as Dave said)

-Agile, while doing the process, there may be confirmation work and a process involving stakeholders.

-The general direction should be clear.

-Test documentation? After confirming the complete function, we should be very clear about how to test the process and effect.

-After the selection, discuss our future plans and roughly determine what to do in the future

-Finish the requirement by the 29th

9. Ethics form (estimated 10 minutes)

-Han and Colin are responsible for the assignment.

10. Website (estimated 15 minutes)

-I bought a template and picked a template to show you. only one index.html For convenience.

-Project brief (such as timeline), project document (download link, data, code version), what are you doing at this stage (cycle, file, introduction, continuous update), team member introduction (email, division of labor), and other modules to see what other requirements are

-How to add document to the website. The result of the current discussion is to put links. Tym will do it.

-Add project introduction, copy to be sent.

-The logo doesn't have to be. It's OK to design one at will, and it's OK to find it online.

- Discuss the definition of correction (estimated 15 minutes)

On correction, we put forward our own views

Meet the definition? You can prove it in your own way.

Tell the user that the algorithm is correct.

How to realize this requirement and possible ways

9. Next meeting preparation

- **Chairperson:** Chen
- **Secretary:** Tang Huang
- (next meeting after next) Huang + Jiang
- **Time:** Thursday

## Action points

Tasks	Member	DDL
Contact teacher for questionnaire (to distribute it)	Huang	10.29
Questionnaire design. 4-6 questions for each member	huang	10.21
Requirements validation		
Ethics forms	Han, Jiang	10.29

## Problems

Priority (Highest 0, Lowest 5)	Description	Expected Result	Member	Temporary Solution	Expected Solve Time
3	Is it better to write agenda and minutes in Word to be put into report?				10.22
0	Definition of correctness				10.29
1	Platform				11.5

## Comments

From sec: actually, I feel that the summary of the last meeting is not complete enough.)

The meeting proceeded smoothly, and most of the discussions needed to be done were completed within the scheduled time and a result was reached.

There are a few things that are not very good:

1. I don't understand some processes very well, and the preparation work is not enough, so that I don't know how to guide the discussion in the right direction. For example, today's discussion on the follow-up plan did not get a clear result because I was not familiar with the whole process and did not express the content to be discussed well. You need to do more preparation next time.
2. The topic change is very stiff, just reading agenda. I'll learn about it the next time the other team members do the chair person. Maybe when we talk about each discussion point, we can give some ideas and talk about something to help people think of more things.

# Team 10 - 7th Meeting

## Information

**Time:** 2020.10.22 10:00~10:30

**Location:** PMB449

**Chairperson:** Shiliang Chen

**Secretary:** Yiming Tang & Yani Huang

**Translator:** Yiming Tang & Yani Huang

**Attendance:** 7/7

<b>Attend</b>	<b>Heshan Du, Yiming Tang, Shiliang Chen, Yani Huang, Ruiz Han, Yijie Lu, Yuting Jiang</b>
Late	0
Absent	0

## Agenda

*expected 30m*

1. opening (expected 1m)

This is the agenda of today's meeting. (**Show Dr. Heshan the Agenda**)

Today's meeting is gonna be devided into 3 parts, which may take about 30 minutes

- quick review of tasks done last week
- discussion on several topics
- Q&A time

2. quick review (expected 2m)

- Yani has been working on the group site with Yiming and they are also going to design a group logo for us
- a preview version of it has been developed and detailed content will be added in few days
- Yiming has sent you an email with a minute file attached, and we have received several files from you
- we've discussed about the target group.
- We've been working on analysing more similar products and writing requirements of

our product and also come up with several new ideas. They'll be declared later in the meeting

- we've designed a survey in order to gather and verify requirements. This will be discussed later.
  - We've decided to use Agile method in order to include more client participation. However, since we are new in SE, somehow detailed plan will also be made to help manage the whole process.
  - we've done personality test and got more familiar with each other

does anyone has any questions in this part?

### 3. discussion (Expected 20m)

#### 1. About agenda (Expected 3m)

- The one I gave you is written based on the template you provided. but we've got a more detailed formal English one like the one Yingming sent you. Do you think the simple one is quite enough or could you give us any suggestions on the one we provided. (**Show her the one we provided**)
- We've looked through the report of 1617 group. We found they attached their minutes in the report. Shall we do the same thing?
- I remember Dave said something about using LaTeX. Is it possible or necessary to use LaTeX for the report or even the agenda and minutes?

#### 2. About target group (expected 5m)

- based on our experience, it would be a great help if we could play with an animated sorting algorithm app when we first time get in touch with them especially in our first year of university
- so we come up with an idea that the target group could mainly be university students majored in computer science or any other course may need knowledge of sorting algorithms and students who are interested in this.
- Under this circumstance, it would be easier to do user research. We can simply gather requirements right in our uni since we have SESA staff Yani. She said SESA could help with that.

#### 3. About survey (**Show her the survey**) (expected 2m)

- Based on different background knowledge people have, the survey would automatically display related questions to the specific group of people
- Any suggestions on the survey we made

#### 4. About requirements (**Show her the requirements**) (expected 6m)

- This is a very early stage version only for preview
- We focus more on innovations which are brand new features among similar products
- As for correctness, we are still not clear about it. Do you mean we need to tell user that this sorting algorithm can do the sorting task correctly? Can you give us a more specific example on this.

#### 5. About platform & language (expected 2m)

- Single platform or multi-platform?
- which platform you prefer us to do on or based on the survey/research?

#### 6. About ethic forms (**Show Dr. Heshan the ethic form**) (expected 2m)

- shall we write all of our names here and submit it as a team?

- o when can we deploy the survey? until the ethic forms are all completed?

Any questions for this part?

#### 4. Q&A (expected 6m)

---

## Minute

---

## Outcomes

(summary of all discussed main points, all decisions, all action points)

### 1. Discussion

- o Dr. Heshan would like agendas be in her template's style. So what every chairperson has to do is edit the agenda in the style and send it to Dr. Heshan before the meeting.

### 2. Review

- o Dr. Heshan is overall satisfied with our process.
- o About survey
- o Dr. Heshan likes our draft questionnaire. Her advice is put English in front of Chinese, because the official language is EN.
- o Dr. Heshan mentioned correctness is a difficult point, but in this meeting we are still not clear about correctness.

### 3. About requirements

- o Dr. Heshan thinks we are not dealing with requirements, but designing the software. This is too far from current schedule. She says we seem to be already having the software model, writing requirements based on this. This is not in line with the software engineering process.
- o She gave us these suggestions: don't write "click", "button" in the requirements, it is better to describe which functions it would implement; also, distinguish functional requirements and non-functional requirements; finally, write out the most important requirements, instead of show the process.

