The University of Hong Kong

Faculty of Engineering

Department of Computer Science

COMP7704
Dissertation Title
Efficient Object-structure-based E-learning IOS APP (Mind Map Revealing)

Submitted in partial fulfillment of the requirements for the admission to the degree of
Master of Science in Computer Science

By
ZHANG Yingying
3035150294

Supervisor's title and name: Dr. Lucas C.K.Hui
Date of submission: 01/08/2015

The University of Hong Kong

Master Thesis

Efficient Object-structure-based E-learning IOS APP (Mind Map Revealing)

ZHANG Yingying

COMP 7704
Faculty of Engineering
Department of Computer Science

Declaration of the Candidate

I, the undersigned, hereby declare that the work contained in this thesis is my own original work, and has not previously in entirely or in part been submitted at any university for a degree.

Only the source cited in the document has been used in this draft. Parts that are direct quotes or paraphrases are identified as such.

I agree that my work is published, in particular that the work is presented to third parties for inspection or copies of the work are made to pass on to third parties.

The University of Hong Kong,

ZHANG Yingying

Abstract

The e-learning app is a novel idea that using object-based storage instead of original file-based storage which can enhance e-learning effectiveness. Compared with the original storage way, this project uses the object-based storage such as mind map to store all students' class notes in the system. In this project, I have done the design and two versions implementation of the mind map including HTML5 version and native IOS version. Mind map can show all the files in one big picture and it can distribute all the topics in different levels. Using different branches to represent the different parts we learned in the class. The high level nodes in mind map represent larger scope topic we learned. The students can edit the mind map including add a node, delete a node, copy or cut part of mind map, paste these nodes, undo some action and redo some action. Furthermore, they can change the text information of the mind map, the branch color, the location of the node and zoom in or zoom out a mind map view. Beside the basic operation of mind map, students also can filter all the topics. For example, students can search "multiply" topic to show all the node related to the multiply topic. This will help students learn and review more effectively.

Contents

1. Introduction

This research aims to develop efficient software using in the e-learning on the mobile platform including IOS and Android. The whole thesis will proceed as follows. Firstly, I will introduce the background of e-learning and some related work during this project. In section 2, I will discuss the mind mapping which contains the benefit of using mind map and apply the Objects-linkages and revealing to the e-learning software. Section 3 shows the overall design the whole system. In section 4, I will explain all the implementation details including server implementation, web server implementation based HTML5 and mobile client on IOS and Android platform. Section 5 discuss the limitations and future work. Finally, section 6 presents the difficulties encountered during this project before making a conclusion.

1.1 Background

In the traditional learning environment, the class room and the teaching content are all controlled by the teachers and students just as a listener or receiver to receive the knowledge passively. Although this form of education is widely used and understood, there is another form of education that has come into practice with the advent of new technology such as e-forums and online classes. The very use of technology for learning has been found to have a positive effect on the commitment on the students' part to learn the knowledge.

The following figure shows the data come from the Google Trends summarized the

attention rate of e-learning from 2004 to now.

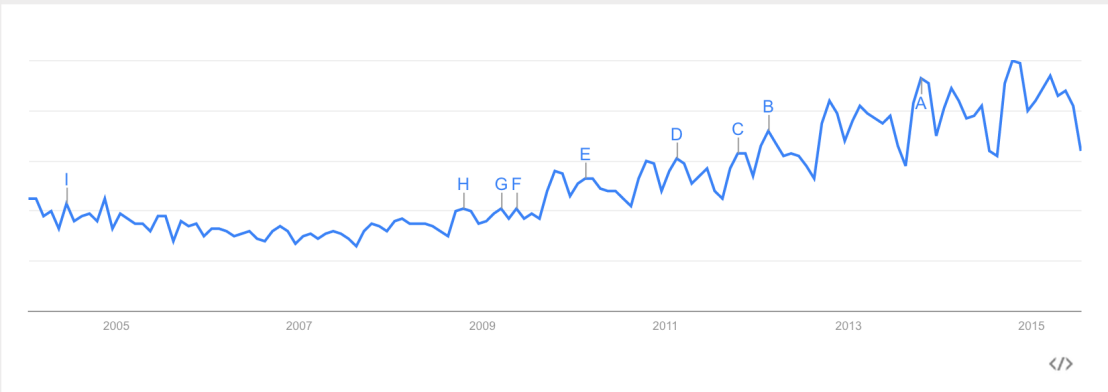

Figure 1-1. Attention Rate of E-learning from 2004 to now. Those data come from the Google Trends.

E-learning [1] integrates technologies and new knowledge to allow learners to learn new knowledge at anytime and anywhere without constraints of time and space. The e-learning is learner-centered and the learning pace can be controlled by the learner themselves not the required subject quota or the fixed curriculum. In general, these E-learning systems present the learning materials in diverse media such as image, sound, video and animation. They give the students a very new learning experience. On the other hand, there are also many advantages of traditional classroom learning compared with the E-learning. First, in the traditional classroom, we can get immediate feedback and we can familiar to both the teachers and students based on the face-to-face communication. Second, learning in traditional classroom can cultivate the students' social skills. Therefore, the best way to improve the students' learning style is to combine them together. We can use the new technology to improve the interaction in the classroom. This project aims to develop an efficient software using in the e-learning to handle these problems. In order to handle the efficient searching in the big e-learning materials, our system use object-based storage to replace the traditional file-based

storage. With the object-based storage, the users can filter subject or topic to find the useful information quickly. Furthermore, with the object relationship, the knowledge we learnt can build the relationship automatically.

With the increasing prevalence of e-learning, many primary schools in Hong Kong have released their own instructions based on e-learning methods, including deploying some mobile devices such as iPad, opening discussion e-forums. The primary purposes are to promote e-learning among students.

1.2 Related Work

In order to develop the efficient object-based e-learning software, we have done a lot research on different aspect which contains the reverse engineer on multiple mobile applications and the cross-platform framework research.

1.2.1 Reverse Engineering

Reverse engineering [2], also called back engineering, is the processes of extracting knowledge or design information from some software and analyze the structure and library to get whole design and implementation of one application. Next I will explain the whole process of reverse engineering on IOS platform.

IOS platform is evolved from the Mac OS X which is based on the UNIX, they have the similar Filesystem Hierarchy Standard. We check all the structure of these file system and find the important part: The Application and Dynamic Library. We first check the entrance of the application in “info.plist” and find the dynamic library when

the dynamic loader load the library to the app memory. Then next we use the tools such as MobileSubstrate, Cycript and SBSettings to get some part information of the application. Using this information, we can get more knowledge about how to develop a new software and learn the application structure more deeply.

1.2.2 PhoneGap Research

PhoneGap [3] is a free and open source framework platform based on HTML, CSS and JavaScript, in which developer could quickly create mobile apps using standard API. PhoneGap supports main mobile platforms such as iPhone, Android, Windows Phone, Blackberry and so on. One of the significant features is that it supports write once run anywhere using its standard web technologies. Besides, PhoneGap also supports many hardware features and native platform features, such as accelerometer, camera, compass, file, geolocation, media, network, notification and so on.

The different mobile platform SDKs of PhoneGap can be downloaded from PhoneGap web site http://phonegap.com/download. In android platform, building application using PhoneGap needs following three steps:

1. Create basic android application with PhoneGap jar library (cordova.jar). In android main activity, it must load initial html web file from local storage using loadUrl("file:////index.html") function.

2. Import PhoneGap library into assets folder, including jQuery library, android specified phoneGap js, and config.xml configure file.

3. Program index.html using jQuery.js and PhoneGap.js. The example of basic html

file like following:

```
<!DOCTYPE HTML>
<html>
  <head>
      <title>Cordova</title>
      <script type="text/javascript" src="cordova.js"></script>
      <script type="text/javascript" src="jquery.js"></script>
      <script type="text/javascript" src="jquery.mobile.js"></script>
      <link rel="stylesheet" href="jquerymobile/jquery.mobile.css" />
  </head>
  <body>
      <script type="text/javascript">
        function test(){
            $.ajax({
                // program here
            })
        }
        test();
      </script>
   </body>
</html>
```

Figure 1-2. PhoneGap example

As can we see from the above simple example file, we only need write functions based on jQuery and API provided by PhoneGap.js. The developers only focus on about business logic and interface design using CSS.

1.2.3 Sencha Touch Research

Sencha [4] Touch is a JavaScript framework for building cross-platform mobile web applications for the mobile devices. Using this framework, we can develop a native-looking HTML5 application using JavaScript and CSS. Sencha Touch provides many UI components for mobile devices and part of them shows in the following figure. We just need to write the JavaScript and CSS script code to develop the mobile programs and run it on multiple platforms.

The Sencha Touch also supports the event handle on the mobile devices including touch

event, scroll event, tap event, double tap, swip event and pinch event.

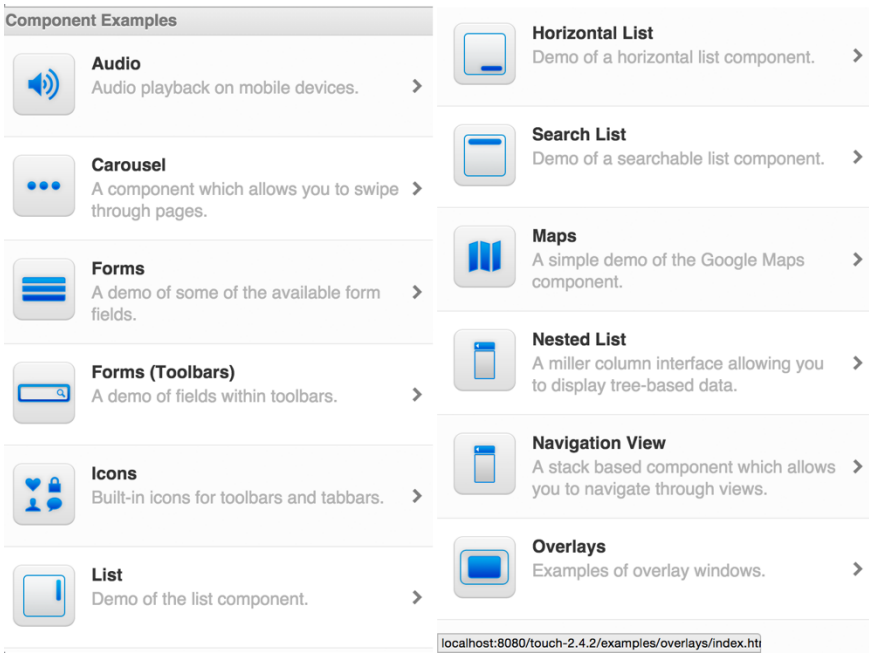

Figure 1-3. Sencha Touch component

2. Dynamic Mind mapping

2.1 Introduction

A mind map [5] is a diagram that connects information around a central subject and it can show the hierarchical relationship between different subjects. A mind map generally has a root node and some children node around the root node. It always represents the ideas in out brain and it can show the text, image and other style. Major ideas are connected to the central root topic. Every first level node has their children node to represent this topic more detailed. It is more effective than other brainstorming or folder format. When we use the mind map, we can get a “big picture” of all the system and have an overview of all the topics. On the other hand, the mind map also

records the relationship between different topics. We can filter the topic and find the information quickly.

