Final Year Project

Systems Development and Administration (ITP4913M)

Final Report

Group members:

Chan Po Chi ()
Lam Wai Keung (130754019)
Cheng Kwan Yin (130089549)
HuiKamLun (130788533)

Contents

Abstract

In this project, we are going to develop an interactive eLesson module on the module supporting multiple platforms like iOS and Android. The project will be divided into four different parts: Extracting the PowerPoint data, server side programmingwith plugins in the Moodle platform and iOS and Android client.

First, we will have a belief description of the objective and aims of this project. After that, we will focus on how we organize the project and parallel working on the project. Then, we will move on to the problems we encounter with the proposed solutions. After that, we will jump to the methodology and requirements. Finally, it is our project schedule.

Acknowledgement

The Final Year Project is coming to the end, without the support by the people who continuously give us advices and recommendations, our project could never proceed so smoothly. Therefore we want to express our gratitude to the following person:

Firstly, we would like to thanks Mr. Alan Po sincerely, who is our final year project supervisor. In the whole project, he always give us the biggest help such as guiding what each of us should do to get our project started at the beginning stage, give us instruction and comment to make our system and report become much more mature and perfect, tell us what is our most important task so that most of our system functionalities are remained, all of these advices are so important and helpful to uswhich made our project couldeventuallybe developed successfully.

Secondly, we would like to thanks Mr. James Lo, who is our consultant recommended by Alan, he is always very nice and willing to help us to solve any kind of obstacles like setting up the moodle, FTP tools, and he usually shares some useful materials or APIs that let our project becomes easier.

Also, we want to thanks Mr. Akira Chan, who is our second consultantrecommended by Alan, he is proficient with moodle system and architecture, he could give us a clear instruction on how to achieve our target, and give comment on our report so that we can make it better.

Finally, we want to thanks all the lecturers that taught us before. They gave us a strong foundation to handle different kinds of technical problems and also this project.

Introduction

The traditional class is unidirectional, boring and mostly focusing on lots of theory. Lecturers are teaching by their prepared material or PowerPoint, however, there are no interaction between lecturer and student. Students are reflected relatively passive to be participated into the class or asking question. To make the lecture more fun and interactive, we come up with this project.

In this project, we are going to develop an interactive eLesson module on the module supporting multiple platforms like iOS and Android. The project will be divided into four different parts: Extracting the PowerPoint data, server side programming with plugins in the Moodle platform and iOS and Android client.

In the PowerPoint part, teacher can upload a PowerPoint to the Moodle in the meantime, the PowerPoint will be converted into web based automatically. The PowerPoint can be viewed in the Moodle website and also mobile device. This requiresextracting the data from PowerPoint and building it by PHP to generate a dynamic webpage.

In the client side, the app will make some request to the Moodle server and receive data in order to access the lesson module. User can be joined into the lesson in the app for answering question and interacting with the teacher in the lecture. When lecturers create a lesson module, students are able to join it. When lecturer change thepage of the PowerPoint in the lesson module, the device of student will be receive a message instantly a change the slide synchronistical. When the

lecturersraise a question in the lesson module, students are able to answer it. After that, there will be a result graph generated.

The Requirements

Functional Requirements for Mobile Application

- Allow student Browse the course list that enrolled in
- Allow student Browse the contents of course
- Allow student Browse the user details
- Allow student join into the lesson module
- Allow student download resources
- Auto changing PowerPoint page in Lesson when teacher change the page
- Allow student to answer questions in lesson module
- Allow student get the result or grade of the questions answered
- The Application will push notification to user when course is started
- Allow Teacher to start the lesson module
- Allow Teacher to browse the student list that joined to the lesson module or otherwise
- Allow Teacher to getting the results or grades from list of student after they submitting the answer in lesson module
- The grade or result will be shown as graph or diagram
- Support different type of questions in lesson module

Functional Requirements for eLessonModule

- **eLessonPlugin**
  - A new page type must be added which supports viewing PowerPoint in Moodle without downloading it to local computer.
  - The new page type should also support online eLesson, which once the online eLessonlaunched, teachers and students screens will be synchronized so that it could ease teachers to manage the class.
  - The new page type must be integrated with existed page types, which means new page type would not affect any functions that lesson module used to have.
  - Different teachers could launch the same eLesson at the same time, each online eLesson must has its own session.
- **Interaction with mobile client**
  - Web services for sending detailed courses information, eLesson pages information or any information needed must be built and well-documented.
  - Web services must be well-organized and reduces any redundant data in order to lessen the burden of campus network.
  - Respond time from server must not exceed five seconds no matter any actions are to be performed.

Functional Requirements for PowerPoint Conversion

- PowerPoint file must be converted into html code, or html page.
- All texts in PowerPoint file must be presented, and also their locations, colors, style, font size etc.
- Themes, textbox, pictures must be placed in right position, and their display priority must be considered.
- Page number, footer, notes should be included.
- Animation should be supported such as fade in and out.

Non-Functional Requirements for Mobile Application

Security:

- The application system should be security which should protect user secret information like password etc.

  It is possible to hidden their secret data using the hidden field.

Maintainability:

- The application system should be maintainable which should be consideredthefuture development .For example, it should support the newest version of mobile app.

  It is possible to implement the newest technology in our project. Like in our IOS development, we are using swift to develop the application. It should be supported at least few years.

Extensibility:

- The application system should be extendable which should allow to addfeatures, and carry-forward of customizations at next major version upgrade.

  It is possible to use this concept by using the Open Closed Principle in the program code.

Performance:

- The performance include the run time, response time of the program.

It is possible to use this concept which we can implement the AsyncTask in the program. Pre-loaded web service or data can help to reduce the response of calling a web services.

Platform compatibility:

- It should support different platform like IOS, android and website. Moreover, it should also support different resolution in different device.

It is possible to implement, developer can set the program code using the % in designing the User Interface.

Scalability (horizontal, vertical):

- It is commonthatthe mobile or tablet device have horizontal and vertical screens, it is possible to implement, developer can set the program code using the % in designing the User Interface.

Non-Functional Requirements for eLesson Module

- eLesson should be easy for teachers, managers or studentsto use even they haven't use eLesson before.
- The consistency between new page type and existed page types should keep the same.
- Add, modify, delete a PowerPoint page should be simple and user-friendly.
- Process of uploading the PowerPoint to eLesson should not be too long.
- Launching online lesson should not carry network a large burden, reduce any redundant data.

Non-Functional Requirements for PowerPoint Conversion

- Transaction time should be short.
- It should be able to handle different items and blocks of a PowerPoint file like themes, watermark etc.

- The output html file should be beautiful and well-organized.
- The output html file should be almost the same comparing with viewing the PowerPoint file using Microsoft PowerPoint application.