### 4. Next meeting (informal)

**Chairperson:** Yani

**Secretary:** Yuting

**Time:** 27 Oct 2020

---

## Comments

---

Due to the time limit, ethics and correctness were not discussed clearly.

I have problem on preparations on the requirements.

To save paper, don't bring hard copy, email Dr. Heshan instead.

Chairperson and secretary shall sit around Heshan. This would help.

Chairperson shall control the time adequately. If the situation is losing control, secretary shall help speed up.

For speeding up the whole process, it's ok to read agenda. Still, try hard to perform naturally.

# Team 10 - 8th Meeting

---

## Information

---

**Time:** 2020.10.27 (45 min)

**Location:** Library

**Chairperson:** Yani Huang

**Secretary:** Yuting Jiang

**Translator:** Yuting Jiang

**Attendance:** 6/6

<b>Members</b>	<b>Yiming Tang, Shiliang Chen, Yani Huang, Ruizi Han, Yijie Lu, Yuting Jiang</b>
Late	0
Absent	0

---

## Agenda

---

*The whole meeting is expected to take 45 minutes.*

1. Complete the requirement survey . (25 min-30 min)
    - Introduction of the survey.
    - Questions about correctness.
    - Specify the total number of the survey we expect to collect.
    - Decide who is going to have meeting with the "Introduction of Algorithm" lecturer.
  2. Check the site together. (5 min)
  3. Discuss topic for the next formal meeting. (10 min- 15 min)
-

# Minute

---

## Outcomes

1. Abridge and confirm the questions in the survey (About 20 minutes).
  - Delete Q5 and the UI question.
  - Change the options of Q4, Q7, Q10, Q11.
2. Think about the requirement (About 30 minutes).
  - Animated display function(control the speed, last step and first step. stop button).
  - The tutorial for beginners.
  - Multi-language.
  - What kind of sorting algorithms will be include? (six provided, multi-choices for users to choose).
  - Display the history record for users.
  - Cheat sheet (brief notes).
  - Questions about correctness.
3. Discuss the form of the proof the correctness (About 20 minutes).
  - The first version: do not include the abstract concepts(e.g. partial correctness), just prove that for any type of legal input, the algorithm will show the correct output. And for the illegal inputs, the algorithm will show the error or warning.
  - The second version: detailed introduce the necessary concepts, maybe include the proof rules. Explain concepts through intentional code failures. And user chooses whether to display the proof or not.
4. Check the site together (About 10 minutes).
  - Confirm the personal information.
  - Confirm the documents post on the site.
  - Have a public link for our Github.
5. Decide who will take the interview with the IPA module convenor (About 2 minutes).

**Chairperson:** Yuting JIANG

**Secretary:** Yijie LU and Ruizi HAN

**Time:** 10:00a.m. on October 29th

## What is going to be discussed in the next meeting:

- Review what has been finished.
  - Check the info sheet and survey and the interview questions.
  - Decide when to release the survey and when to take the interview.
  - Discuss which version of correctness proof is better.
-

## Work summary for last stage

Task	Members	Report	Question	Completeness
Finish the ethic forms and info sheet.	Ruizi HAN and Yuting JIANG	/	How to submit	90%
Survey design.	Yani HUANG and Yijie LU	/	Correctness Questions and introduction part and swap the language order.	60%
Finish the basic design of our website.	Yani HUANG and Yiming TANG	/	/	100%

## Tasks for next stage

Task	Members	DDL
Finish the questions of survey and write an English version.	Yani HUANG	10/28 12:00p.m.
The introduction part of survey.	Yijie LU and Ruizi HAN	10/28 12:00p.m.
Write the two versions of correctness proof plan.	Shiliang CHEN, Yijie LU, Yuting JIANG and Yiming TANG	10/28 12:00p.m.
Prepare the interview.	Shiliang CHEN, Yiming TANG and Yani HUANG	/

## Problems

Priority(0 for highest, 5 for lowest)	Problem Description	Possible consequence	Proposer	Tentative Solution	Expected completion time
0	How to display the proof of sorting algorithms?	Cannot finish the requirement document.	All	Two ideas(detailed above).	10/29

---

## Comments

---

The meeting proceeded smoothly but was not completed within the expected time. Maybe control the time more precisely next time.

# Team 10 - 9th Meeting

---

## Information

---

**Time:** 2020.10.29 10:00~10:30

**Location:** Online

**Chairperson:** Yuting Jiang

**Secretary:** Yijie Lu and Ruizi Han

**Translator:** Yijie Lu and Ruizi Han

**Attendance:** 7/7

<b>Members</b>	<b>Heshan Du, Yiming Tang, Shiliang Chen, Yani Huang, Ruizi Han, Yijie Lu, Yuting Jiang</b>
Late	0
Absent	0

---

## Agenda

---

*The whole meeting is expected to take 30 minutes.*

*Materials: agenda, minutes of last meeting, participant consent form and information sheet, questions for the interview, questionnaire, two plans about how to prove the correctness, SE PPT(lec3).*

1. Review what we have finished (About 5 minutes).
  - The ethic form
  - The minutes
    - For the report, which template is better?
    - Shall we send the minutes of last week meeting to heshan?
  - The website (introduce briefly)
2. Check the materials for interview and survey (About 8 minutes).
  - Check the participant consent form and information sheet.
  - Check the questions for the interview.
  - Check the survey.
  - Confirm whether we can release the questionnaire and make an appointment with the

module convener.

3. Decide which idea for correctness proof is better (About 15 minutes).
  4. Discuss the requirements specification and definition (About 5 minutes).
    - How to document the requirements?
  5. Q&A (About 2 minutes).
- 

## Minute

### Outcomes

1. Review what we have finished (Time spent: 30 minutes).
  - The minutes
    - For the appendix of report, both minute templates are ok, it depends on us. We decide to use our template.
    - The minutes do not need to be sent to supervisor every week, until there is something needs to be confirmed.
  - Website checking
    - Grammar and vocabulary problem of introduction: 1) the first long sentence need to be split into bullet points. 2) change the sentence "project is promising", it is not suitable to say a project is "promising". 3) change the sentence "the software is the entry level".
    - Content problem of introduction: 1) the key point should be the property of the software (like open source) and the aim of the software (like help users to understand and prove the correctness of sorting algorithm). 2) more types of the sorting algorithm should be mentioned: bubble, quick, merge...
    - Diagram design: 1) Gantt chart: add the dependency between the tasks. 2) Prototype: a little similar to previous work, it needs to be modified and be more innovative to impress users.
    - Team role: 1) We need to add a software developer. 2) The appropriate method is that the team can be divided into small group and take responsibility to each function of the software or a specific sorting algorithm and put them together in the end.
2. Check the materials for interview and survey (Time spent: 20 minutes).
  - Participant consent form and information sheet
    - Check each form whether the name of university is correct: "Ningbo China".
    - delete the bracket of "omit ...".
  - Questions for the interview
    - If we need to ask question 2, we may need to prepare a prototype for the teacher's better understanding.
    - Before the interview, we need to do some preparation, such as provide some slides

and book a room.