2.2 Advantages

The key feature of Mind Map revolutionizes the way we work, learn and think compared to the traditional file-based storage. There are many benefits of the Mind Map using in the e-learning app. Firstly, it optimizes the students' time increases creativity, boosts memory and improves understanding of complex topics. With the e-learning app, students can work on their maps from home, school and even on the go to seamlessly link their classroom and homework activities. Secondly, it can structure the class content. Mind maps can store and structure vast amounts of information. They display hierarchy, show relationships between some topics and enable you to see the "big picture" at a glance. Thirdly, it enhances the productivity. Mind mapping enables students to learn faster, take better notes and brainstorm more effectively. No matter if you're writing homework, studying for an exam or working on a group project, mind mapping will help you save tremendous amounts of time. The e-learning app can search efficiently by different topics. Students also can add some maps to the current mind map to show all the content in one "big picture". Finally, the mind map can support the association link between any nodes. The app can find some new knowledge between these learned topics to help students learning efficiently and deeply.

2.3 Object-based Storage

The main idea of using Object-based storage [6] instead of original file-folder-based storage is because this kind of storage does not require users to remember where they put the nodes. This problem will especially cause when there are lots of notes or students. If students want to store all knowledge they learnt from studying primary school to secondary school, we can handle these big data storage using Object-based storage. Even for only one user, we also need to organize and solve big-data revealing problem. Using file-folder-based storage [7] cannot handle these large notes and we need to remember where we put the notes. It will be a nightmare after store many notes in our learning system. We can use the Object-based storage such as mind map to store the relationship between different topics to record this information automatically and then we can implement efficient e-learning.

2.4 Mind map using in e-learning

With the increasing of files stored in the system, the users will feel troublesome while finding some useful information and cannot get out the old files easily. We do not want this happen in especially on primary and secondary students e-learning. The program wants to store all students' assignments, class notes and questions into the system. The students do not need to care about where to put an assignment and remember the relationship between different notes. The mind map can store this information automatically. Firstly, Mind map can store and structure vast amounts of information. They display hierarchy, show relationships between all the topics. Secondly, it enables

us to see the “big picture” at a glance and find our topic quickly. It optimizes our time, increases creativity, boosts memory and improves understanding of complex topics. Finally, we do not need to remember where we put the notes. It enhances the productivity. The mind map can show the learning sequence and show the different topics in different levels. For example, we have learned the addition of numbers and when we learn the multiply of the numbers, we will find hidden link between additions and multiply. We can build these association link between different topics to find some hidden and useful knowledge to help students learn more effectively [8].

3. System Design

3.1 System Architecture

The entire system can be divided into three main parts. First part is the Server based on PHP, it handles all the requests from mobile app and HTML5 Web Server to operate the database including add a node, modify a node, show a mind map and delete a node. The second part is HTML5 Web Server. It can draw a mind map using HTML5 canvas and send request to PHP Server to get a Graph JSON file and change the JSON file to a mind map. The third part is the Mobile App. The App can be divided into native IOS version and multi-platform version. The native IOS version can run in the IOS platform and multi-platform version can run many mobile platforms including IOS, Android.

The following figure shows the architecture of the system.

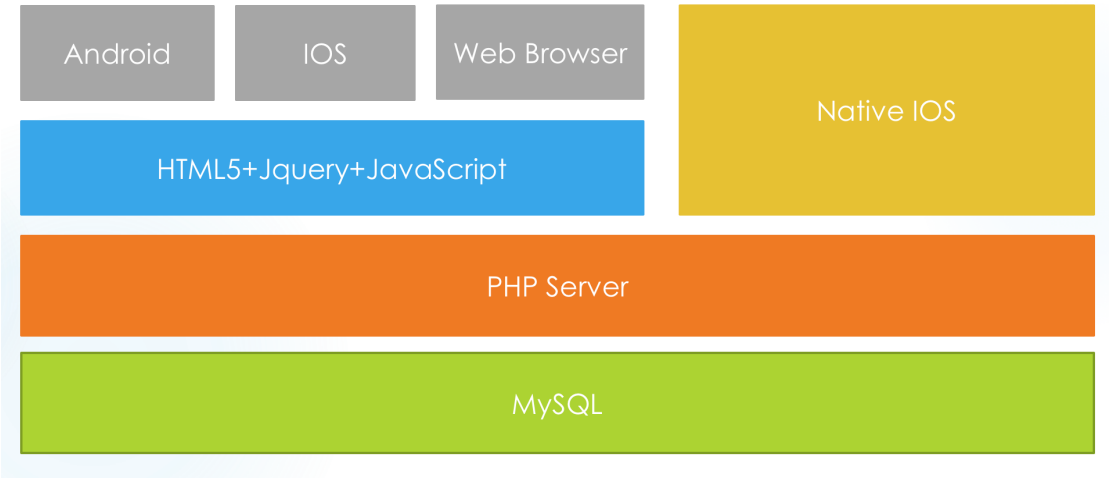

Figure 3-1. System Architecture

3.2 Overall Workflow

The system workflow can be divided into two parts. The first part is the direct communication with the App Client and the PHP Server and the second part is the communication between App Client, HTML5 Web Server and PHP Server.

The figure below shows the direct communication with the App Client and Server based on PHP. If the App Client wants to add a node or delete a node, it will send a post request to the PHP Server via Internet and pass the related parameters. When the PHP Server received the request from App Client, it will parse the parameters and operate the database to get or delete a record from database. The database will return the result to the PHP Server and the server will return the result file to the App Client. Then the App Client will run the program to do the corresponding action. For example, if App Client wants to add a node to the mind map, it will send the “userID”, “nodeID”, “subject”, “topic” and “parentTopic” to the PHP Server and the server will use these

data to check the record in database. If all the data is valid, the node will be inserted to the corresponding subject and parentNode. The whole process shows as follows:

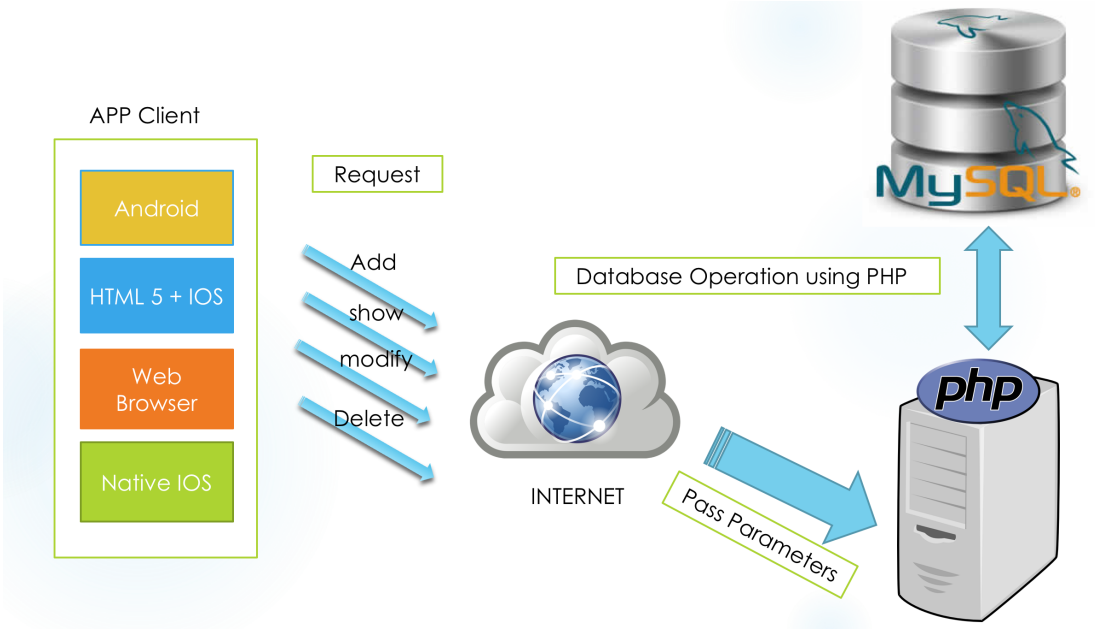

Figure 3-2. The workflow between App Client and PHP Server

The figure below shows process of Objects-linkages and revealing in App client. The native app client mind map revealing is a same process to the before workflow. Next I will explain the detail of multi-platform version app Objects-linkages and revealing.

Firstly, the App Client in Android or IOS will send a request to the HTML5 Web Server using the JavaScript interaction between the mobile devices and the HTML5 Web Server. When the HTML5 Web Server received the request, it will forward the request to the PHP Server to send an AJAX request to the PHP Server. The PHP Server will do the database operation according to the parameters. Then the PHP Server gets the record from database and construct the graph JSON file to the HTML5 Web Server. The HTML5 Web Server will deserialize the graph JSON to a mind map and trigger the callback in the mobile devices to show the mind map in App Client.

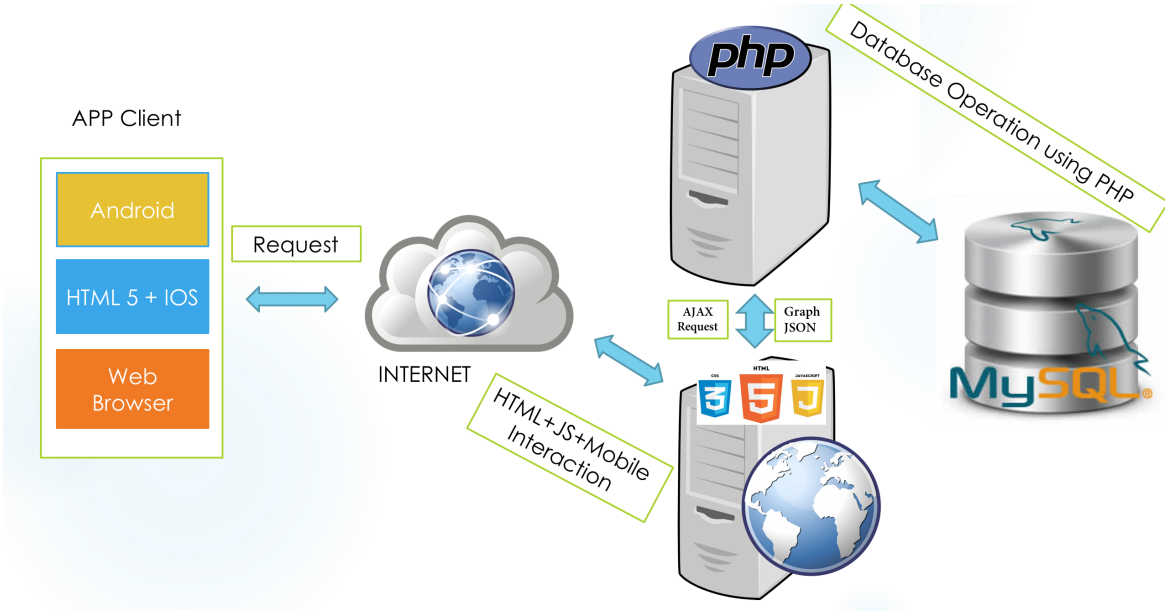

Figure 3-3. The process of mind map revealing in the multi-platform mobile App Client

3.3 Object-oriented Design

3.3.1 Use case Diagram

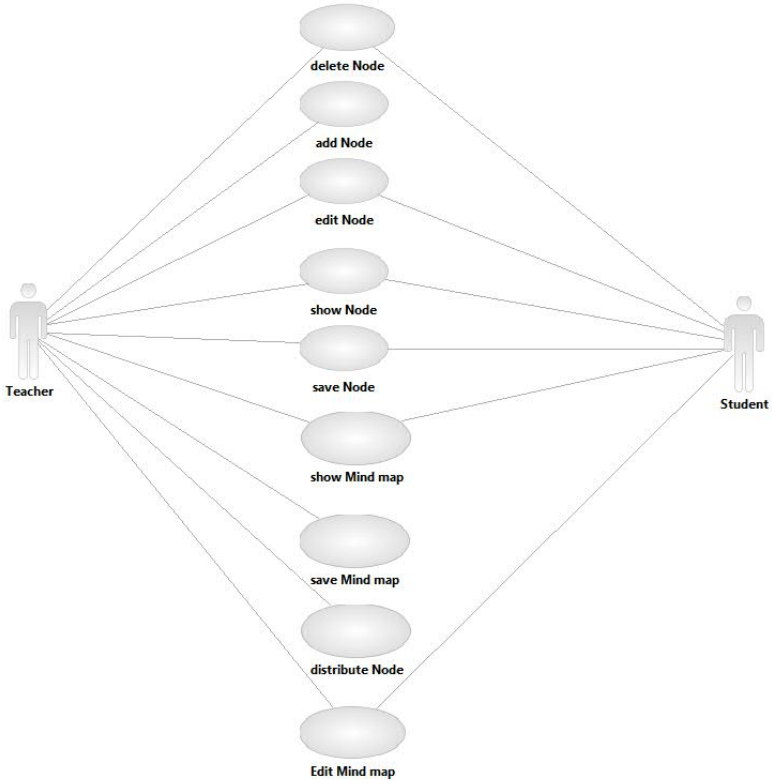

Figure 3-4. Use Case diagram

As we can see from the above use case diagram [9], there are two roles in this system: teacher and student. The teacher can add a node, delete a node, edit a node, show a node, save a node, show mind map, save mind map, edit mind map and distributed the node to the student. The student received the distributed the node from teacher and can operate the node and show a mind map or edit the mind map.