Problem Analysis

Use Case Diagram

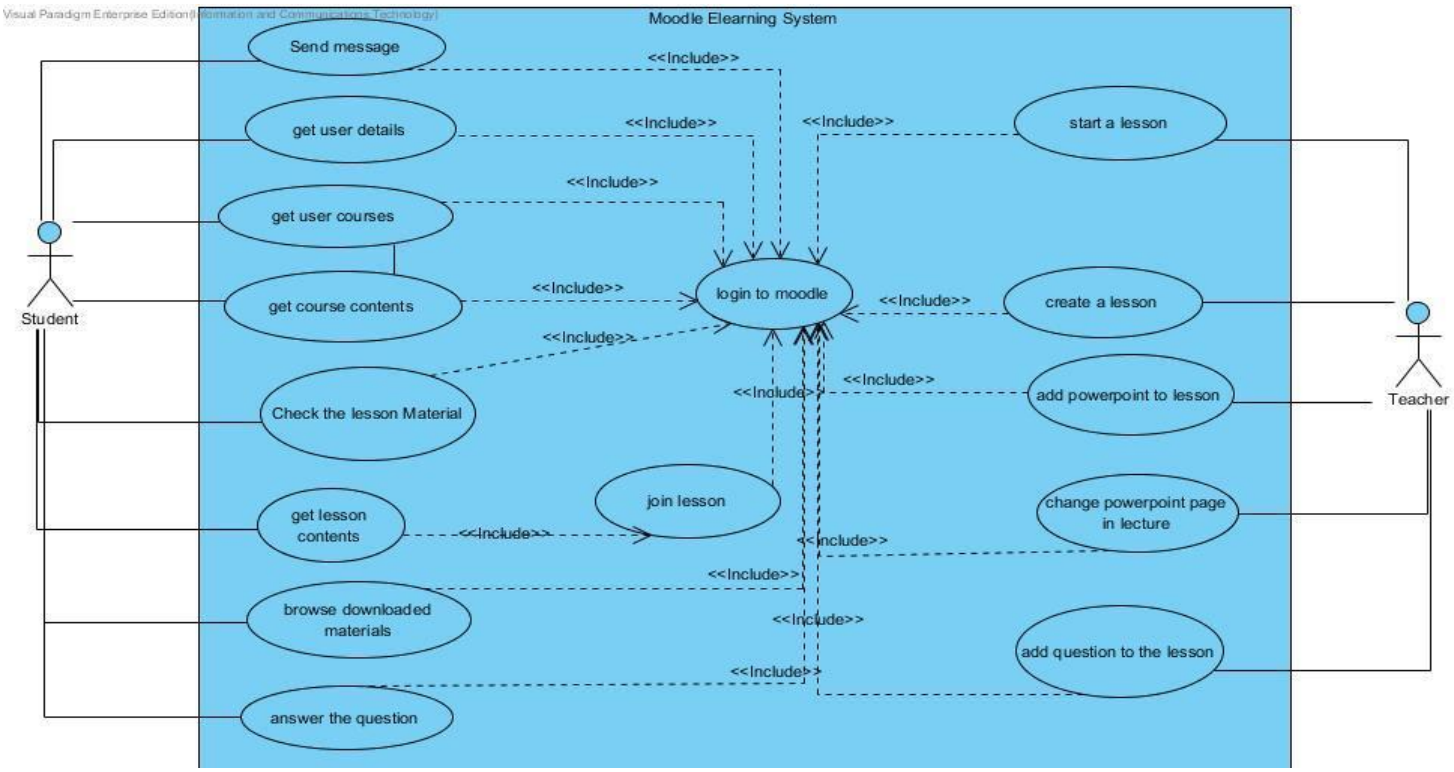


Figure 1 Use Case Diagram

Use Case Description& State Transition Diagram

Use Case Name	login to Moodle
Use Case ID	UC002
Actor(s)	user
Brief Description	Student can be login to the Moodle platform by their mobile phone
Pre-conditions	Student have a valid user name and password
Flow of events	1. student entering username and password in the mobile client 2. student click login button 3. Moodle authorize the username and password 4. Moodle return a token if the username and password are valid
Post-conditions	return a login successful web service token
Alternative flows and exception	Student entering wrong username or password, cause a error return by JSON
Assumptions	username and password are created

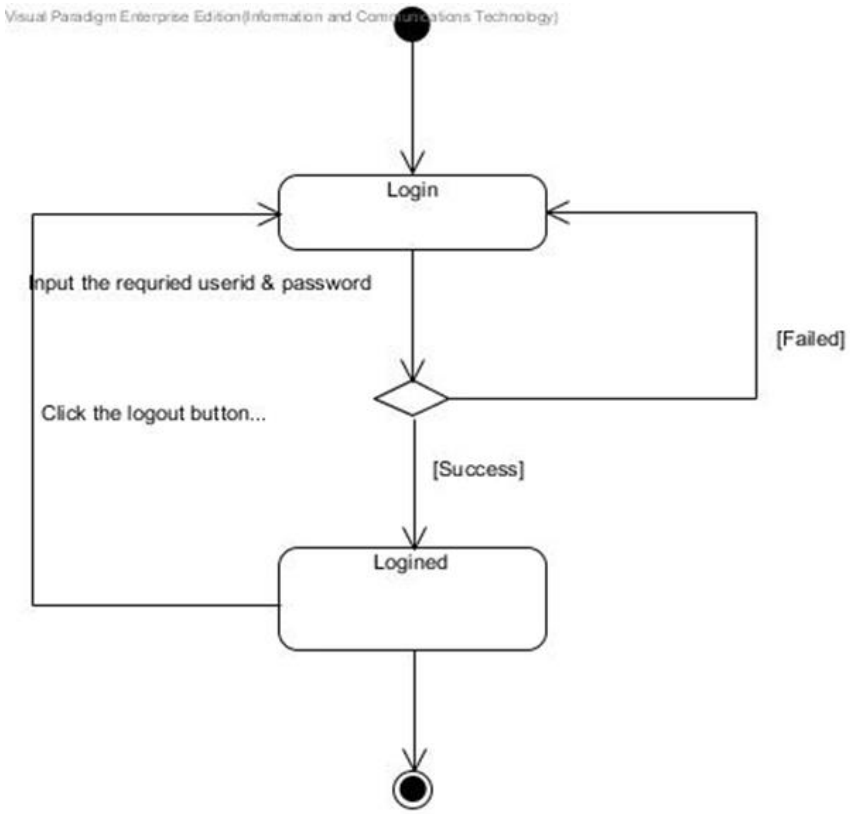


Figure 2UC002

Use Case Name	get user details
Use Case ID	UC003
Actor(s)	user
Brief Description	The user details like user icon, user name, email and enrolled courses can be viewed in mobile application
Pre-conditions	
Flow of events	1. Include (login to Moodle) 2. User browse their user info. 3. Moodle return the user details
Post-conditions	
Alternative flows and	

exception	
Assumptions	Student successfully logged in

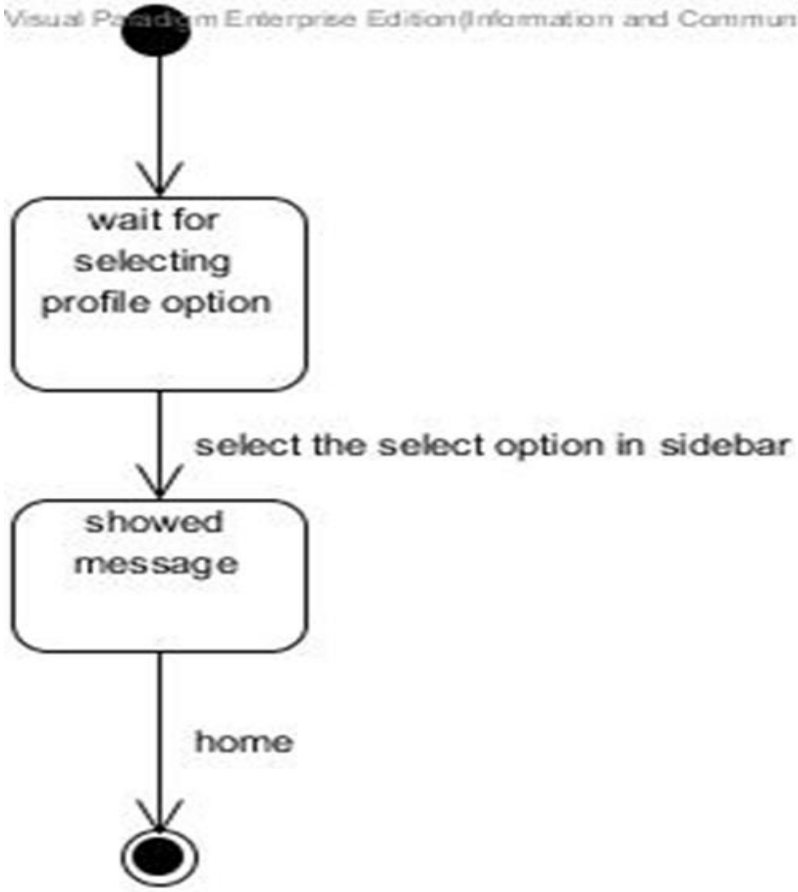


Figure 3UC003

Use Case Name	get user courses
Use Case ID	UC004
Actor(s)	user
Brief Description	Student can browse a list of courses that they are enrolled in
Pre-conditions	Student have been enrolled into the course

Flow of events	1. Include (login to Moodle) 2. Student browse the list of courses 3. Moodle pass back course details 4. Mobile app get the course teachers 5. Moodle pass back the teacher details 6. Mobile app present the course name, course description, course teacher in course list page
Post-conditions	
Alternative flows and exception	Server pass back error code due to server error
Assumptions	Student have been enrolled into the course