- Questionnaire
  - Q11: Change "which one do you think is useless" to "which one do you like".
  - Q12: Make sure all the data we collected is useful for the requirement specification later: Q12 needs to be deleted.

3. Decide which idea for correctness proof is better (Time spent: 15 minutes).

- Use the first proposal: simple one, we need to do it as simple as possible. We'd better not mention abstract concepts in the software since they are too complex for beginners. We can see how it goes firstly, then think about whether we need to add those more complicated knowledge.
- Applying the first proposal, we can focus on the mathematical induction to prove sorting algorithms' correctness.
- Applying the first proposal, the input does not need to be very complicated, and integer is fine for easy understanding. For illegal input, we can choose characters: A,B,C.

4. Discuss the requirements specification and definition (Time spent: 10 minutes).

- Choose the latest version which we have learned in FSE. Follow the FSE powerpoint and textbook.

5. Time control problem (Time spent: 10 minutes).

- Meeting Time should be controlled in 30 minutes. Yiming Tang will be the one who control the time of meeting, he will interrupt the discussion when the time is not enough.
- Yani Huang suggested that we may need to make a brief introduction for what we have done in each week firstly, so time can be leave for more essential topic.

6. Preparation for next meeting (Time spent: 2 minutes)

- **Chairperson:** Yijie Lu
  - **Secretary:** Shiliang Chen
  - **Time:** 11/3 19:30
  - We may discuss our team role again.
- 

## Last stage's action points review

Task	Members	Report	Question	Completeness
Ethics form related documents	Ruizi Han and Yuting Jiang	/	/	100%
Finish the questions of survey and write an English version.	Yani Huang	/	/	100%
The introduction part of survey	Yijie Lu and Ruizi Han	/	/	100%
Write the two versions of correctness proof plan.	Shiliang Chen, Yijie Lu, Yuting Jiang and Yiming Tang	/	/	100%

## Action points

Task	Members	DDL
Modify the website introduction.	/	11.3
Redesign Gantt chart and prototype.	Yiming Tang, etc.	/
Fix some problems of consent form and information sheet.	Yuting Jiang	10.29
Discuss and prepare more interview questions.	All	11.3
Finish the final version of questionnaire.	Yani Huang	10.29
Produce the English questionnaire document for submitting.	Ruizi Han	10.29
Specific timeline of the project	Shiliang Chen	11.3
Literature review	All	/
Interview presenting slides	/	/

## Problems

<b>Priority (0 for highest, 5 for lowest)</b>	<b>Problem Description</b>	<b>Possible consequence</b>	<b>Proposer</b>	<b>Tentative Solution</b>	<b>Expected completion time</b>
2	Time control of formal meeting	Low efficiency and quality of formal meeting and waste of everyone's time	All	Have someone control the meeting schedule. Make a brief introduction for what we have done in each week firstly.	/
3	Unreasonable arrangement of members' team role	Difficult in software developing phase.	Heshan Du	We will discuss everyone's team role in next informal meeting.	11.3

---

## Comments

1. The meeting time should be tight controlled next time. The chairperson should find out suitable time to interrupt the discussion when time is not enough.
2. Before formal meeting, the chairperson should send relevant materials to the supervisor as early as possible and kindly remind supervisor to have a glance in advance.
3. Reduce the content discussed in the formal meeting. Do not discuss less important content in the formal meeting.

# Team 10 - 10th Meeting

---

## Information

---

**Time:** 2020.11.03 19:30 - 22:00(150 min)

**Location:** Library

**Chairperson:** Yijie Lu

**Secretary:** Shiliang CHEN

**Translator:** Shiliang CHEN

**Attendance:** 6/6

<b>Members</b>	<b>Yiming Tang, Shiliang Chen, Yani Huang, Ruiz Han, Yijie Lu, Yuting Jiang</b>
Late	0
Absent	0

---

## Agenda

---

*The whole meeting is expected to take 150 minutes.*

1. Confirm the structure of literature review . (60 min-70 min)
    - Introduction of the literature review.
    - what part should be added to the review.
    - Discuss what we have found for the review.
  2. Discuss which group should take responsibility to Focus Group and which one should take responsibility to interview. (5 min)
  3. Discuss the timeline, confirm how many UML diagram should be used and what prototype should be.(60 min)
  4. Discuss topic for the next formal meeting. (10 min- 15 min)
-

# Minute

---

## Outcomes

1. Literature review part
  - Yijie has provided the structure of literature review
  - After analyzing some of the existing products, we found them very similar. We could take the advantage from them, even they are quite common design
  - However, we can optimize the animation, data visualization and algorithm visualization
  - We can also optimize by improving the ability of interaction
  - Question: Do we need to add papers
  - Answer: We could. By proving our software is useful in expanding description
2. Arrangement of focus group
  - three groups will be there for discussions.
  - Y1 with little knowledge
  - Y2 with knowledge but does not understand correctness
  - Y3 with abundant knowledge with correctness
3. Timeline
  - review the timeline
  - make sure everything goes fine under the timeline
  - gantt chart will be later made according to the timeline
4. Requirements
  - gather requirements: functional and non-functional and user story
  - specification: UML(use case, class, sequence), with rapid prototype
5. Morning meeting
  - yiming will maintain the public repo
  - discussed interview questions, questionnaire and focus group
  - email heshan with relevant files
6. Next meeting

**Chairperson for next:** Yani Huang

**Secretary:** Yiming Tang, Shiliang Chen

**Time:** 11/05 10:00

**What is going to be discussed in the next meeting:**

- overview of last week's work
  - report on our literature review work
  - suggestions on requirements gathering
  - Any paper thing in our report?
  - Confirm the timeline
-