3.3.2 Class Diagram

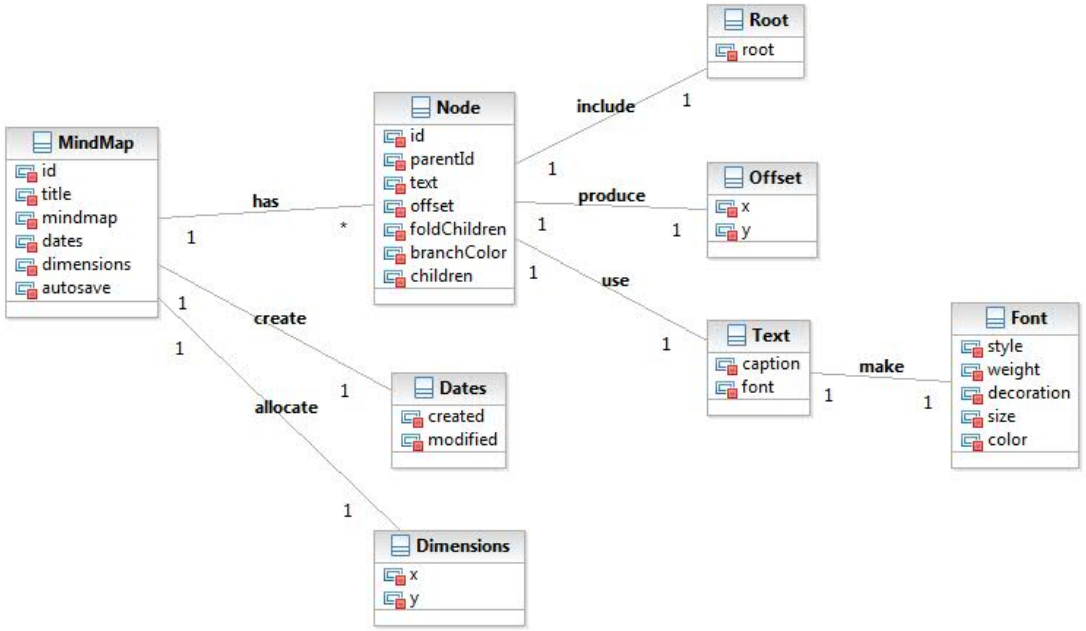

Figure 3-5. Class Diagram

The class diagram [10] shows the relationship between different classes. The "MindMap" class store the dates, dimensions of a mind map and one mind map can have a lot of nodes. The "Node" class store the information of the node offset and the Text information. This diagram records the hierarchy structure between different nodes.

3.3.3 Dynamic Modeling

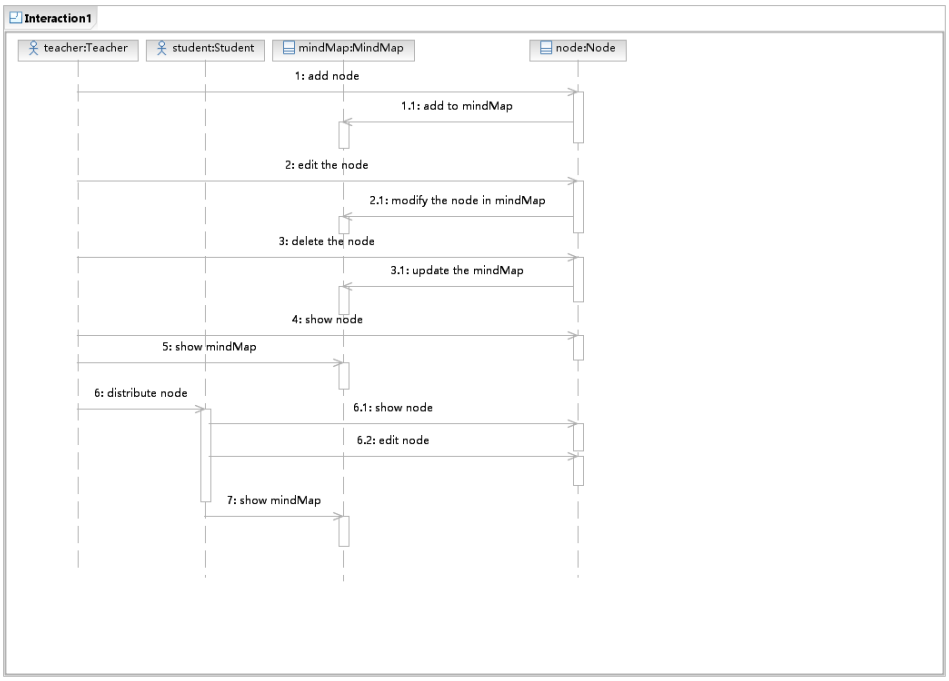

Figure 3-6. Mind map sequence diagram

The sequence diagram [11] shows the interaction between teacher and student in this system.The teacher can add a Node to the mind map and edit the node to add some learning content to the node.The mind map will update the node automatically.The teacher can show a node or delete a node in the mind map. Only teacher ditributes the node to the student, the student can show or edit the node or show a mind map for the ditributed nodes.

3.4 User Interface Design

3.4.1 Basic design

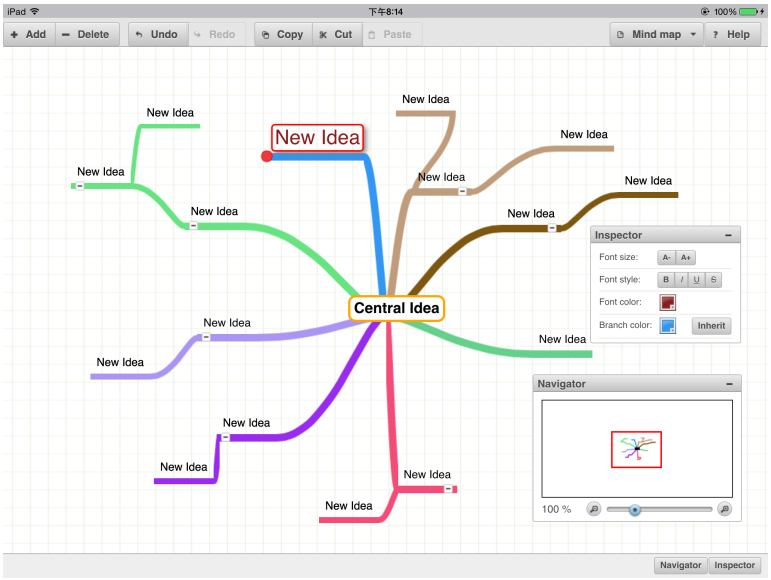

Figure 3-7. Basic function of a mind map

There are many functions in the mind map. For example, the user can add a node, delete a node, undo an action, redo an action, copy part of nodes, cut part of nodes or paste them on some nodes.

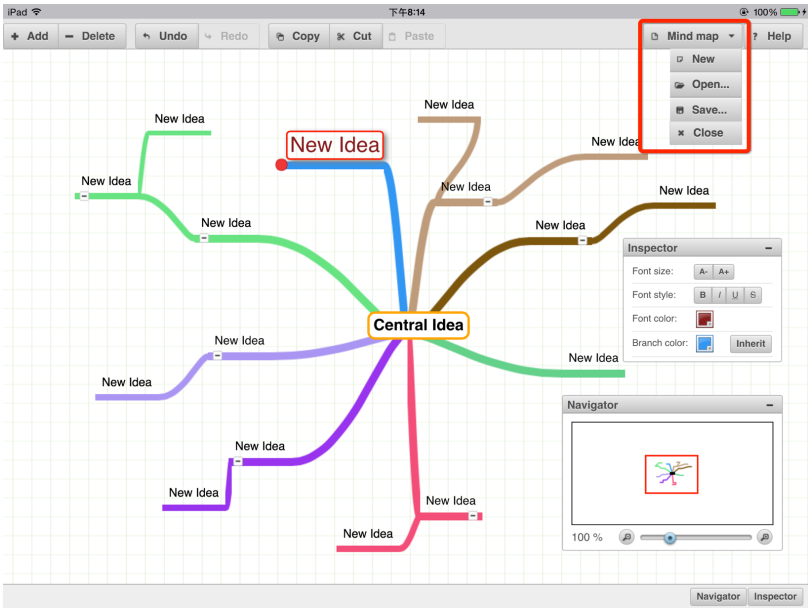

Figure 3-8. Mind map function

We can also save and open a mind map from local storage or cloud platform including Google Drive, Dropbox and Amazon Cloud Drive. We can zoom in or zoom out a mind map using the Navigator item and see an overview of a very big picture. We can change the Text information and branch color using the Inspector part.

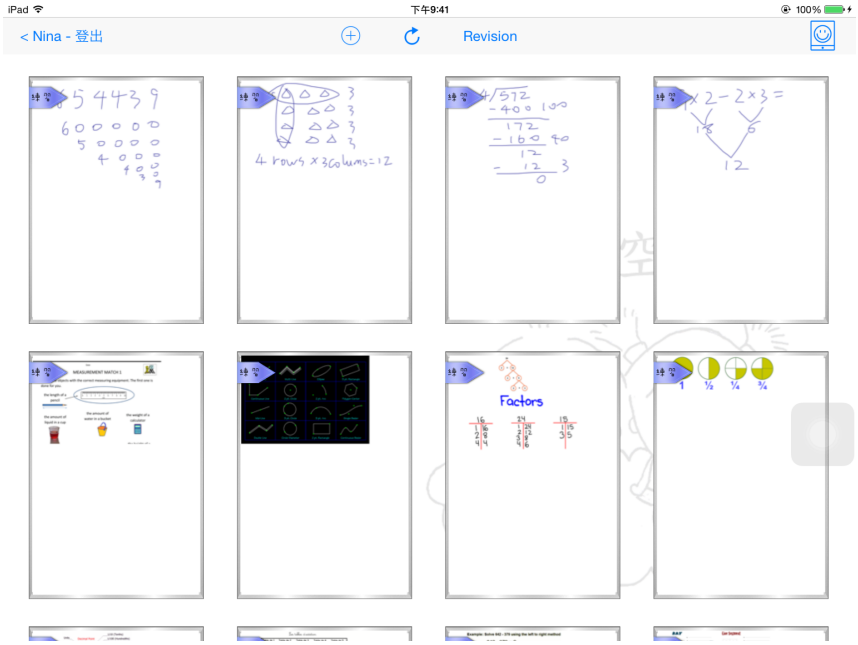

Figure 3-9. Node handling

We add these notes and edit them in the system, then we can see all the notes can show in one mind map. This mind map can show different topics in hierarchy. Different levels have different branch color and text color. We can show the relationship of all the nodes using mind map and we don't need to care the location of all the notes owing to the mind map can record these relationships automatically. Using different branch color to show a clear hierarchical view. As we can see from following mind map, the root node is "Math" and its different level children have the same color with their sibling node and all the nodes are well-distributed.