Use Case Name	get course contents
Use Case ID	UC005
Actor(s)	user
Brief Description	Student can select a course in the course list and view the course contents
Pre-conditions	Student have been enrolled in that course
Flow of events	1. Include (login to Moodle) 2. Student select a course to browse detail 3. Mobile app make the request by passing the courseid 4. Moodle pass back the course content details 5. Mobile app represent the course content in the interface

Post-conditions	
Alternative flows and exception	Server return unexpected error code
Assumptions	

Use Case Name	get lesson contents
Use Case ID	UC006
Actor(s)	student
Brief Description	Student can get the lesson contents in order to start the lesson
Pre-conditions	Student have a valid user name and password
Flow of events	1. Include(join lesson) 2. mobile interface display the lesson contents
Post-conditions	
Alternative flows and exception	Server failure error in return JSON
Assumptions	teacher have been start the lesson

Use Case Name	download resources
Use Case ID	UC007
Actor(s)	Student
Brief Description	Student can browse the resources from the course content, lesson module and download the resource the mobile device

Pre-conditions	Student have the privilege to the resource
Flow of events	1. Include(get user courses) 2. Include(get course contents) 3. DO 4 or 5 4. Select lesson and Review the lesson 5. Include(get lesson contents) 6. download the resources from the lesson module
Post-conditions	
Alternative flows and exception	
Assumptions	The internet condition is suitable for downloading the resource

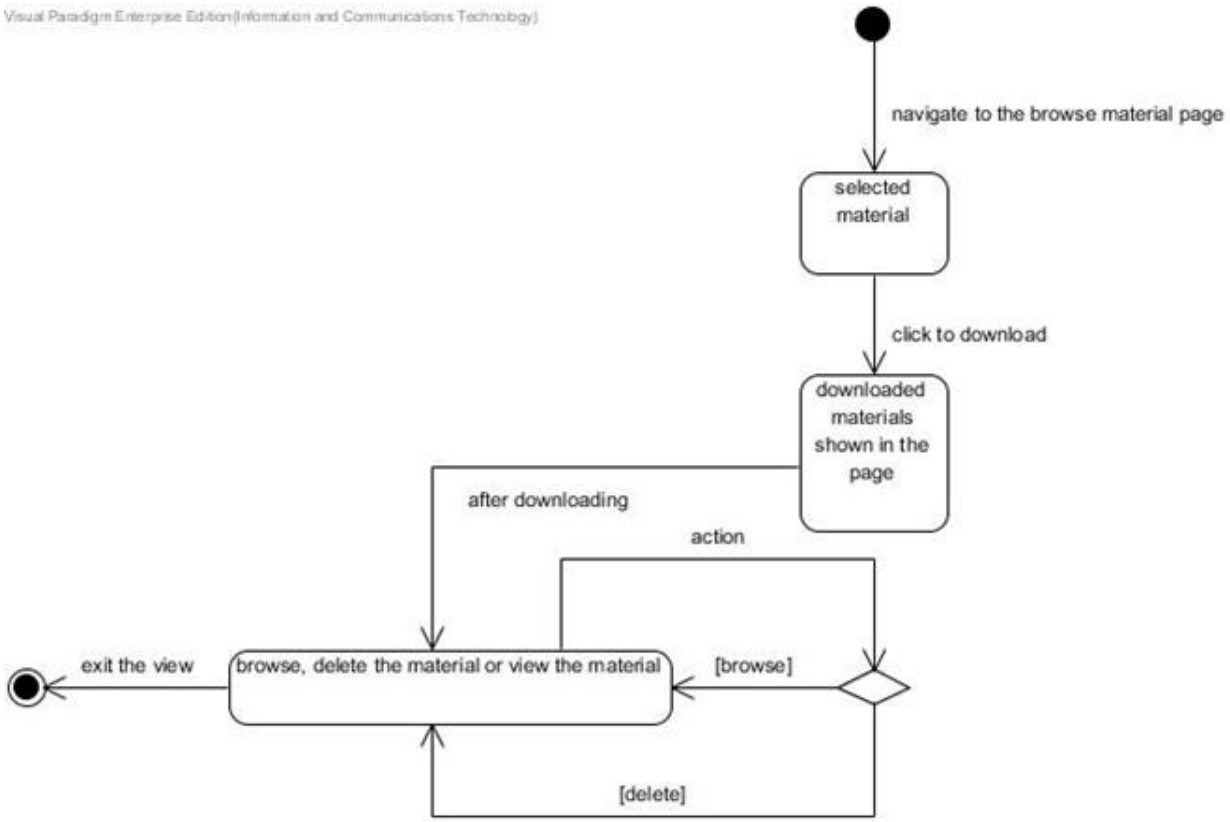


Figure 4 UC007

Use Case Name	join lesson
Use Case ID	UC008
Actor(s)	student
Brief Description	Student join to the lesson module in Moodle to start the lesson
Pre-conditions	The teacher have been start the lesson
Flow of events	1. Include(login to Moodle) 2. Include(get user course) 3. Include(get course contents) 4. student select a lesson and click the join button 5. Moodle check that the lesson is enabled to join and return the

	lesson contents
Post-conditions	
Alternative flows and exception	Student cannot join into the lesson because teacher not started the lesson
Assumptions	

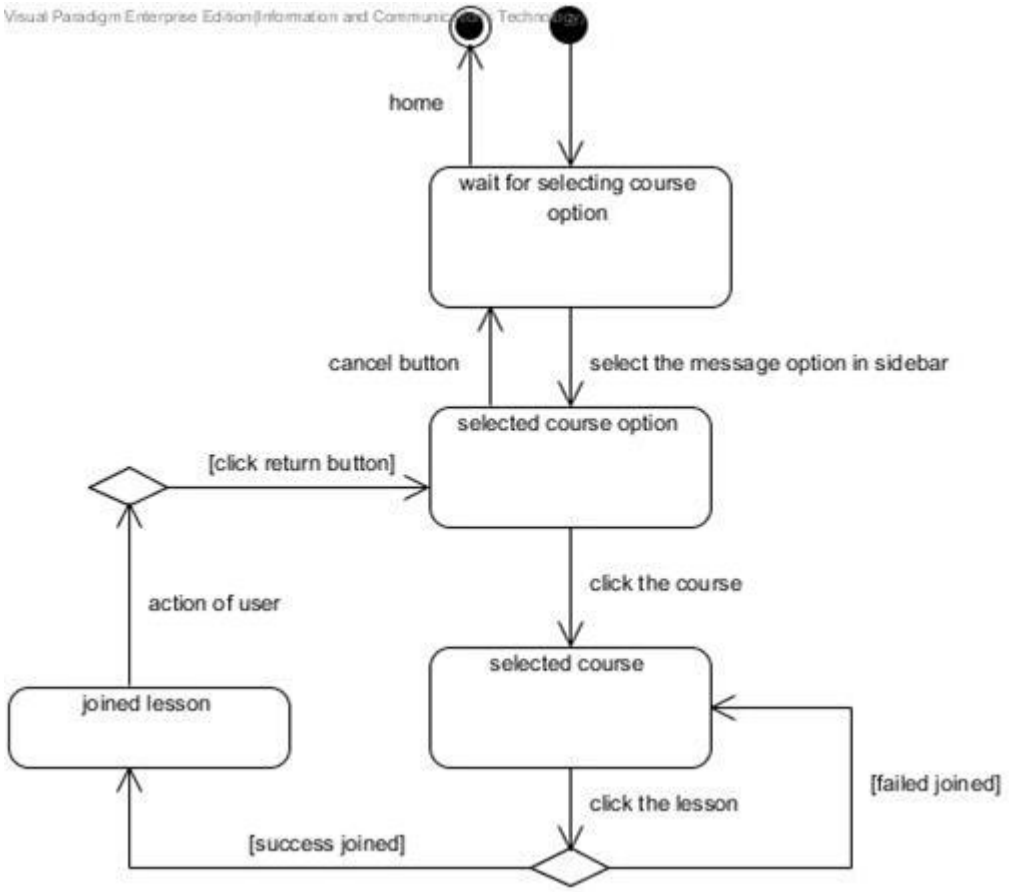


Figure 5 UC008

Use Case Name	browse downloaded materials
Use Case ID	UC009
Actor(s)	student

Brief Description	Student can be browse the downloaded materials and opens the material in other application
Pre-conditions	Student have been downloaded some materials
Flow of events	1. Include(login to Moodle) 2. student navigate to the browse material page 3. downloaded materials shown in the page 4. student can browse, delete the material or view the material
Post-conditions	
Alternative flows and exception	
Assumptions	username and password are created