## Work summary for last stage

Task	Members	Report	Question	Completeness
/	/	/	/	/

## Tasks for next stage

Task	Members	DDL
study optimization of animation	Ruizi Han	11/5
study data visualization	Yiming Tang	11/5
study algorithm visualization	Yuting Jiang	11/5
study visualgo	Yijie Lu	11/5
study scratch	Yjie Lu	11/5
study maze	Yani Huang	11/5
study algorithm-visualizer and mobile app	Shiliang Chen	11/5
discuss literature review things	All	11/4
complete the literature review	All	11/8
manage focus group	Yani Huang	11/8
confirm the timeline	All	11/4
make gantt chart	Yijie Lu	11/8
prepare for demonstrating ideas of our product	All	11/10

## Problems

Priority(0 for highest, 5 for lowest)	Problem Description	Possible consequence	Proposer	Tentative Solution	Expected completion time
5	similar interface problem	we made similar products	/	make the good part similar, make innovation in other part. Explain it in the report	/
5	Any paper thing in our report	prove our idea firmly	/	to prove our product is useful and add it in the expand description	/
0	DO NOT forget expand description	fail	Yijie Lu	/	/

---

## **Comments**

---

The meeting proceeded smoothly but was not completed within the expected time. Maybe control the time more precisely next time.

# Team 10 - 11th Meeting

---

## Information

---

**Time:** 2020.11.5 10:00 - 10:30(30 min)

**Location:** PMB 449

**Chairperson:** Yani Huang

**Secretary:** Shiliang Chen, Yiming Tang

**Translator:** Shiliang Chen

**Attendance:** 6/6

<b>Members</b>	<b>Yiming Tang, Shiliang Chen, Yani Huang, Ruiz Han, Yijie Lu, Yuting Jiang</b>
Late	/
Absent	/

---

## Agenda

---

*The whole meeting is expected to take 30 minutes.*

1. Review tasks done (8 min)
    1. Literature review: Introduce current work and stage one by one. (4 min)
    2. Progress of requirements (Prepared a time line based on Dave's plan)(4 min)
  2. Get feedback from Heshan of our current work (10 min)
  3. Q&A (does our literature review cover all aspects? Do we need to draw UML other than use case, class, sequence?). (12 min)
- 

## Minute

---

## Outcomes

1. Review tasks done in last week
  - Have begun literature review
  - Yani (Maze, interaction) → Ruizi(animation optimization) → Yiming(data visualization) → Yuting(algorithm visualization) → Shiliang(code visualization) → Yijie (visualgo and scratch)
  - write paragraph while doing literature review with pros and cons (Deadline: Next Monday 2020/11/09)
2. Discuss requirements
  - Don't hurry
  - Prepare prototypes before interim report
3. About interim report
  - Technical research, market research and existing software research should be included in the literature review part
  - Begin to learn latex
  - Begin to do literature review
  - In the introduction/description part, we can use papers to prove our ideas and our project is useful and motivation as well
  - Files of focus group need to be recoded in the appendix
4. focus group
  - it would be 35 minutes with 5 - 6 people joined in
  - raise questions and record
  - make questions easier to understand, make the environment more relaxing
  - time is not decided yet
  - questions for Y1, Y2, Y3 would be slightly different

**Chairperson:** Yiming Tang

**Secretary:** Yijie Lu

**Time:** 11/10 19:30

**Remark:**

- check literature review for every one
- feedback of focus group
- Prepare for the introduction part
- discuss Latex
- stress the importance of git (introduce project feature of github)
- prototype presentation

---

## Work summary for last stage

<b>Task</b>	<b>Members</b>	<b>Report</b>	<b>Question</b>	<b>Completeness</b>
work on literature review	All	/	write paragraphs	50%
manage focus group	Yani Huang	/	keep going	50%
make gantt chart	Yijie Lu	/	/	0%

## Tasks for next stage

<b>Task</b>	<b>Members</b>	<b>DDL</b>
Literature review	All	11/9
make gantt chart	Yijie Lu	/
manage focus groups	Yani Huang	/
learn LaTex	All	/
Prepare for the introduction part	All	/
Prepare for prototype	Yani Huang, Yiming Tang, Shiliang Chen	11/10

## Problems

<b>Priority(0 for highest, 5 for lowest)</b>	<b>Problem Description</b>	<b>Possible consequence</b>	<b>Proposer</b>	<b>Tentative Solution</b>	<b>Expected completion time</b>
0	Review part should be done by the chairperson of last time	/	Yiming	/	/

---

## Comments

The meeting proceeded smoothly and was completed within the expected time.

The whole meeting today lasts about 24 min, time control is getting better but chairman of future meeting still need to pay attention when the content of the meeting is complex.

Content of meeting could be further simplified, report the progress in a briefer way.

# Team 10 - 12th Meeting

## Information

**Time:** 2020.11.10 (60 min)

**Location:** Library

**Chairperson:** Yiming Tang

**Secretary:** Yijie Lu

**Translator:** Yiming Tang, Yijie Lu

**Attendance:** 6/6

<b>Members</b>	<b>Yiming Tang, Shiliang Chen, Yani Huang, Ruizhi Han, Yijie Lu, Yuting Jiang</b>
Late	/
Absent	/

## Agenda

*The whole meeting is expected to take 60 minutes.*

1. Review of last week
2. Questionnaire
  1. See the results
  2. What is the results
  3. Depending on these results, what should be changed in the current requirements
  4. When to end
3. Are we following the full year timetable plan?
  1. It seems that nobody is taking control of the whole process.
4. internal report
  1. divide the work
5. Prototype
  1. According to the existing resources, it may have to be redone. At that time, I did it according to the idea of the game. But now, if we want to abandon it, first, is Heshan

willing to (because she said at that time that she saw our idea was very interesting, and she wanted to make a game for us), and the second reason was not enough. I think if we decide to give up, it is better to give her some reasons in detail. This part can also be written in the internal report.