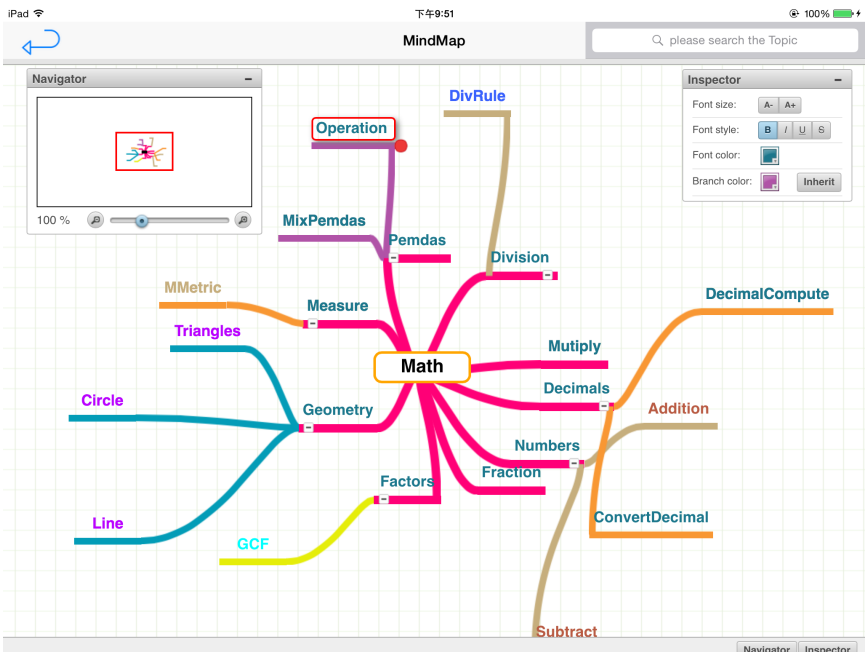

Figure 3-10. Mind map revealing

We can filter a mind map by subject or by other tag such as time. When one mind map is very big, we can search the topic and to show part of the mind map. Furthermore, we can show the mind map dynamically. The system will compute the count and location of nodes and show all the nodes exactly. As we can see from above mind map, there are many topics in one mind map and all the topics are computed and distributed in a very correct location. For example, if we only want to learn the “Geometry” topic and we can search this topic and the system will return all the children nodes of this topic and show them dynamically.

The topic “Geometry” and all its children show as follows:

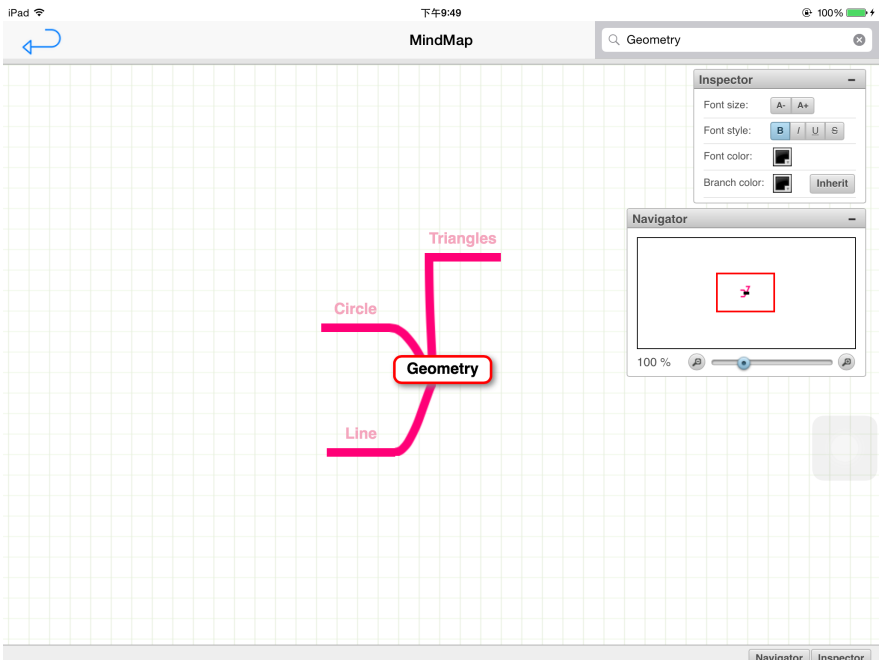

Figure 3-11. Search result of mind map

We can show the note content we add before when we click the nodes in mind map. If we want to review some topic, we can find the topic very quickly using the mind map. We can find the relationship between different topics and we will learn efficiently.

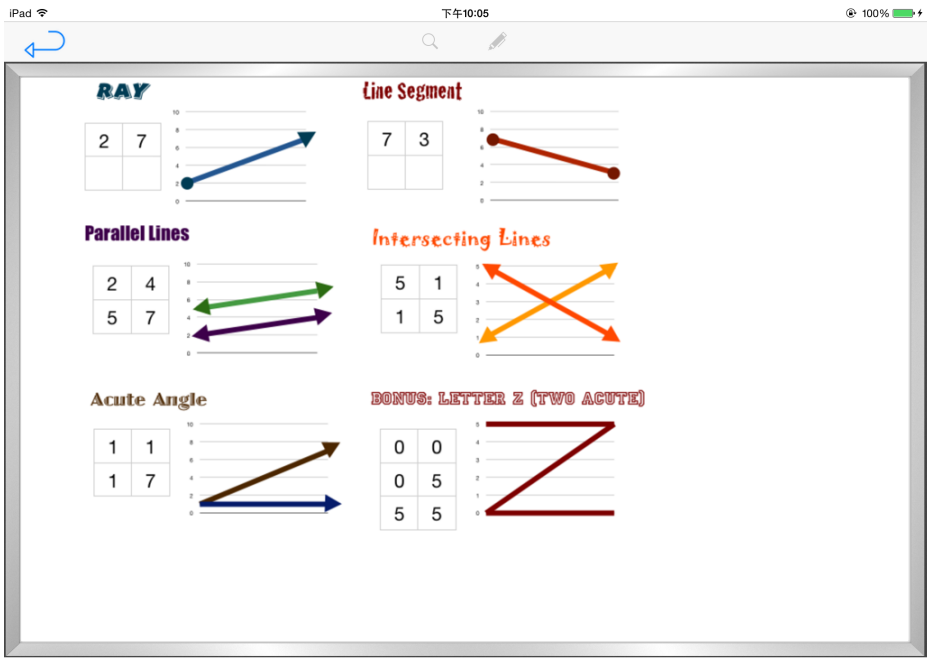

Figure 3-12. Node Content

3.5 Database Design

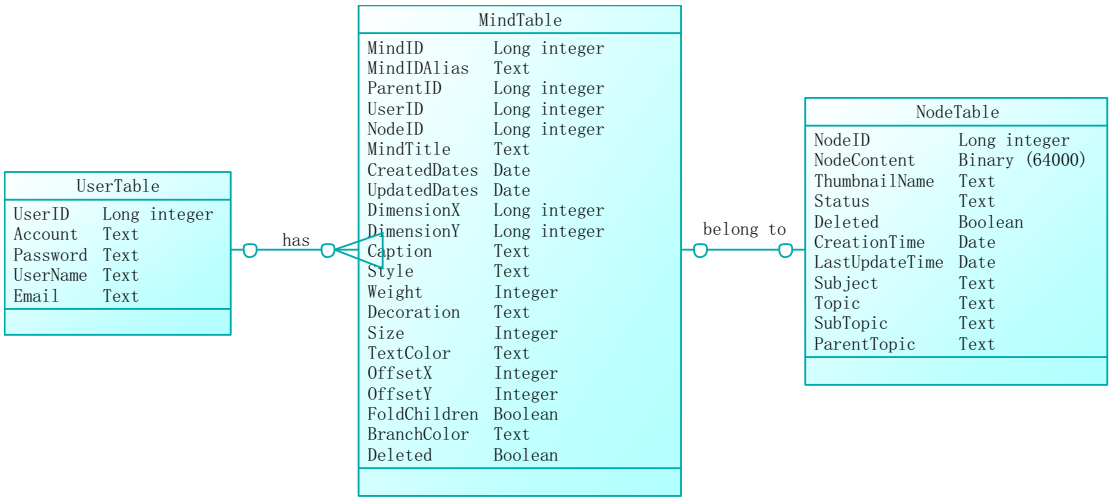

Figure 3-13. The ER diagram of system

The figure 3-13 shows the high level database structure, which consists of three main tables: UserTable, MindTable and NodeTable. The UserTable is used to record user basic information, such as username, password, name and email. According to MindMap data structure, each node information locates in the MindTable, which includes node's parent, text style, color, weight, size, whether deleted or not and so on. The field ParentID in MindTable is a pointer, which points to its parent, thus, the program could build MindMap according to parent-child relation. The remaining table NodeTable stores note information for e-learning system. Because everyone has at least one mindmap, the relation between UserTable and MindTable is one-to-many. It is clearly known that every note belongs to one MindMap node, thus, the relation between MindTable and NodeTable is one-to-one.

4. System Implementation

4.1 Server Implementation

We store all the information of the mind map in MindTable and next I will explain how can we change the database record to a mind map structure.

mindID	mindIDAlias	parentId	userID	nodeID	mindTitle	createdDates	modifiedDates	dimendionsX	dimendionsY	autosave	caption	style	weight	decoration	size	textColor	offsetX	offsetY	foldChildren	branchColor	deleted
9	f3618a50-7e0c-40d4-9ab7-91dcdda5fddf	2	7	4	E-Learning	2015-07-12 11:53:35	2015-07-12 11:53:35	4000	2000	0	Pemdas	normal	bold	none	18	#000000	0	0	0	#000000	0
10	cf0a2ba9-aaa6-4d40-afab-981a31125f4c	2	7	5	E-Learning	2015-07-12 11:54:31	2015-07-12 11:54:31	4000	2000	0	Measure	normal	bold	none	18	#000000	0	0	0	#000000	0
11	579b507f-4e9e-4062-9af7-15e3c5f768ea	2	7	6	E-Learning	2015-07-12 11:56:12	2015-07-12 11:56:12	4000	2000	0	Geometry	normal	bold	none	18	#000000	0	0	0	#000000	0
12	588f59bf-0432-4242-99f7-27ba1446e8ad	2	7	7	E-Learning	2015-07-12 11:58:05	2015-07-12 11:58:05	4000	2000	0	Factors	normal	bold	none	18	#000000	0	0	0	#000000	0
13	61a242d8-0196-4864-8a13-d85fcba2d4ad	2	7	8	E-Learning	2015-07-12 12:00:03	2015-07-12 12:00:03	4000	2000	0	Fraction	normal	bold	none	18	#000000	0	0	0	#000000	0
14	5936430c-8cc3-4bd5-8ea7-94a902921419	2	7	9	E-Learning	2015-07-12 12:01:32	2015-07-12 12:01:32	4000	2000	0	Decimals	normal	bold	none	18	#000000	0	0	0	#000000	0
17	d9881bae-b09b-4440-b44a-ec6e31516819	11	7	13	E-Learning	2015-07-12 12:11:18	2015-07-12 12:11:18	4000	2000	0	Line	normal	bold	none	18	#000000	0	0	0	#000000	0
18	cc44d3a8-0a2d-4a98-917e-c5fd5c2106a4	9	7	14	E-Learning	2015-07-12 12:13:29	2015-07-12 12:13:29	4000	2000	0	Operation	normal	bold	none	18	#000000	0	0	0	#000000	0
19	d1fe5cb1-1463-4923-bd9f-e12252351ba	10	7	15	E-Learning	2015-07-12 12:16:56	2015-07-12 12:16:56	4000	2000	0	MMetric	normal	bold	none	18	#000000	0	0	0	#000000	0
20	5c32c307-120a-4902-be9d-7f3625906042	12	7	16	E-Learning	2015-07-12 12:20:13	2015-07-12 12:20:13	4000	2000	0	GCF	normal	bold	none	18	#000000	0	0	0	#000000	0
21	678415fb-f49a-43d5-80f3-	11	7	17	E-Learning	2015-07-12 12:23:30	2015-07-12 12:23:30	4000	2000	0	Circle	normal	bold	none	18	#000000	0	0	0	#000000	0

Figure 4-1. Database record of mind map

The above figure shows nodes records in MindTable. In order to reduce the database operation, I search all the record of one user and store all the data of one user according the userID. The return result also stores the relationship between parent node and children node. Using parentId attribute, we can get the hierarchical relationship between all the nodes.