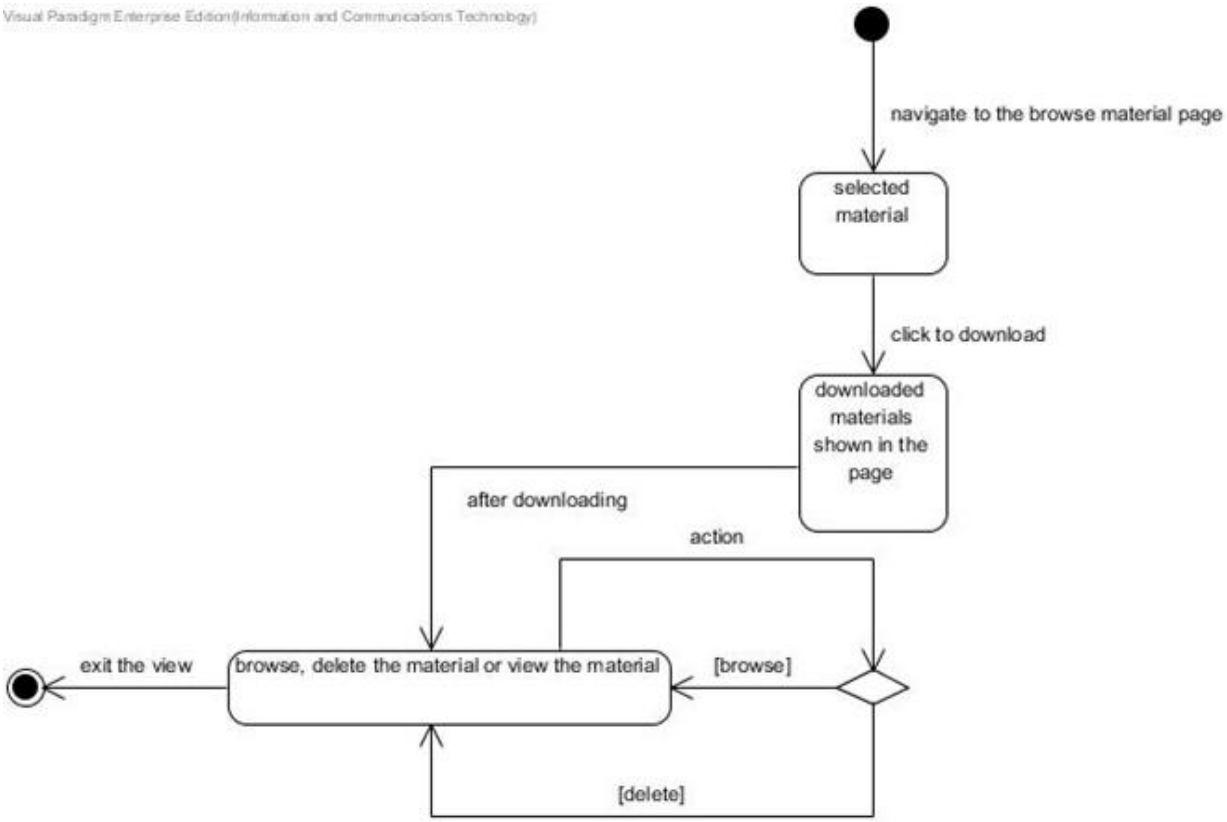


Figure 6 UC009

Use Case Name	answer the question
Use Case ID	UC100
Actor(s)	student
Brief Description	Student can answer the question in lesson module
Pre-conditions	
Flow of events	1. Include (get lesson contents) 2. Join the lesson 3. Answer the question 4. Submit the answer
Post-conditions	
Alternative flows and exception	
Assumptions	1. The lesson have been started

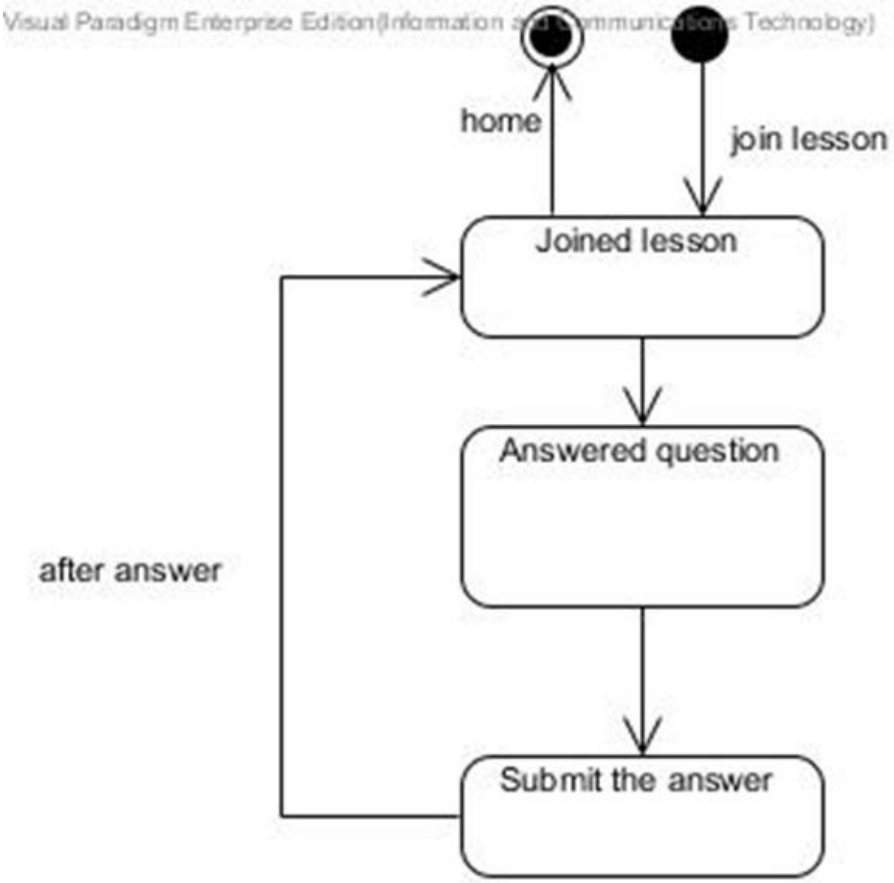


Figure 7 UC100

Use Case Name	Send message
Use Case ID	UC208
Actor(s)	student
Brief Description	Student can message the others
Pre-conditions	
Flow of events	1. select the message option in actionbar 2. click the target user icon 3. input the message 4. Submit the message
Post-conditions	1.sent failed return to 3

Alternative flows and exception	
Assumptions	1.user must be logined in the system