2. Question: can it correspond to the requirement we wrote?
  3. Collect your general and abstract ideas about prototypes, and I will make them into pictures.
  4. Although we don't know what platform, we can do the desktop version first. Then design something else.
  5. Linkage of the following [demonstration mode]
  6. interview
    1. I heard that AJ is not easy to get in touch with on focus group, so how do we design it? Or when? I think it will be the end of the term if you don't make an appointment with him, or you can interview other teachers
    2. When to send email, rehearsal and interview;
      - Find another teacher!
      - If teachers want to use it, do you need to distinguish [demonstration mode (used by teachers)] and [learning mode (used by students)]
  7. UML
- 

## Outcomes

1. Review about last meeting (About 5 minutes).
2. Discuss questionnaire (About 30 minutes).  
Most users prefer:
  - Focus on functionality
  - 15 min -30 min using time
  - on the PC platform.
  - good animation.
  - understand the principle, definition and code of sorting algorithms
  - C, Java, Python language using
  - functionality priority
3. Timeline management (About 5 minutes).
  - Shiliang takes charge.
4. Confirm prototype (About 10 minutes).
  - After giving the requirement, Yiming will take this job
  - We do not use game, which needs to be confirmed with heshan
5. Interview (About 10 minutes).
  - Shiliang will write letter to AJ
  - We might also talk with Heng Yu

**Chairperson for next:** Ruizi Han

**Secretary:** Yijie Lu and Yuting Jiang

**Time:** November 12th

### **What is going to be discussed in the next meeting:**

- Reiterate the prototype is done by us, and Yiming introduced some works he did
- Report the questionnaire, pick some important question to discuss, importantly mentions we do not consider game anymore and explain why
- confirm the platform we use(PC)

---

### **Work summary for last stage**

Task	Members	Report	Question	Completeness
Focus Group in Y1 and Y2	All	/	/	100%
Literature review	All	/	/	50%
Questionnaire	All	/	/	80%

### **Tasks for next stage**

Task	Members	DDL
Questionnaire report	Yuting and Shiliang	11.12
Technical research paper	Ruizi and Yiming	11.12
Focus group in Y3	All	11.12
Prepare for the focus group question	shiliang	11.12
Prototype	Yiming	/
Use case diagram	Yani	/
Functionality report	Yijie	11.11
Email to AJ	shiliang	11.10

## Problems

Priority(0 for highest, 5 for lowest)	Problem Description	Possible consequence	Proposer	Tentative Solution	Expected completion time
3	everyone should pay attention to wechat	cannot control time well	All	1 idea	every meeting

## Problems

Priority(0 for highest, 5 for lowest)	Problem Description	Possible consequence	Proposer	Tentative Solution	Expected completion time
/	/	/	/	/	/

## Comments

Now I feel that the moderator should do such work: summarize the work, throw out the topic, cause discussion, draw results, and promote the topic.

Because the first two projects need Yani, and she didn't come at first, so I did the later project first.

We identified the target population: the beginner of sorting algorithm.

After the meeting, we determined the requirements of the final version of the large document and the content to be reported next time.

Write something else I want to say: I found that when I made a prototype, even if I made a diagram, people didn't necessarily understand what I meant. So we have to explain them. Of course, pictures are better than no pictures.

# Team 10 - 13th Meeting

## Information

**Time:** 2020.11.12 10:00~10:30

**Location:** PMB 449

**Chairperson:** Ruizi Han

**Secretary:** Yijie Lu, Yuting Jiang

**Translator:** Yijie Lu, Yuting Jiang

**Attendance:** 7/7

<b>Members</b>	<b>Yiming Tang, Shiliang Chen, Yani Huang, Ruizi Han, Yijie Lu, Yuting Jiang</b>
Late	0
Absent	0

## Agenda

The whole meeting is expected to take 30 minutes.

1. Report the result of questionnaire (About 5 minutes)
  - Shiliang Chen and Yuting Jiang take this part
  - Present some important result
2. Present the outcome of Y1 and Y2 focus group (About 5 minutes)
  - Ruizi Han and Yani Huang take this part
  - Describe some innovative ideas
3. Discuss the software design direction (About 5 minutes)
  - Interesting, like game, or more focus on functionality?
4. Report what we will do in next phase (About 10 minutes)
  - Interview
  - What we will do after finish requirement gathering
  - Quick look at our functional requirements
  - Talk a little about our prototype
    - First explain that the prototype in bid document is designed by Timing Tang.

- Yiming Tang introduce the initial version of prototype
  - Literature review
  - Timeline
5. Rasing questions and allocate tasks of next stage
- Raise questions
  - Decide the chairperson and secretary of next meeting
  - Decide the specific time of next meeting
  - Review all new action points, confirm and go through next stage's tasks

---

## Minute

---

## Outcomes

1. Questionnaire report (Time spent: 5 minutes)
  - Yuting reported 6 important information we have collected, and Heshan said the work is fine
  - Heshan suggested that the report can be written by LaTex.
2. Focus group information collecting (Time spent: 5 minutes)
  - Ruizi said Y1's ideas are not quite useful
  - Ruizi said Y2's ideas are relative useful, and we will consider to design a software has more functionality rather than has more game elements
  - Heshan agreed with ruizi.
3. Discuss the software design direction (About 5 minutes)
  - Heshan said that the previous prototype's selecting module like the visualgo's.
  - Yiming showed updated prototype, he will delete the skin shop and other game elements.
  - Shiliang stressed that we will focus on the functionality of the software, and the game mode will be deleted.
  - Heshan said our design pattern is fine, but the prototype should be written in English
4. Report what we will do in next phase (About 10 minutes)
  - check the literature review
  - feedback of focus groups
  - Prepare for the introduction part
  - stress the importance of git (introduce project feature of github)
  - prototype presentation

### Next meeting:

- **Chairperson:** Yuting Jiang
- **Secretary:** Yani HUANG

- **Time:** 11/17

## Last stage's action points review

Task	Members	Report	Question	Completeness
Questionnaire report	Yuting and Shiliang	/	/	100%
Technical research paper	Ruizi and Yiming	/	/	not finished
Focus group in Y3	All	/	/	100%
Prepare for the focus group question	shiliang	/	/	100%
Prototype	Yiming	/	/	not finished
Use case diagram	Yani	/	/	not finished
Functionality report	Yijie	/	add non-functional	50%
Email to AJ	shiliang	/	/	100%

## Action points

Task	Members	DDL
Finish the literature review	Yijie LU, Ruizi HAN	11.17
Finish the focus group report(Year 3)	Shiliang Chen	11.17
Find out resources about technical analysis	Yiming TANG and ALL	11.17
Informal talk with Heng YU	Yijie LU, Yuting JIANG	11.13
Informal talk with Dave	Yani HUANG, Shiliang CHEN	11.18
Second version of prototype	Yiming TANG	11.17
Function documentation(non-functional)	Yijie LU	11.17
Release and collect questionnaires(for Year 1 students)	Shiliang CHEN	11.17
Finish the final questionnaire report	Yuting JIANG	11.18

## Problems

Priority(0 for highest, 5 for lowest)	Problem Description	Possible consequence	Proposer	Tentative Solution	Expected completion time
1	Decide whether to quit game mode	Affect the prototype design and functions documentation	All	Discuss in next meeting	/

---

## Comments

---

For today's meeting, I prepared a draft script because I am really nervous about this formal meeting. Hence this led to an awkward situation, I paid too much attention on finishing what I planned to say, and did a poor job in responding supervisor and asking for more feedback. I didn't help the group to get useful information from supervisor, I should focus on interaction more next time and keep thinking during the meeting.