The whole process of data structure transformation can be described using pseudo code as follows:

```
result[] = records[userID];
for(int i=0;i<result.length;i++){
    relation.add(parentId->childrenId);
}
Root = records[nodeID];
childrenId[] = relation[Root];
for(int i=0;i<childrenId.length;i++){
    queue.add(childrenId[i])
}
while(queue not null){
    currentMindID = pop(queue);
    childrenId[] = relation[currentMindID];
    queue.add(childrenId);
    Node = result[currentMindID]
    parentNode.add(Node);
}
```

Figure 4-2. Pseudo code of data structure transformation

Firstly, we can search all the nodes from the database using userID and store these nodes to a result array. Secondly, we traverse all the result array to construct a "parentId->chilerenId" relation array using id and parentId in every record. Thirdly, we construct the root node using the root nodeID. We can get all the children of the root node base the "parentId->childrenId" relation array. Then we add all the children of the root node to a queue. Finally, we using a while loop to construct all the children node in every branch and add them to the root node until the queue is null. We will get all the mind map structure after these operations and we will get a parentId and childrenId mapping table.

The parentId->chilerenId mapping table of database record in figure as follows:

Table 4-1. Parent to children mapping table

Key(parent ID)	childrenIdArray
parentId 2	mindID 9 10 11 12 13 14
parentId 11	mindID 17 21
parentId 9	mindID 18
parentId 10	mindID 19
parentId 12	mindID 20

This mapping table can be transformed to the mind map structure as follows:

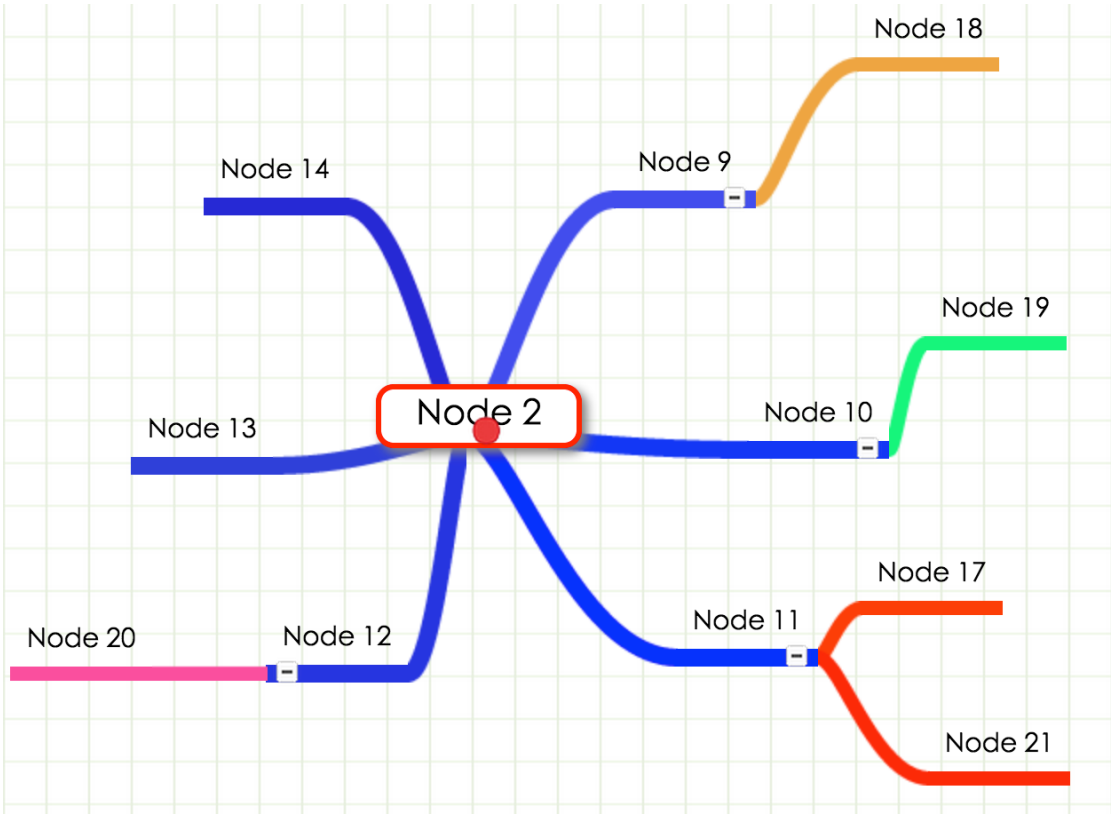

Figure 4-3. Mind map structure

4.2 Web Server Implementation base on HTML5

HTML5 [12] is a fully open, uncontrolled platform and it is widely used in many aspect including web browser and many mobile platforms. Basically HTML5 has it's many new syntactical features, which include the <video>, <audio>, and <canvas> elements, as well as the integration of SVG content. Due to these new elements, it will be very easy to integrate some materials such as multimedia and graphical content to web

without using flash and third party library. When we use the powerful HTML5 platform, we can use one application to run the web browser, IOS platform and Android Platform simultaneously. We just need to develop one version application and it will save a lot of develop costs. Furthermore, we need to upgrade our programs due to the frequent upgrade of mobile platforms.HTML5 can handle these problems.

In this project, we use the <canvas> element to draw a mind map on the canvas view. It is a graphics container to draw the text, paths, boxes, circles and images using JavaScript. We also use the localStorage [13] and sessionStorage [14] to save and open a mind map from local web browser.

4.2.1 Publish/Subscribe Pattern

Publish/Subscribe [15] is a messaging pattern and these two parts generally are separated to avoid direct communication between the publishers and subscribers. The senders always called publishers and the receivers called subscribers. Rather than single objects calling on the methods of other objects, receivers subscribe to a specific task or activity of another object and is notified when it occurs. Publishers notify subscribers when events occur. The process of Pub/Sub pattern is that the subscribers first register to the messaging system and when the publisher sends a message, the messaging system will notify the subscribers to receive the message and to handle the message.

In this project, all the event handled using Publish/Subscribe pattern. For example, if the user wants to open a mind map, the system will set the current document and will publish a “DOCUMENT_OPENED” event to the system. The subscribers will hook up

with the eventbus [16] to listen the global event. When received the "DOCUMENT_OPENED" event, they will execute the program to open a mind map.

4.2.2 Deserialization JSON to show Mind map

Multi-platform program is based on publish/subscribe pattern, which is a message pattern used for communicating between components. There are several advantages of pub/sub pattern. One of the most significant pros is that the publishers are loosely coupled to subscribes, thus, the whole program has much more scalability because it is easy for us to add more subscribes. The whole sequence diagram of deserialization JSON and showing Mind map showed in following figure.

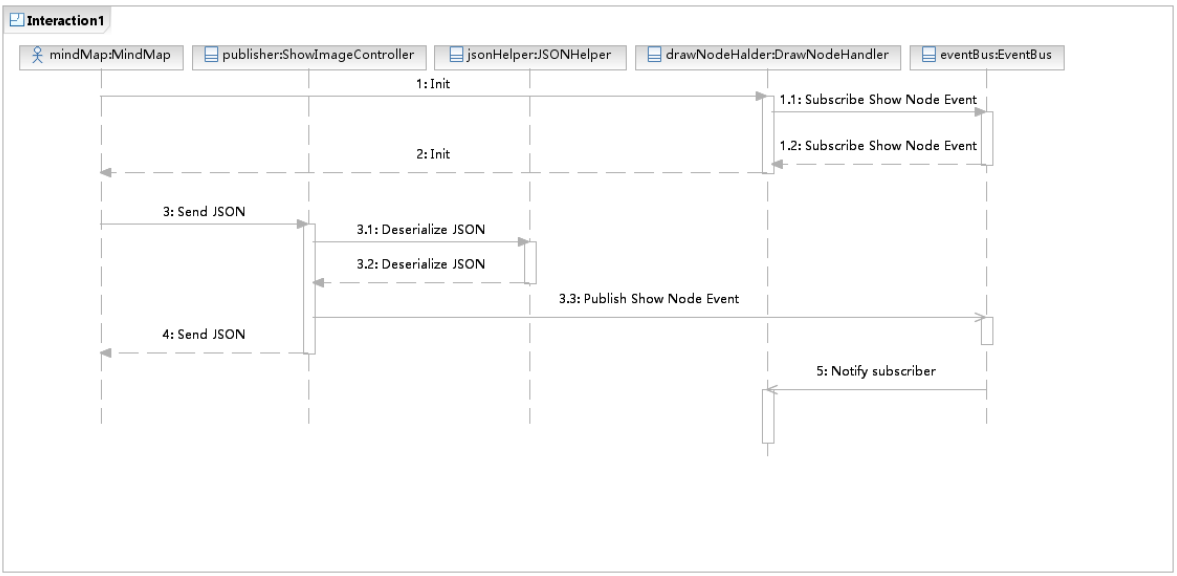

Figure 4-4. The sequence diagram of deserialization JSON and showing MindMap

As we can be seen from the Figure 4-4, the whole process consists of three steps. First of all, Mind map calls init functions to subscribe show node event in the eventbus. When Mind map get JSON file form web server, it calls the send JSON function to draw image, which will call synchronous deserialization JSON function and

asynchronous publish show node event function respectively. Finally, the eventbus will notify draw node handler to draw mind map according to deserialization JSON object.

4.2.3 Local Storage

With the HTML5 new features, we can store more data using localStorage locally. Before HTML5, we can use cookies to store the application data and the more we visit the web, the more cookies we accumulate. The browser may start and run slower because of many cookie files. The cookie only stores a small piece of text file and it will send it to the server when we visit a website again. This will lead to some secure problems if hackers want to intercept our privacy cookie file to do some malicious operation like steal the payment cookie. Unlike cookie, the storage limit is far larger (at least 5MB) and the data never transferred to the server. It is more secure and it will not affect the application performance.

The HTML Local Storage provides two objects for the storage on the client. The localStorage and sessionStorage. The localStorage can store data permanent and sessionStorage can only store data in one session. When the browser tab is closed, the data will be deleted.