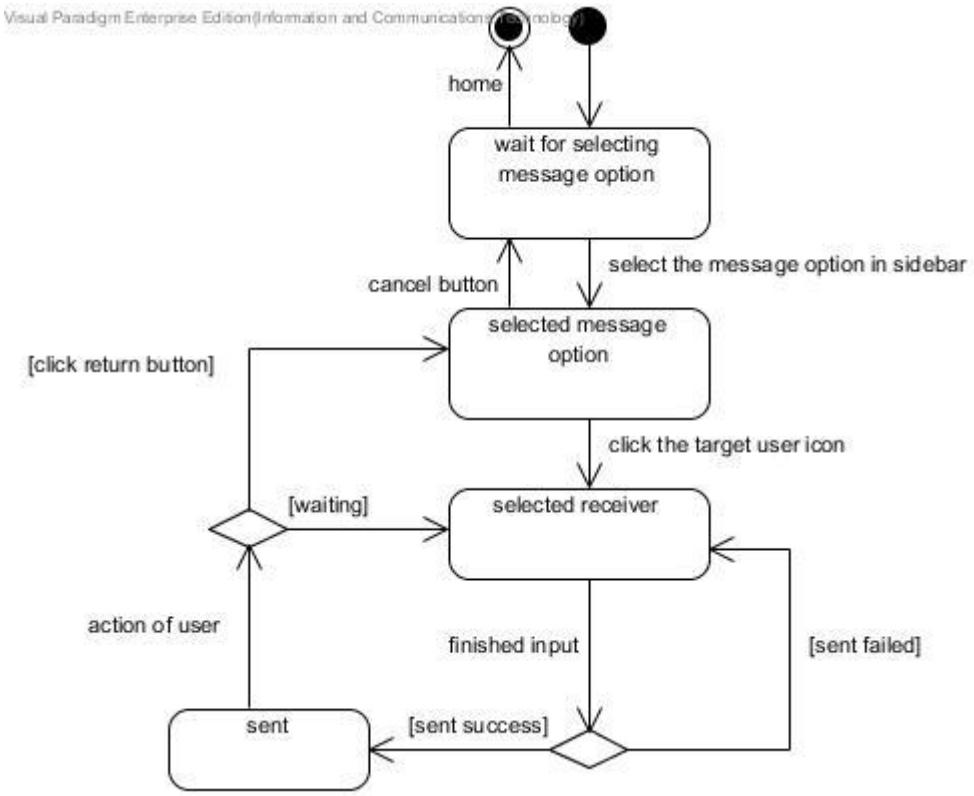


Figure 8 UC208

Use Case Name	start a lesson
Use Case ID	UC201
Actor(s)	teacher
Brief Description	Teacher are able to start a lesson in the Moodle and allow the students to join
Pre-conditions	
Flow of events	1. Include(login to Moodle) 2. teacher click into the lesson and start the lesson 3. wait for the student to join the lesson 4.end up the lesson
Post-conditions	The lesson state change to started
Alternative flows and exception	Student entering wrong username or password, cause a error return by JSON
Assumptions	The lesson have been created

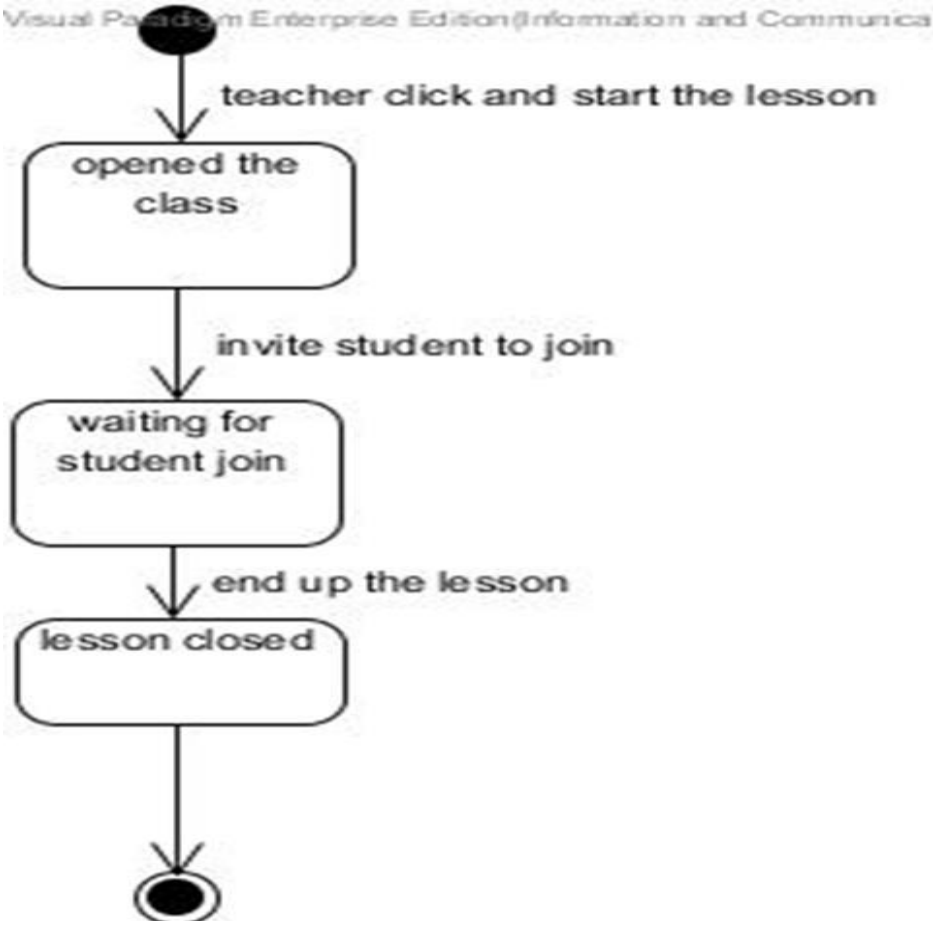


Figure 9 UC201

Use Case Name	create a lesson
Use Case ID	UC202
Actor(s)	teacher
Brief Description	Teacher can create a lesson for teaching purpose
Pre-conditions	Teacher have been logined in
Flow of events	1. Include (login to Moodle) 2. Teacher edit the page 3. Teacher add a lesson to the course 4. Teacher submit lesson name and lesson description and other information
Post-conditions	The lesson have been added to the course
Alternative flows and exception	Student entering wrong username or password, cause a error return by JSON
Assumptions	Teacher have been enrolled to the course

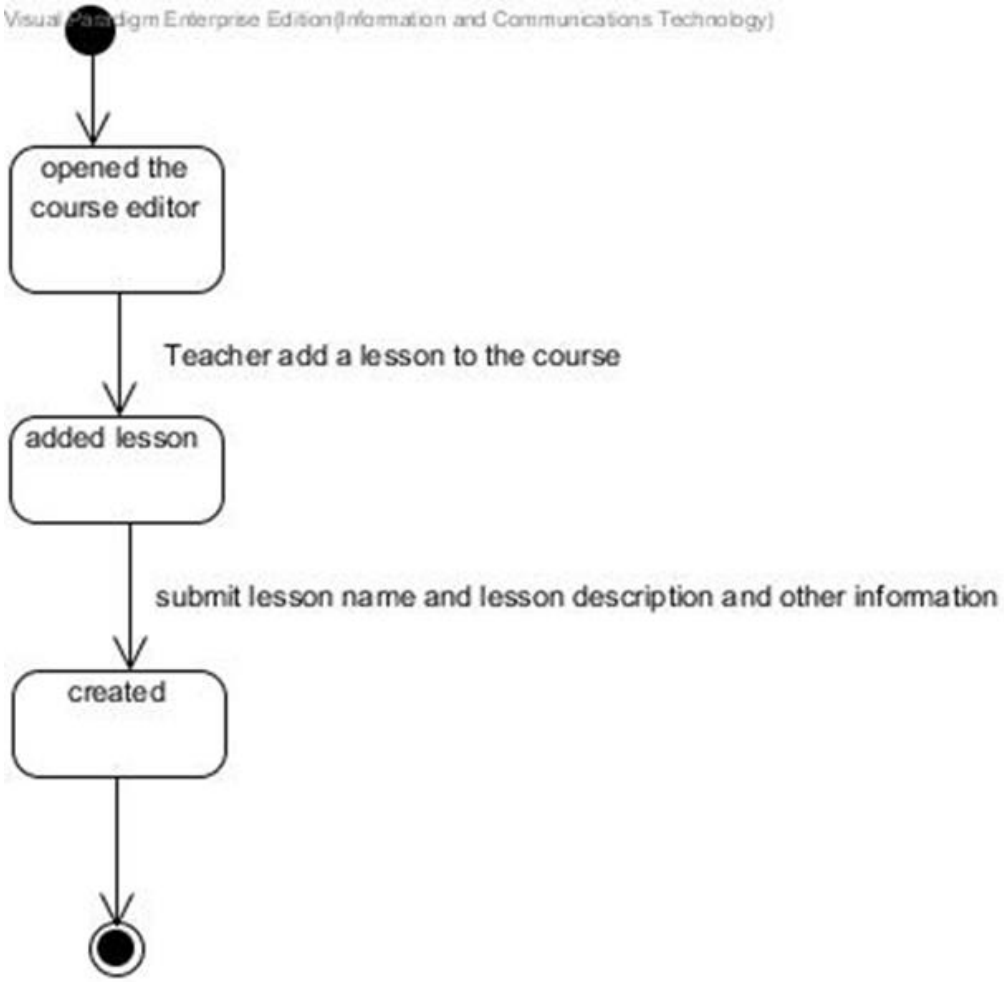


Figure 10UC202

Use Case Name	add PowerPoint to lesson
Use Case ID	UC203
Actor(s)	teacher
Brief Description	Teacher can edit the lesson contents, add a new content and add a PowerPoint for lecture
Pre-conditions	The lesson have been created The PowerPoint is in pptx format
Flow of events	1. Include (login to Moodle)