# Team 10 - 14th Meeting

---

## Information

---

**Time:** 2020.11.17 20:00~21:00

**Location:** Library

**Chairperson:** Yuting JIANG

**Secretary:** Yani HUANG

**Translator:** Yani HUANG

**Attendance:** 6/6

<b>Members</b>	<b>Yiming Tang, Shiliang Chen, Yani Huang, Ruizi Han, Yijie Lu, Yuting Jiang</b>
Late	0
Absent	0

---

## Agenda

---

*The whole meeting is expected to take 90 minutes.*

1. Report the process of correct deadlines (About 15 minutes)
  - Literature review
  - Function documentation
  - Focus group report(total)
  - Prototype
2. Questionnaire analysis (About 10 minutes)
  - Focus on the different results
3. Website update (About 5 minutes)
4. Technical research (About 15 minutes)
  - Summarize current process and next step
  - Maybe focus on OS, Programming Language, Hardware, Software...
5. Description of Problem to be solved (About 5 minutes)
6. Check user stories (About 10 minutes)
7. UML diagrams (About 25 minutes)
  - Check current process (use case and sequence)
  - Assign tasks

- 
8. Next meeting prepare (About 5 minutes)

---

## Minute

### Outcomes

1. Yani and Shiliang ask Dave to comment on the current schedule
  2. Correction: (Yuting adds the concept) Algorithm which terminated when putting legal inputs can be seen as algorithm fits the correctness rules. Instead, doesn't fit.
  3. Yijie is responsible for writing expanded description
  4. UML: Use case diagram: Shiliang, Yani; Sequence diagram: Ruizi
  5. Prototype: change to English, add small logo and explanation (such as "learned recently")
  6. Survey: Yuting is responsible for updating the data to the report
  7. Report: Yuting and Yani integrate report using LaTeX.
  8. Website maintenance: Yani
  9. Requirement document: Yijie is responsible for updating requirement related documents.
- **Chairperson:** Yiming
  - **Secretary:** Shiliang, Yani
  - **Time:** 2020/11/20

---

### Last stage's action points review

Task	Members	Report	Question	Completeness
/	/	/	/	/

---

### Action points

Task	Members	DDL
/	/	/

## Problems

Priority(0 for highest, 5 for lowest)	Problem Description	Possible consequence	Proposer	Tentative Solution	Expected completion time
/	/	/	/	/	/

---

## Comments

---

Everything is finished except the technical research. The directions of technical research still need to be discussed in next meeting. And we need to decide which programming language we will use as soon as possible.

# Team 10 - 15th Meeting

---

## Information

---

**Time:** 2020.11.19 10:00-10:30(30 m)

**Location:** PMB449

**Chairperson:** Yiming Tang

**Secretary:** Shiliang Chen & Yani Huang

**Translator:** Shiliang Chen

**Attendance:** 7/7

<b>Members</b>	<b>Yiming Tang, Shiliang Chen, Yani Huang, Ruizi Han, Yijie Lu, Yuting Jiang Dr. Heshan</b>
Late	0
Absent	0

---

## Agenda

---

*The whole meeting is expected to take 30 minutes.*

1. Openning (1m)

Draft of Literature review is done, Interim Report and Use Case are under writing.

2. report of progress (20m)

1. Literature Review

Have a look at draft.

2. Requirements

Confirm Functional Requirements.

Other:

- UML and Use Case will be done by Shiliang and Yani

- Sequence will be done by Ruizi
3. questionnaire report (2m)

We have interviewed the lecturer of Introduction to Programming and Algorithms. He helped to distribute the questionnaire. Please refer to the questionnaire report for detailed updates.

4. interview report (5m)

The interview report has sent to you. Any suggestions?

Shiliang will introduce a bit.

5. Prototype (10m)

Prototype is still under drawing. We have changed our mind, so now there is only one picture of it.

6. otheres

Website maintainer Yani

GitHub repository maintainer Yiming

## Minute

---

### Outcomes

1. Literature review

- picture in LaTex has some issues
- citation style issue
- citations are not all included
- grammar mistakes
- reference list style shall be consistent
- a short introduction at the beginning of each module
- Introduce a software with its name
- similar software shall be put together
- URL as footnote
- scratch picture in English
- grammar mistakes shall be fixed

2. requirement document

- introduce concept of requirements
- non functional one will be consulted with BGL
- requirements shall not be too detail
- multilingual issue
- grammar mistakes shall be fixed

3. expanded description

- fix the third part issue
- grammar mistakes shall be fixed

4. prototype

- introduced by email

**Chairperson for next:** Shiliang Chen

**Secretary:** Ruizi Han

**Time:** November 24th

**What is going to be discussed in the next meeting:**

Revison:

1. use case diagram
2. requirement
3. expanded description
4. literature review
5. technical research

---

### **Work summary for last stage**

Task	Members	Report	Question	Completeness
/	/	/	/	/

### **Tasks for next stage**

Task	Members	DDL
improve use case diagram	Shiliang and Yani	11/20
improve requirement	Yiming	11/22
improve sequence diagram	Ruizi	11/23
improve literature review	Yijie	11/23
technical research	All	11/23
improve expanded description	Yiming	11/23
risk management	Yuting	11/23
improve questionnaire report	Shiliang	11/23
improve focus group report	Yuting and yani	11/23
consult BGL for use case	Yani	11/20
consult Dave 1. gap in literature/application 2. key work	Shiliang	11/19
problem encountered	/	11/20

## Problems

Priority(0 for highest, 5 for lowest)	Problem Description	Possible consequence	Proposer	Tentative Solution	Expected completion time
/	/	/	/	/	/

---

## Comments

---

/

# Team 10 - 16th Meeting

---

## Information

---

**Time:** 2020.11.24 19:30-22:00 (2h30m)

**Location:** Library Project Room 11 2F

**Chairperson:** Shiliang Chen

**Secretary:** Ruizi Han

**Translator:** Yiming Tang & Ruizi Han

**Attendance:** 6/6

<b>Members</b>	<b>Yiming Tang, Shiliang Chen, Yani Huang, Ruizi Han, Yijie Lu, Yuting Jiang</b>
Late	0
Absent	0