The localStorage and sessionStorage using in the mind map as follows:

```
mindmaps.LocalStorage = (function() {
  return {
    put : function(key, value) {
      localStorage.setItem(key, value);
    },
    get : function(key) {
      return localStorage.getItem(key);
    },
    clear : function() {
      localStorage.clear();
    }
  };
})();
```

```
mindmaps.SessionStorage = (function() {
  return {
    put : function(key, value) {
      sessionStorage.setItem(key, value);
    },
    get : function(key) {
      return sessionStorage.getItem(key);
    },
    clear : function() {
      sessionStorage.clear();
    }
  };
})();
```

Figure 4-5. Storage in mind map

For example, if I want to save a document to the localstorage. Overwrites the old document if one with the same id exists. The program as follows:

```
saveDocument : function(doc) {
    try {
      localStorage.setItem(prefix + doc.id, doc.serialize());
      return true;
    } catch (error) {
      // QUOTA_EXCEEDED
      console.error("Error while saving document to local storage",
          error);
      return false;
    }
  }
```

Figure 4-6. localStorage in mind map

Loads a document from the local storage.

```
var getDocumentByKey = function(key) {
    var json = localStorage.getItem(key);
    if (json === null) {
      return null;
    }
loadDocument : function(docId) {
      return getDocumentByKey(prefix + docId);
    }
```

Figure 4-7. Load document using localStorage

Finds all documents in the local storage object

```
getDocuments : function() {
    var documents = [];
    // search localstorage for saved documents
    for ( var i = 0, max = localStorage.length; i < max; i++) {
      var key = localStorage.key(i);
      // value is a document if key confirms to prefix
      if (key.indexOf(prefix) == 0) {
        var doc = getDocumentByKey(key);
        if (doc) {
          documents.push(doc);
        }
      }
    }
    return documents;
  }
```

Figure 4-8. Find all documents using localStorage

Deletes a document from the local storage.

```
deleteDocument : function(doc) {
    localStorage.removeItem(prefix + doc.id);
},
```

Figure 4-9. Delete the document using localStorage

4.2.4 Cloud Storage

In order to store the mind map file to the cloud such as Google Drive, Dropbox and Amazon Cloud Drive, I integrated the FilePicker API [17] to the system. With FilePicker, the system can get access to these cloud platforms without authentication.

Filepicker.io is a tool that allows users to access and upload their files to the cloud both from their local disk as well as online platform. We can integrate the API to develop an application including these functions. he API uses JavaScript protocol. The JavaScript API gives developers the ability to choosing a file and save a file to the different cloud platforms including Dropbox, Google drive.

If we want to integrate Filepicker files into our application, the first thing to do is include the library and apply for a key for the integration.

```
<script src="//api.filepicker.io/v0/filepicker.js"></script>
```

We define the service as follows:

```
services: [
        filepicker.SERVICES.GOOGLE_DRIVE,
        filepicker.SERVICES.DROPBOX,
        filepicker.SERVICES.BOX,
        filepicker.SERVICES.URL
    ]
```

Figure 4-10. filepicker service

Using the JavaScript API for save files, we simply pass options and a URL to save a file to the cloud. Shows the save dialog where the user can save the current mind map. Skips the dialog and saves directly when options. saveAs = true is passed and a cloud storage file is currently open.

Here's how it works:

```
  this.save = function(options) {
    options = options || {};

    if (!filepicker || !navigator.onLine) {
      options.error && options.error("Cannot access cloud, it appears you are offline.");
      return;
    }

    var doc = mindmapModel.getDocument();
    var data = doc.prepareSave().serialize()

    var success = function(url) {
      console.log('saved to:', url);
      eventBus.publish(mindmaps.Event.DOCUMENT_SAVED, doc);

      if (options.success) {
        options.success();
      }
    };

    // save dialog
    filepicker.getUrlFromData(data, function(dataUrl) {
      filepicker.saveAs(dataUrl, mimetype, saveOptions, success);
    });
  }
}
```

Figure 4-11. save file to cloud using filepicker

Using the FilePicker API to open a mind map from cloud, we pass the options and URL to get the data stored in cloud and we parse the JSON file to the mind map object. Show the mind map object in the mind map model view.

```
this.open = function(options) {
  options = options || {};

  if (!filepicker || !navigator.onLine) {
    options.error && options.error("Cannot access cloud, it appears you are offline.");
    return;
  }

  filepicker.getFile(mimetype, openOptions, function(url, data) {
    // load callback
    options.load && options.load();
    // load mindmap
    $.ajax({
      url: url,
      success: function(data) {
        try {
          // convert to object first if response is a string
          if (Object.prototype.toString.call(data) == '[object String]') {
            data = JSON.parse(data);
          }
          var doc = mindmaps.Document.fromObject(data);
        } catch (e) {
          eventBus.publish(mindmaps.Event.NOTIFICATION_ERROR, 'File is not a valid mind map!');
          throw new Error('Error while parsing map from cloud', e);
        }
        mindmapModel.setDocument(doc);
        // execute callback
        if (options.success) {
          options.success(doc);
        }
      },
      error: function(jqXHR, textStatus, errorThrown) {
        if (options.error) {
          options.error("Error: Could not open mind map!");
        }
        throw new Error('Error while loading map from filepicker. ' + textStatus + ' ' + errorThrown);
      }
    });
  });
};
```

Figure 4-12. Open a document using filepicker

4.2.5 Universally Unique Identifier

In order to guarantee the uniqueness of the nodeID in mind map, we use the Universally Unique Identifier(UUID) [18] to represent the node in mind map. A UUID is 128 bits long, and can guarantee uniqueness across space and time.

The UUID definition as follows:

```
UUID                   = time-low "-" time-mid "-"
                         time-high-and-version "-"
                         clock-seq-and-reserved
                         clock-seq-low "-" node
time-low               = 4hexOctet
time-mid               = 2hexOctet
time-high-and-version  = 2hexOctet
clock-seq-and-reserved = hexOctet
clock-seq-low          = hexOctet
node                   = 6hexOctet
hexOctet               = hexDigit hexDigit
hexDigit =
      "0" / "1" / "2" / "3" / "4" / "5" / "6" / "7" / "8" / "9" /
      "a" / "b" / "c" / "d" / "e" / "f" /
      "A" / "B" / "C" / "D" / "E" / "F"
```

Figure 4-13. UUID definition

We use it in the mind map system and show the implementation as follows:

```
mindmaps.Util.createUUID = function() {
  // http://www.ietf.org/rfc/rfc4122.txt
  return 'xxxxxxxx-xxxx-4xxx-yxxx-xxxxxxxxxxxx'.replace(/[xy]/g, function(c) {
    var r = Math.random() * 16 | 0, v = c == 'x' ? r : (r & 0x3 | 0x8);
    return v.toString(16);
  });
};
```

Figure 4-14. Generate UUID in mind map

4.2.6 Node dynamic relocation

In order to dynamic creating the Mind map based on children, one of the most significant problem is that how to dynamic compute the children node position. In program, when we add some children node, the whole figure should be redraw to fitness with the new nodes.

The algorithm of dynamic node position in Cartesian coordinates consists of three steps: Firstly, assign the initial position to parent node. Then, compute degree based on the number of children + 1. Finally, according to the following formula, to calculate the location of each node.

$$X_{child} = R \times \cos(a) + X_{parent}$$
$$Y_{child} = R \times \sin(a) + Y_{parent}$$

In formula, X_{parent} and Y_{parent} is a parent position. R is radius of circle. X_{child} and Y_{child} represents the child position.

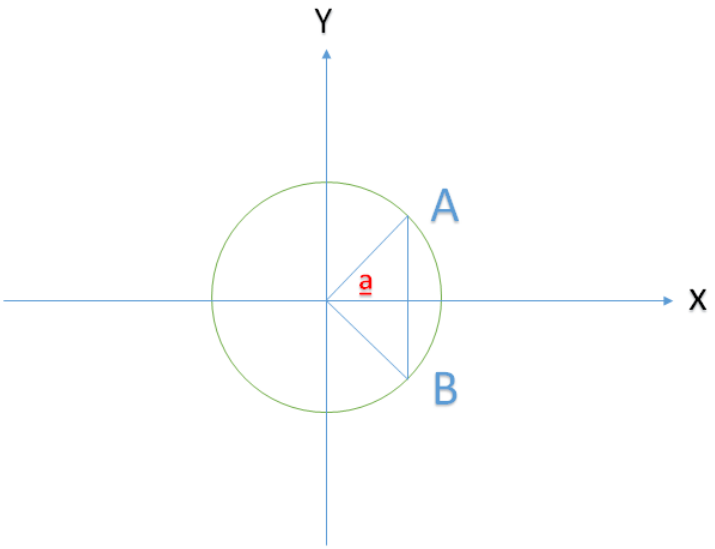

Figure 4-15. The example of computing children location

For instance, assume there are three nodes in the Mind map. Assume we assign (0, 0) to root node O and the R is 1. Because the root node has two children A and B respectively. The half-circle is divided into three equivalent sections, thus, the degree a is $180^{\circ}/(2+1) = 60^{\circ}$. The X position of A is $\sin(60) * \mathrm{R} + 0 = 0.866$, and the Y position of A is $\cos(60) * \mathrm{R} + 0 = 0.5$. When we add two children node A1, A2 to A, the X position of A1 is $\sin(60) * \mathrm{R} + 0.866 = 1.732$. The Y position of A1 is $\cos(60) * \mathrm{R} + 0.5 = 1.0$.

4.3 Mobile App Implementation

We have implement the two version applications on the mobile platforms. The first one is native IOS version and it can run on the IOS platforms and the second one is the multi-platform version and it can run on most mobile platform including IOS, Android and BlackBerry. In this part, I will explain the mind map node implementation, the database implementation, the native IOS implementation and the multi-platform

implementation.

4.3.1 Node Implementation

The node stores all the information we need in the mind map. Beside the basic information like id, title, dimensions, we also need to record the relationship of different nodes and store the hierarchical relationship between parent node and children node.

The following figure shows the node structure of the mind map.

Figure 4-16. Node structure of mind map

4.3.2 Database Implementation

The following figure shows the MindTable structure in database.

浏览 结构 SQL 搜索 插入 导出 Import 操作 清空 删除

	字段	类型	整理	属性	Null	默认	额外
	mindID	int(11)			否		auto_increment
	mindIDAlias	varchar(100)	utf8_general_ci		否		
	parentId	int(11)			否		
	userID	int(11)			否		
	nodeID	int(11)			否		
	mindTitle	varchar(50)	utf8_general_ci		否		
	createdDates	datetime			否		
	modifiedDates	datetime			否		
	dimendionsX	int(11)			否	4000	
	dimendionsY	int(11)			否	2000	
	autosave	varchar(50)	utf8_general_ci		否		
	caption	varchar(50)	utf8_general_ci		否		
	style	varchar(50)	utf8_general_ci		否	normal	
	weight	varchar(50)	utf8_general_ci		否	normal	
	decoration	varchar(50)	utf8_general_ci		否	none	
	size	int(11)			否	20	
	textColor	varchar(50)	utf8_general_ci		否	#000000	
	offsetX	int(11)			否		
	offsetY	int(11)			否		
	foldChildren	varchar(50)	utf8_general_ci		否	false	
	branchColor	varchar(50)	utf8_general_ci		否	#000000	
	deleted	tinyint(1)			否	0	

Figure 4-17. Database structure of MindTable

Next I will use an example to explain the relationship between the mind map, the JSON structure file and the database structure.