	2. teacher click edit lesson 3. teacher insert a new content 4. teacher select content type and upload a pptxPowerPoint 5. teacher click save button
Post-conditions	the PowerPoint have been added to the lesson
Alternative flows and exception	
Assumptions	

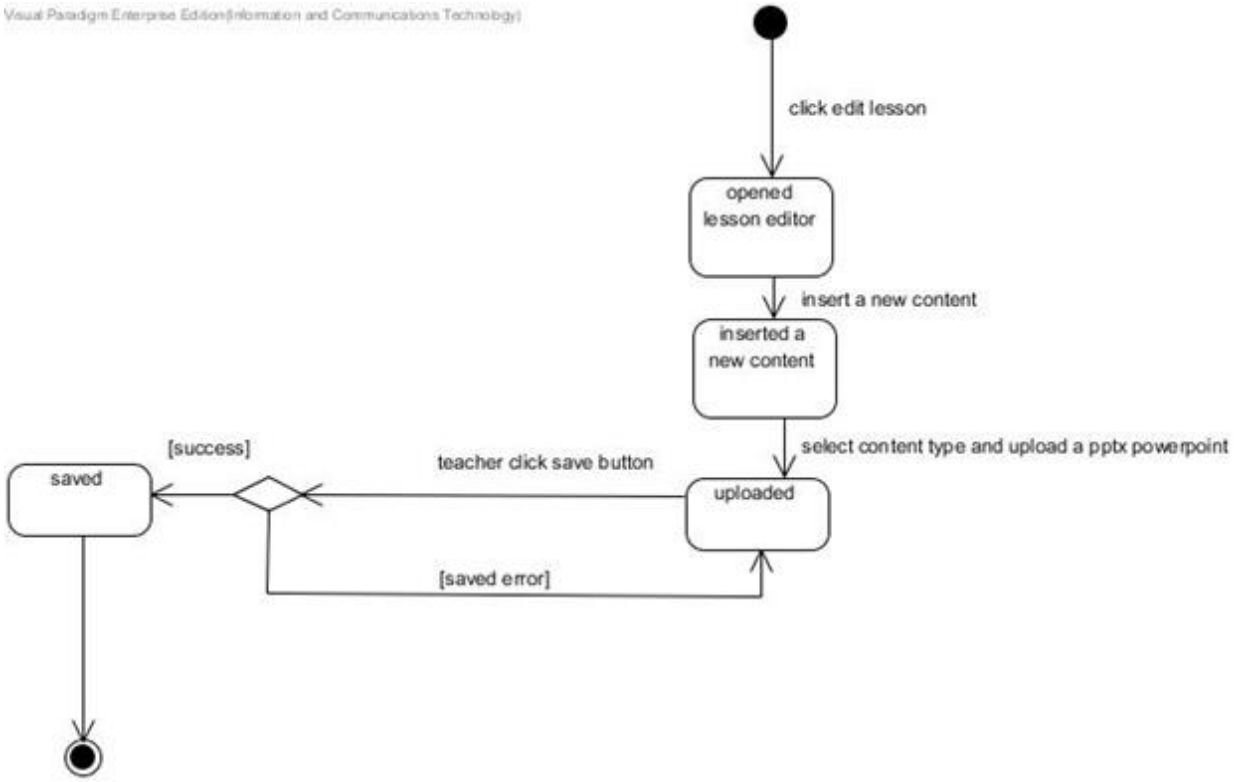


Figure 11UC203

Use Case Name	change PowerPoint page in lecture
Use Case ID	UC204

Actor(s)	teacher
Brief Description	teacher can change the PowerPoint page when teaching and student who joined to the lesson will be auto changing the PowerPoint page
Pre-conditions	1. Include (login to Moodle) 2. Teacher started the lesson 3. The PowerPoint have been added to the lesson
Flow of events	1. Teacher click the navigation control and change the PowerPoint page
Post-conditions	change the page sucessfully
Alternative flows and exception	
Assumptions	

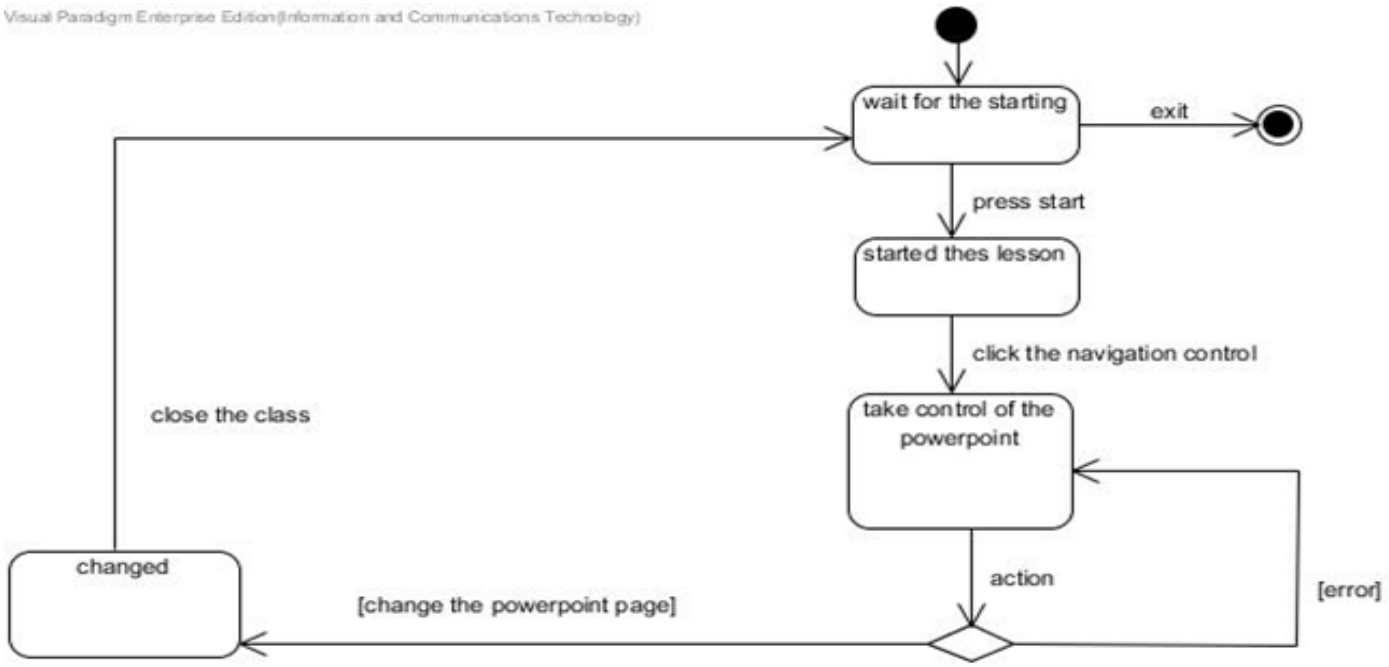


Figure 12UC204

Use Case Name	add question to lesson
Use Case ID	UC206
Actor(s)	teacher
Brief Description	Teacher can edit the lesson contents, add a new content and add a question for lecture
Pre-conditions	The lesson have been created.
Flow of events	1. Include (login to Moodle) 2. teacher click edit lesson 3. teacher insert a new content 4. teacher select content type and set the question

	5. teacher click save button
Post-conditions	
Alternative flows and exception	
Assumptions	