---

## Agenda

---

*The whole meeting is expected to take 150 minutes.*

### 1. Review

Yiming Tang reviews last week

### 2. Document progress report

Introduce [the principle of alternation], to minimize the error probability

#### 1. Literature Review

1. Lu briefly talks about the current progress, how the modification is, if there are any problems that need help
2. arrange the next reviewer
3. If fully relavent, Do context and motivation have key work
4. Focus Group

- Jiang and Huang briefly talk about the revision progress of the focus group report,
  - arrange the next reviewer
5. Risk Management
    - Huang will talk about how the risk management and control module is done, whether it is finished or semi-finished, and then arrange for the next reviewer
  6. Encountered Problems
    - Jiang would like to introduce the production progress of this piece, and then arrange the next reviewer
    - Identification and description of problems, clear analysis including causes, actions and causes, including remedial measures
2. Expanded description
    1. Tang Yiming talks about the progress of the extended description modification, how to modify the third part, what content to put in, and then arranges for the next reviewer
    2. Contextualised with clear explanation of the gap in literature/application, and how the GRP project will fill this gap.
  3. Questionnaire report
    1. Chen talks about the progress of the revision of the questionnaire report, and then arranged for the next reviewer
  4. Sequence diagram
    1. Han ruizi will introduce the progress of the sequence diagram, if there are any problems, and confirm with you
    2. If there is no problem, it should be submitted to the next person for review. If there is any problem, continue to modify it
  5. Prototype
    1. Tang Yiming shows prototype progress
    2. Emphasize the need for a clear and detailed description and explanation
  6. Requirements Validation
    1. Tang Yiming talks about the progress of the modification of the requirements document. Everyone pass them one by one (have time), and another person reviewed it (do not have time)
    2. Use case diagram
      - Huang Yani will talk about the progress of use case diagram, BGL's reply, and then arrange for the next reviewer
    3. Interview report
      - arrange for the next reviewer
    4. User story
      - arrange for the next reviewer
  3. Technical research & architecture design
 

Report on each member's progress and arrange the next detailed plan and personnel arrangement,

- Determine OS, Programming Language, Hardware, Software and reasons
- Platform, tool, technology, algorithm, data structure

#### 4. Overall progress report

Some pictures

Utilization of GitHub team work

Tell me what to do at the next meeting

Assign candidates for next meeting

Restate the task of everyone today

- #### 5.
- Classroom Booking
  - Readme to update host
  - The Secretary asked GitHub to assign tasks
  - Send out the task list that night
  - Everyone should follow up on the progress in time
  - Remember to write a personal diary every day

## Minute

---

### Outcomes

#### 1. Summarize of last formal meeting

- We need to focus more on some detailed part of interim report, such as grammer.
- We should go through those documents ourselves first before the formal meeting.

#### 2. Progress reports for existing documents' revision

- Pair documenting: when one member finished his/her part, another member will help check again.
- Key points for improving and modification:
  - Passage structure and logic
  - Content
  - Grammer
  - Extract important parts which should be put in interim report
  - Use simple short sentences
- Literature Review

| Yijie Lu

- Focus group

| Yuting Jiang and Yani Huang

- Risk Management

| Yani Huang

- Encountered Problems

Yuting Jiang

- Expanded description

Shiliang Chen

- Questionnaire report

Ruizi Han

- Sequence diagram and description

Ruizi Han

- Prototype and description

Yiming Tang

- Latex

Yuting Jiang

- Confirmation of requirements

Confirmed among team members

- Use case diagram

Yani Huang and Shiliang Chen

Consult Bryan about use case diagram

- Interview report

Yiming Tang

- User story

Yijie Lu

### 3. Technical research & architecture design

- Distributed technical research tasks, analyze pros and cons
- Platform: Ruizi Han
- Language: Shiliang Chen
- IDEs: Yani Huang
- Data structure: need further discussion

**Chairperson for next:** Yijie Lu

**Secretary:** Yuting Jiang, Ruizi Han

**Time:** November 26th

**What is going to be discussed in the next meeting:**

1. Confirm requirement with supervisor
2. Show our user story
3. Talk about technical research, briefly introduce our ideas
4. Discuss architecture design

---

## Action points

Task	Members	DDL
Improve and modify literature review part	Yijie Lu	11.28
Improve and modify user story	Yijie Lu	11.28
Improve and modify focus group report	Yuting Jiang, Yani Huang	11.28
Improve and modify encountered problems	Yuting Jiang	11.28
Improve and modify risk management part	Shiliang Chen	11.28
Improve and modify use case diagram	Yani Huang, Shiliang Chen	11.28
Consult Bryan use case related questions	Yani Huang, Shiliang Chen	11.26
Write technicle research - IDE part	Yani Huang	11.28
Prototype	Yiming Tang	11.28
Write prototype description	Yiming Tang	11.28
Improve and modify interview report	Yiming Tang	11.28
Improve and modify expanded description	Shiliang Chen	11.28
Write technicle research - language part	Shiliang Chen	11.28
Write technicle research - platform part	Ruizi Han	11.28
Write description of sequence diagram	Ruizi Han	11.28
Improve and modify questionnaire report	Ruizi Han	11.28
Modify requirements document according to Dr. Du's opinion and email it to Dr. Du	Shiliang Chen	11.27

---

## Comments

This meeting has not been held smoothly.

The main task for this meeting are reporting and discussion and redistribution of items reported. However, the progress was not smooth and the assigned tasks were not well completed.

- Shiliang Chen and Yani Huang's use case diagram has stalled because they haven't ask Bryan.
- Yiming Tanf's requirement document has not been modified.
- Lu Yijie's literature review task is wrong
- Everyone's technical research is progressing poorly

**This may due to the following reasons:**

- The expected time has not claimed before the meeting
- DMS course burden is too heavy
- Classroom's temperature is too high
- Meeting which focus on discussion are more appropriate
- Progress may need to be followed up before the meeting
- Chairperson may supervise the task for a certain period of time after the meeting.