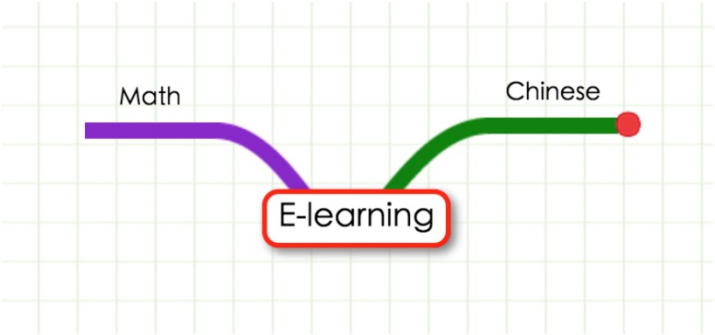

Figure 4-18. Mind map example

The mind map example corresponding JSON file showing as follows:

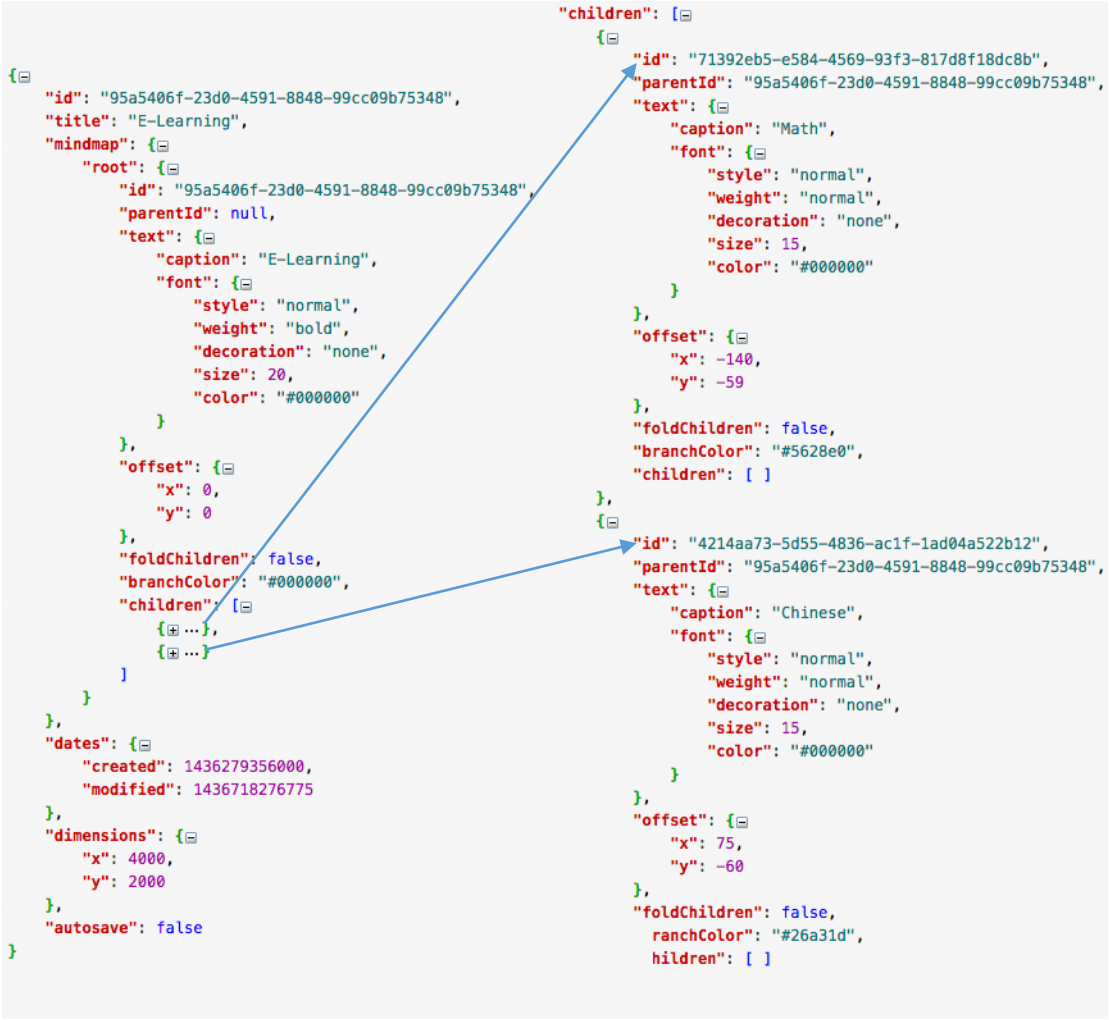

Figure 4-19. JSON file of the mind map example

As we can see from the mind map example, we have the root node “E-learning” and two children node “Math” and “Chinese”. In the JSON file, we store the root node and children node and add the children node to the corresponding parent node. We store the text information such as the style, weight, text color and size. We also record the location of nodes using offset attribute. In the database structure, we can see the the MindIDAlias is a UUID and it is the id in JSON file. The parentId record the id of the parent node. The text data and the location of the node are all stored in one record of the database.

4.3.3 Native IOS implementation

In the mobile platform, we implement the two version application: the native IOS version and the multi-platform version. The following figure shows a overview of the native IOS version. We can add a node, delete a node and edit a node. We can change the node color to represent different branches.

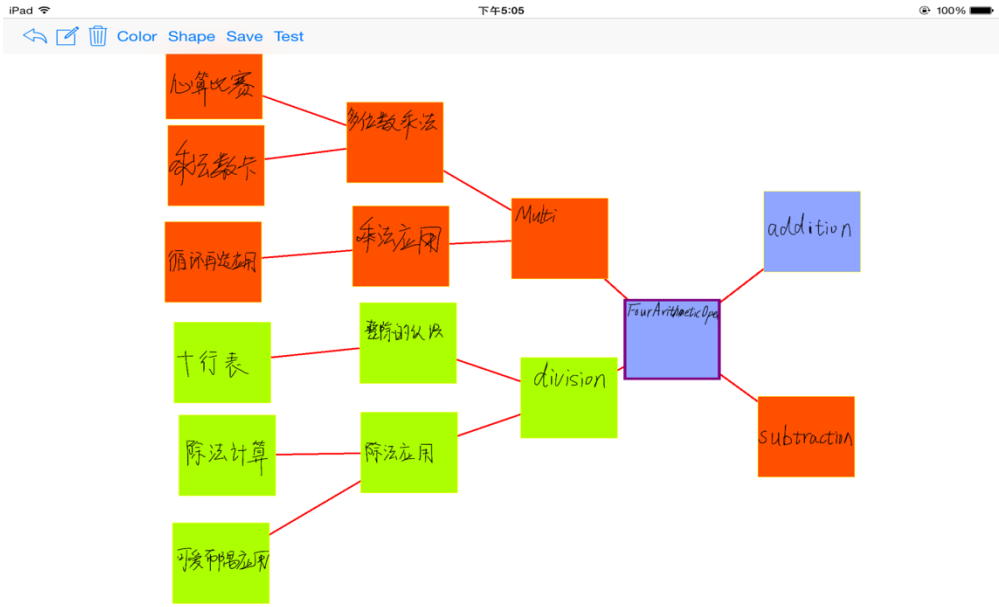

Figure 4-20. Native IOS Implementation

4.3.4 Multi-platform implementation

Android platform

We use the webview component [19] run the web server on the Android platform. We can set many JavaScript configurations for the webview. We set the JavaScript Enabled using the settings. We can control the plugin states, open the viewport, set the zoom, open the localstorage and app cache and set the off-line file access which also configured in the HTML5 manifest.

We set the WebChromeClient and WebViewClient for the webview component to trigger the callback function and complete the communication between the mobile devices and web server.

There are many callback functions such as onJsPrompt, onJsAlert, onJsConfirm, onExceededDatabaseQuota and onReachedMaxAppCacheSize in the WebChromClient class. We use these callback functions to customize the style of these dialogs, enlarge the database and the cache size.

We use the WebViewClient callback functions to listen the progress of the web server loading. We can listen the web request using shouldOverrideUrlLoading callback function and we can also listen current pages' progress to complete some operation during the loading of websites.

We can add the JavascriptInterface and write the callback for the HTML5 server for the webview and pass the object name, then in HTML5,we can use this object to call the function pre-defined in the JavascriptInterface. For example, we want to press the node to call the android local function to show the node content. We can write as follows:

```
mWebView.addJavascriptInterface(JavaScriptInterface,"Android");

public class JavaScriptInterface  {

// Because secuity problem,we must use annotation after android 4.0
        @JavascriptInterface
        public int callbackFun() {

        }
}
```

Figure 4-21. JavaScirptInterface

In order to avoid malicious callback function, we must use the annotation "@JavascriptInterface" after Android 4.0.

In the HTML5 file, we can use the "Android" object to call the callbackFun() function to complete the corresponding operation. The following figure shows the overview of the HTML5 web server running on android platform.

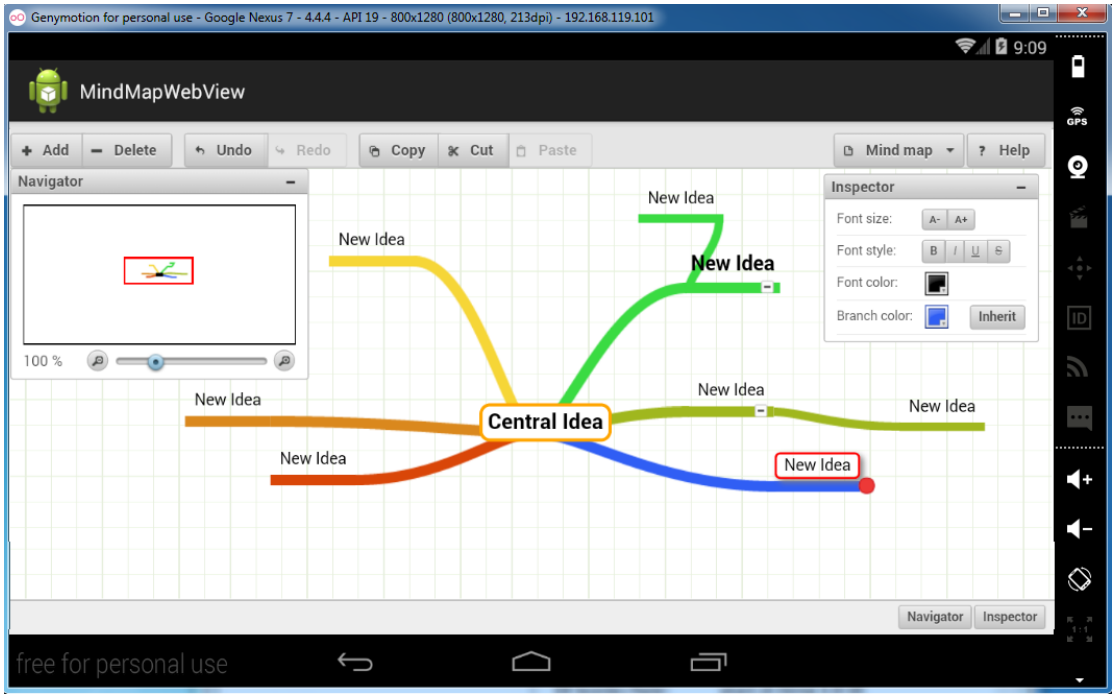

Figure 4-22. Multi-platform running on android platform

IOS platform

We use UIWebView component [20] on the IOS platform to integrate the HTML5 web server. If we want to use the webview, we first need to set the UIWebView delegate for the UIWebView. The UIWebViewDelegate protocol defines methods that delegate of a UIWebView object can listen the operation when web content is loaded.

We can use the webView:shouldStartLoadWithRequest, webViewDidStartLoad, webViewDidFinishLoad and webView:didFailLoadWithError these four callback function to listen the operation of the web content. For example, we can add some JavaScript file to the HTML in the webViewDidFinishLoad function, then the web content will be modified after the web finishing loading.