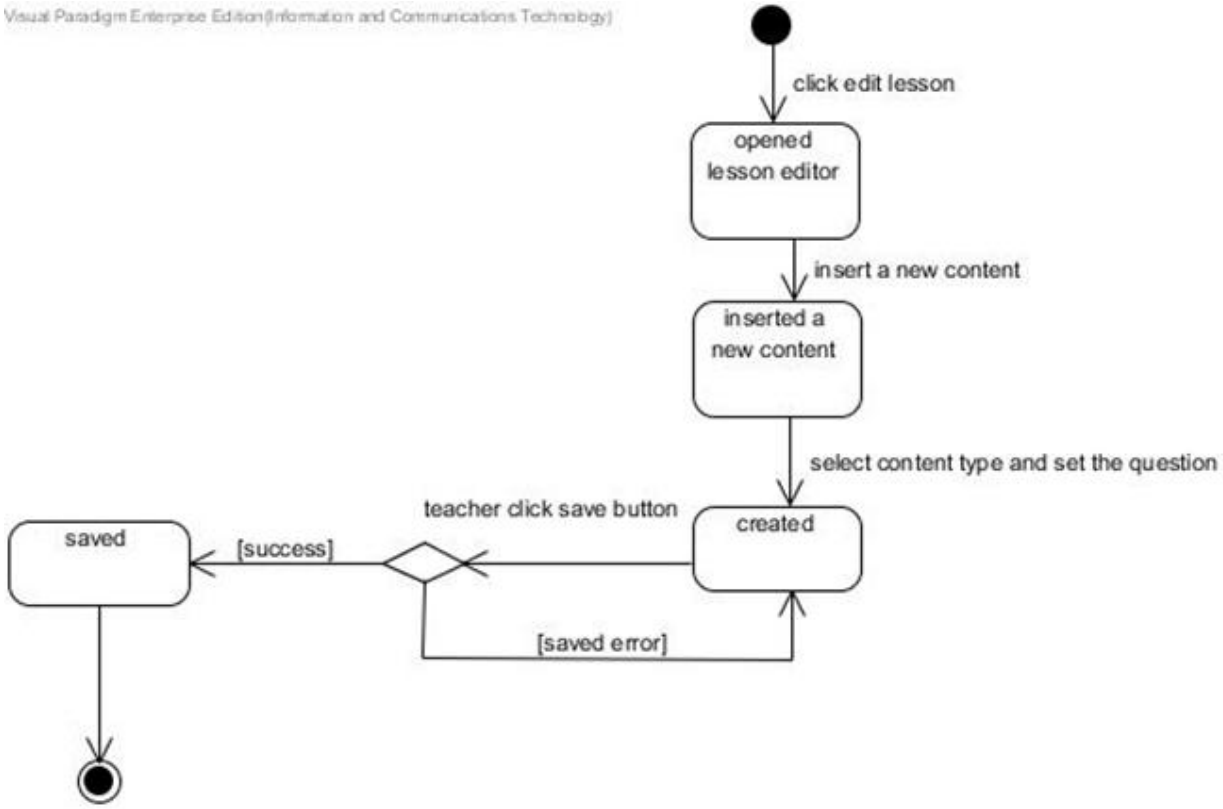


Figure 13 UC206

Use Case Name	Edit question in eLesson question bank
Use Case ID	UC207
Actor(s)	teacher
Brief Description	Teacher can edit the question that is created before

Pre-conditions	The lesson and question have been created.
Flow of events	1. Include (login to Moodle) 2. teacher click question tab 3. teacher click edit the question 4. teacher fills up the data of that question 5. teacher click save button
Post-conditions	The question is modified successfully
Alternative flows and exception	
Assumptions	

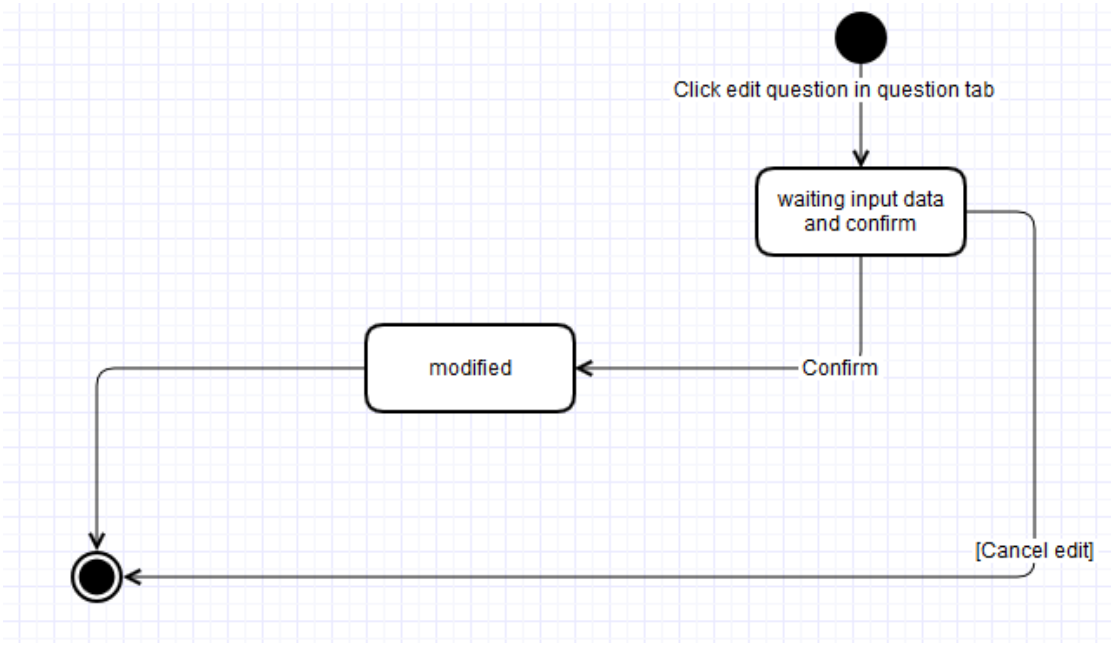


Figure 14: UC207

Use Case Name	Delete question in eLesson question bank
Use Case ID	UC208
Actor(s)	teacher

Brief Description	Teacher can delete the question
Pre-conditions	The lesson and question have been created.
Flow of events	1. Include (login to Moodle) 2. teacher click question tab 3. teacher click delete the question 4. teacher click confirm button
Post-conditions	The question is deleted successfully
Alternative flows and exception	
Assumptions	

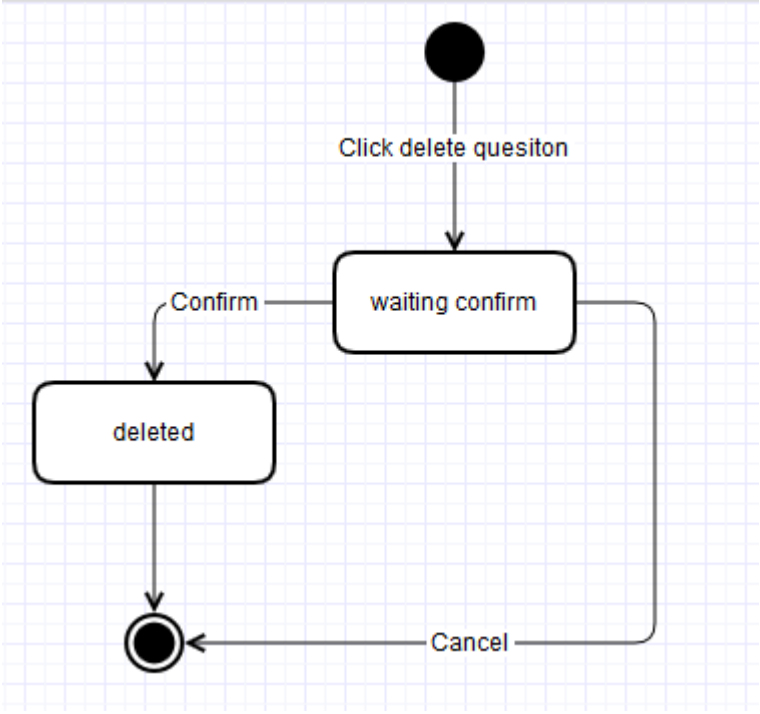


Figure 15: UC208

Use Case Name	Preview question in eLesson question bank
Use Case ID	UC209

Actor(s)	teacher
Brief Description	Teacher can preview the question
Pre-conditions	The lesson and question have been created.
Flow of events	1. Include (login to Moodle) 2. teacher click question tab 3. teacher click preview the question
Post-conditions	The question is showed in new window
Alternative flows and exception	
Assumptions	