# Team 10 - 17th Meeting

---

## Information

---

**Time:** 2020.11.26 10:00~10: 30

**Location:** PMB 429

**Chairperson:** Yijie Lu

**Secretary:** Ruizi Han and Yuting Jiang

**Translator:** Ruizi Han and Yuting Jiang

**Attendance:** 7/7

<b>Members</b>	<b>Yiming Tang, Shiliang Chen, Yani Huang, Ruizi Han, Yijie Lu, Yuting Jiang</b>
Late	0
Absent	0

---

## Agenda

---

*The whole meeting is expected to take 30 minutes.*

1. Confirm the Requirement (About 10 minutes)
  - Confirm the detail
2. Confirm the user story (About 5 minutes)
  - Confirm the detail
3. Confirm the technical Research (About 10 minutes)
  - pre what sections we will take
  - raise questions about the data structure
4. Rasing questions and allocate tasks of next stage (About 5 minutes)
  - Raise questions
  - announce what works we will show on the meeting next time

---

## Minute

---

# Outcomes

1. Confirm the Requirement (Time spent: 15 minutes)
  - Add definition or explanation of requirement and specification, maybe refer to textbook.
  - 4 priorities are too many, maybe deduct to 2, e.g. 'must' part and 'optional' part. Can also explain why we separate requirements in that way.
  - Send Dr. Du the modified version of requirements before Friday.
  - Be explicit to specific words like 'module'.
  - Confirm with Dr. Boon Giin Lee, is multi-language functional or non-functional requirement.
2. Confirm the user story (Time spent: 5 minutes)
  - Dr. Du think user story should be presented before requirements. This should be further discussed.
  - Assign numbers to requirements and user story.
3. Confirm the technical Research (Time spent: 10 minutes)
  - Data structure:
    - For sorting algorithms' data structure, array and linked list can be used, their time complexity are the same, while space complexity are not.
    - For the data structure of software, it is better to confirm this with Dave.
  - Write advantages and disadvantages of platforms, languages and IDEs, could include preference. Maybe include some citations, refer to website or paper.
4. Rasing questions and allocate tasks of next stage (Time spent: 3 minutes)
  - Draft interim report will be presented.
  - UMLs
5. Preparation for next meeting (Time spent: 3 minutes)
  - **Chairperson:** Yuting Jiang
  - **Secretary:** Yiming Tang
  - **Time:** 12.1 19:30

---

## Last stage's action points review

Task	Members	Report	Question	Completeness
/	/	/	/	/

## Action points

Task	Members	DDL
Improve and modify context part of interim report	All	11.28
Write technique research part	Yani Huang, Shiliang Chen, Ruizi Han	11.28
Modify requirements document according to Dr. Du's opinion and email it to Dr. Du	Shiliang Chen	11.27
Ask Dr. Boon Giin Lee use case diagram related questions	Shiliang Chen, Yani Huang	11.26
Improve and modify use case diagram	Shiliang Chen, Yani Huang	11.28
Improve sequence diagram and write description	Ruizi Han	11.28
Improve prototype and write description	Yiming Tang	12.7

## Problems

Priority(0 for highest, 5 for lowest)	Problem Description	Possible consequence	Proposer	Tentative Solution	Expected completion time
/	/	/	/	/	/

## Comments

Chairperson should make sure to send the agenda to supervisor before the meeting.

# Team 10 - 18th Meeting

## Information

**Time:** 2020.12.1

**Location:** Library Project Room 5

**Chairperson:** Yuting JIANG

**Secretary:** Yiming TANG

**Translator:** Yiming Tang

**Attendance:** 6/6

<b>Members</b>	<b>Yiming Tang, Shiliang Chen, Yani Huang, Ruizhi Han, Yijie Lu, Yuting Jiang</b>
Late	0
Absent	0

## Agenda

*The whole meeting is expected to take 120 minutes.*

Interim report draft

1. Check the cover (About 5 minutes)
2. Introduction (About 6 minutes)  
write intro and check version
3. Background and Related Work (About 6 minutes)  
check version
4. Software Requirements Engineering (About 15 minutes)
  - o Confirm final version of questionnaire report, focus group report, and interview report.
  - o Requirements Specification version check
  - o Confirm final version of UMLs and user story
5. Design (About 15 minutes)  
Discuss the content

6. Implementation (About 15 minutes)  
Discuss the content
  7. Problem Encountered and Risk Management (About 8 minutes)
    - Double check of Problem Encountered file.
    - Double check risk management.
  8. Time line (About 5 minutes)
    - Confirm final version
    - Discuss description
  9. Conclusion (About 5 minutes)
    - Discuss the content
  10. Bibliography (About 10 minutes)
    - Confirm the format
    - Assign all reference
  11. Appendix (About 15 minutes)
    - Discuss the content
  12. Any additional content (About 10 minutes)
- 

## Minute

### Interm Report

Put the minutes in as an attachment

#### Cover:

1. three members first line and three members the second line (put Lu below), with a space between the name and the student number in brackets

#### Main body:

1. Intro
  - intro of intro
2.
  - 2.2.1 TR
  - catalog
  - Case unification
  - 2.2.2 Simple software analysis -> existing similiar software
3.
  - 3.1 requirements elicitation > requirements elicitation and gathering
  - 3.1.1 focus group in front of surveyUser story and requirement are the same level, which are placed in 3.2

- Add a short paragraph of requirements validation, which is confirmed in the group first, and then confirmed with Stackholder and Heshan.
- Diagrams should be put in design

4.

- UML, Low precision prototype
- Javaweb, IDEA, Java
- Don't put data structure
- Build tools, like Maven
- TR three people check each other's, write their own conclusion  
The conclusion of technical research: each point is less than 100 words

5. Implementation

- High precision prototypes
- Relevant decisions should also be put in, and the place Colin wrote about "how to decide to do it" should be put in.

6. Issues and Risk management

7. timeline

- Gantt chart. This may be too long. Turn it 90 degrees or split it into two pages.

8. Conclusion

9. Reference

- sorted out a file where there were references, and unified citation format.

10. Appendix

- Questionnaire content, questionnaire report, focus group report and meeting minutes [need to be sorted out, but if there is no time to sort out, the existing meeting minutes will be directly converted to PDF and put on it]
- Write a little description under each headline.
- Determine the version for everyone.  
The final version of the questionnaire report needs to be simplified.
- need to put a short paragraph between 3.x and 3.x.1.

## Tasks for next stage

<b>Task</b>	<b>Members</b>	<b>DDL</b>
5 Implementation	Tang	12.2
Questions and Risk Management	Lu	12.2
Requirements Validation	Tang	12.2
Cover, Table of contents	Jiang	12.2
Technical Research	Han, Chen, Huang	12.2
Sort out minutes	Lu, Huang, Tang	12.9
Short paragraph between 3.x and 3.x.1	Jiang	12.2
Intro of Intro	Han	12.2

## Problems

Priority(0 for highest, 5 for lowest)	Problem Description	Possible consequence	Proposer	Tentative Solution	Expected completion time
/	/	/	/	/	/

---

## Comments

---

/