Comparing with Android, there are several differences on IOS platform. It do not provides the direct interface for the HTML to call the local function in IOS platform. Therefore, we need to define our own protocol to implement the HTML file call IOS local function. For example, we want to press the HTML5 node to call the native IOS function to show the node content. We need to use the shouldStartLoadWithRequest callback function to listen all the request in the HTML5.In the HTML5 file we define our own protocol. The example code as follows:

```
view.nodeMouseDown = function(node) {
    mindmapModel.selectNode(node);
    creator.attachToNode(node);
    var id = node.id;
    var postData = new FormData();
        postData.append("mindIDAlias", id);

            $.ajax(
                    {
                        url : "http://i.cs.hku.hk/~yyzhang2/tk/QueryIDUseUUID.php",
                        type: "POST",
                        data : postData,
                        mimeType:"multipart/form-data",
                        contentType: false,
                        cache: false,
                        processData:false,
                        async : true,
                        success:function(data, textStatus, jqXHR)
                        {

                            window.location.href = 'mindmap://'+data;

                        },
                        error: function(jqXHR, textStatus, errorThrown)
                        {

                        }
                    });


};
```

Figure 4-23. Define a protocol

When the user presses the node, the HTML5 file will send a “mindmap://” protocol with the parameter and we can listen the request in the shouldStartLoadWithRequest callback function, then we can call the IOS local function to complete the operation. The IOS example code as follows:

```objc
- (BOOL)webView:(UIWebView *)webView shouldStartLoadWithRequest:(NSURLRequest *)request navigationType:(UIWebViewNavigationType)navigationType
{

    NSString *url = request.URL.absoluteString;
    NSRange range = [url rangeOfString:@"mindmap://"];
    NSUInteger loc = range.location;
    if (loc != NSNotFound) {

        NSString *para = [url substringFromIndex:loc + range.length];
        [self localFun:para];
    }
    return YES;
}
```

Figure 4-24. IOS example code

The following figure shows the overview of the HTML5 web server running on IOS platform.

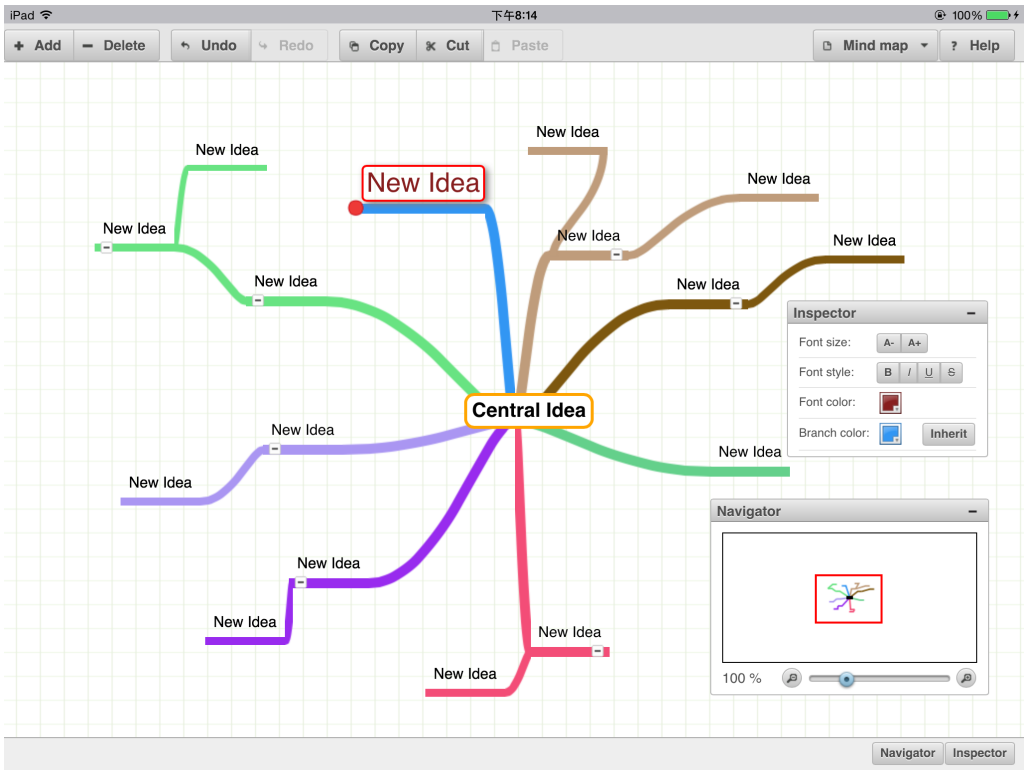

Figure 4-25. Multi-platform running on IOS platform

5. Limitations and Future Work

In this project, I have done the mind map revealing along with some functions using in the e-learning. There are still some limitations of this project. Besides filter the topic, we can filter some more dimensions such as time and some special tag. For example, we can only focus on some topics we learned in last month or some “wrong” questions we learned before. Furthermore, we can add some tag for every node in mind map and then we can find the association link between different topics. For example, we can add a tag for the addition and multiply, when students learn the multiply, they will find multiply is transformed from addition. With association link, students will learn more deeply and effectively.

Currently, we can change all the notes we add in the system to a mind map view and we can do some actions on the mind map. In the future, we can build 3D view for all the notes and give students a clear mind about what they have learned. We can visualize students’ multiple learning sequence using 3D view.

Consider the user-friendly interface, we need to pay more attention to the design of a fitting UI to the students in primary school and put more interesting images or multi-media into the system.

6. Difficulties Encountered

During this project, we encountered many difficulties about the mind map revealing, the cross-platforms and the communications between the mobile application and server. We learned a lot from the difficulties and next I will explain all the difficulties in detail.

6.1.1 JavaScript Interaction with IOS

Not Like Android platform, the IOS platform don't provide the direct method using HTML5 and JavaScript to call the local function in IOS. In this project we need to add a trigger to the node in mind map that if students press the node, the mind map can show the corresponding node content using nodeID. In this activity, the HTML5 need to call the IOS local function to show the node content we add before.

I define a private protocol to handle this problem. In normal web activities, we use the http protocol to transfer the data on the Internet. So I simulate this operation, I send a request and use the IOS UIWebView callback function to listen the request. Actually, I don't want to send a request to the Internet and just want to build a connection between the HTML5 web server and IOS application. Using this connection, the HTML5 can pass the data or action to the local IOS and then IOS can parser the data and action to finish the corresponding operation.

6.1.2 Mobile Event Handle

Using the HTML5 and JavaScript on the mobile devices, we need to handle all the event that is not appear in the web browser like the touch event. Beside the basic mouse

event including mouse down, mouse up, mouse wheeled, mouse over and double click. We also need to handle the touch event which is the most popular action on the mobile devices. First, I will check the touch support, if the devices support the touch event, we can detect it. At the same time, we ignore this event if it is a browser. Initialize the simulated mouse event using the touch event ‘s coordinates including touch.screenX, touch.screenY, touch.clientX and touch.clientY. And then dispatch the simulated event to the target element. We also record the touch start, touch move and touch end and set these handlers on the element to delegate the handlers to them. Then we can use the touch event on the mobile devices.

6.1.3 Handle Different Data Structure

There are many data structures in the e-learning systems. If we show the data in hierarchical classification, we must store the parent to children mapping of every element in the mind map. But when we add the node, we don’t need to store the data in this structure and we must change these two data structure and show them consistently. On the other hand, we also need to handle the transformation from database record to the mind map data structure.

6.1.4 Show Mind Map Dynamically

We can use some topics to filter the mind map and the mind map must show dynamically. We compute location and offset of every node and distributed them well on the canvas view. We traverse all the branches of the mind map and get the node

count of different levels and give them different offset and different colors. Using different colors, students can have a clear mind of all the mind map.

6.1.5 Cross-domain Ajax request

We have two server programs (the server based on PHP and the HTML5 web server) deployed on different domains. When we send a post request from one server to another, it will be blocked. We check the programs with some references and find that the default value for the request is same-domain access owing to the security problems [21]. we need to modify the configuration on crossDomain and force change the value to cross-domain request.

7. Conclusions

With the very use of new technology in school, e-learning will play a more and more important role in education system. This project aims to develop efficient software using in the e-learning on the mobile platform including IOS and Android. In this project, we have designed and implemented object-based file revealing instead of traditional file-based storage. Using the mind map, we can store all the students' notes or assignments in one mind map and show a big picture to the students. With this big picture, we can get an overview about all the knowledge we have learned. If students use the software from primary school to secondary school, they will store a very big data into the system. It is not easy for students to find the assignment, notes or questions we have learned. Therefore, in the program, we implement efficient search using the topic we build. If we just want to focus on the topic like "multiply", we can use this topic to search in mind map, then the system will show the multiply and all its children topic. In the sub-mind map, we can see some content more clearly. In the next phase, we will add more dimensions searching on the mind map. We can search some tag like "always wrong" questions or "undo questions".

During this project, we encounter many challenges including the server, the client and some cross-platform problems. Cross-platform has a double effect on the software development. On one hand, we just need to develop one version application and then it can run anywhere. It will reduce the high development costs and reduce the problems in multiple versions of these mobile platforms. On the other hand, because the limited

interface for the HTML5 and web development in mobile platforms, we need to handle every detailed part about the mobile devices like the event handle, the native function calling and cannot handle the functions related to the hardware.

8. References

[1] Wikipedia. Educational technology.
https://en.wikipedia.org/wiki/Educational_technology
[2] Zdziarski J. Hacking and securing iOS applications: stealing data, hijacking software, and how to prevent it[M]. " O'Reilly Media, Inc.", 2012.
[3] PhoneGap. Developer Portal. http://phonegap.com/developer/
[4] Sencha. Sencha Touch. https://www.sencha.com/products/touch/
[5] Wikipedia. Mind map. https://en.wikipedia.org/wiki/Mind_map
[6] Mesnier M, Ganger G R, Riedel E. Object-based storage[J]. Communications Magazine, IEEE, 2003, 41(8): 84-90.
[7] Donnelly D D, Donnelly N J. Records management apparatus for constructing and maintaining a file folder-based document filing and retrieval system: U.S. Patent 4,973,086[P]. 1990-11-27.
[8] E-learning. Math-learning. http://www.onlinemathlearning.com/
[9] Microsoft. UML Use Case Diagram. https://msdn.microsoft.com/en-us/library/dd409432.aspx
[10] Microsoft. UML Class Diagram. https://msdn.microsoft.com/en-us/library/dd409437.aspx
[11] Microsoft. Sequence Diagram. https://msdn.microsoft.com/en-us/library/dd409377.aspx
[12] W3C. HTML5. http://www.w3.org/TR/html5
[13] w3school. localStorage.
http://www.w3schools.com/html/html5_webstorage.asp
[14] w3school. seesionStorage.
http://www.w3schools.com/html/html5_webstorage.asp
[15] Wikipedia. Publish–subscribe pattern.
https://en.wikipedia.org/wiki/Publish–subscribe_pattern
[16] guava-libraries. EventBus. https://code.google.com/p/guava-libraries/wiki/EventBusExplained
[17] FilePicker. FilePicker Documentation.
https://www.filepicker.com/documentation/
[18] Network Working Group. UUID.http://www.ietf.org/rfc/rfc4122.txt
[19] Android API. WebView.
http://developer.android.com/reference/android/webkit/WebView.html
[20] IOS Developer Library. UIWebView.
https://developer.apple.com/library/ios/documentation/UIKit/Reference/UIWebView_Class/
[21] Microsoft. AJAX Cross Domain Request. https://msdn.microsoft.com/en-us/library/dd573303(v=vs.85).aspx