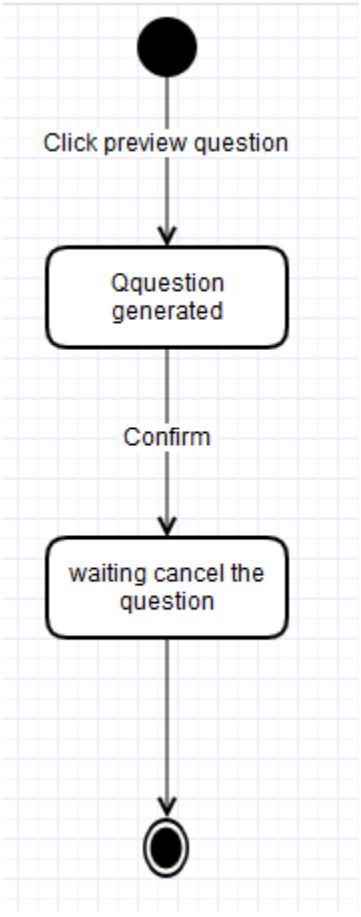


Figure 16: UC209

Use Case Name	Attach question to PowerPoint page
Use Case ID	UC210
Actor(s)	teacher
Brief Description	Teacher can attach the question into PowerPoint page
Pre-conditions	The lesson and question have been created.
Flow of events	1. Include (login to Moodle) 2. teacher click edit tab 3.teacherclick attach question on the row that contains the target PowerPoint page. 4. teacher choose which question should be attached and click continue
Post-conditions	The question is attached in PowerPoint page
Alternative flows and exception	The Question is already attached to the PowerPoint page before
Assumptions	

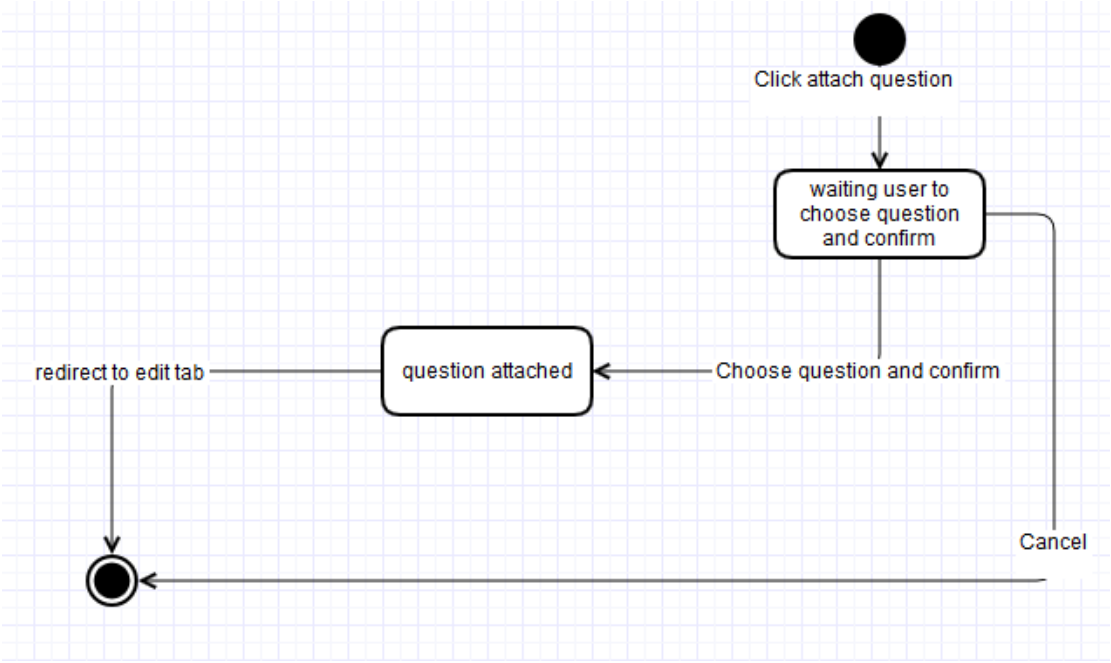


Figure 17: UC210

Use Case Name	Check whether students joined the online lesson
Use Case ID	UC211
Actor(s)	teacher
Brief Description	Teacher can check whether students joined the online lesson
Pre-conditions	The lesson has been created
Flow of events	1. Include (login to Moodle) 2. Include (Start lesson) 3.teacher click online students list
Post-conditions	
Alternative flows and exception	
Assumptions	

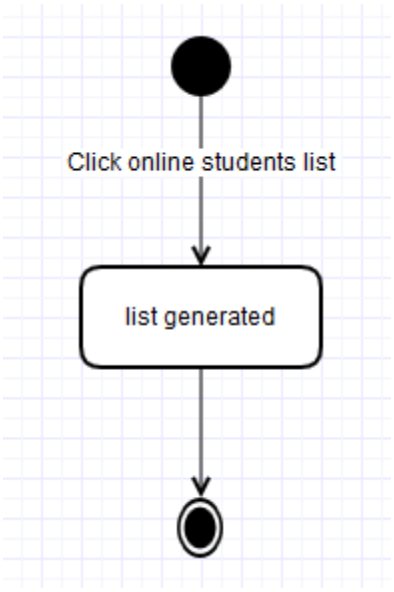


Figure 18: UC211

Use Case Name	View report on student answers
Use Case ID	UC212
Actor(s)	teacher
Brief Description	Teacher can view the report to see the statistic on student answers
Pre-conditions	The lesson and question have been created, question has been attached into question page.
Flow of events	1. Include (login to Moodle) 2. teacher click report tab 3.teacher choose the report he/she want to watch
Post-conditions	
Alternative flows and exception	No student has answer this question before
Assumptions	

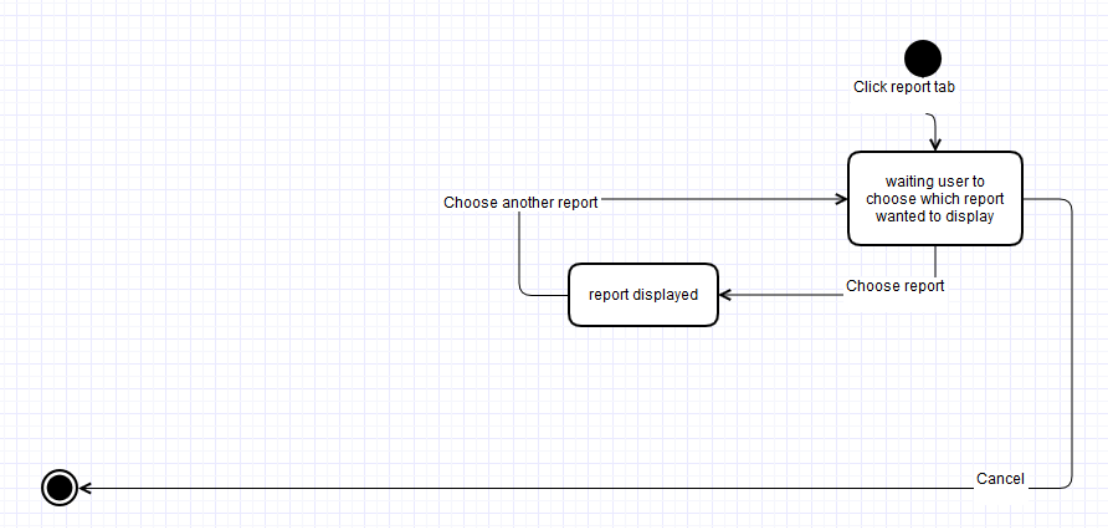


Figure 19: UC212

Class Diagram for iOS Application

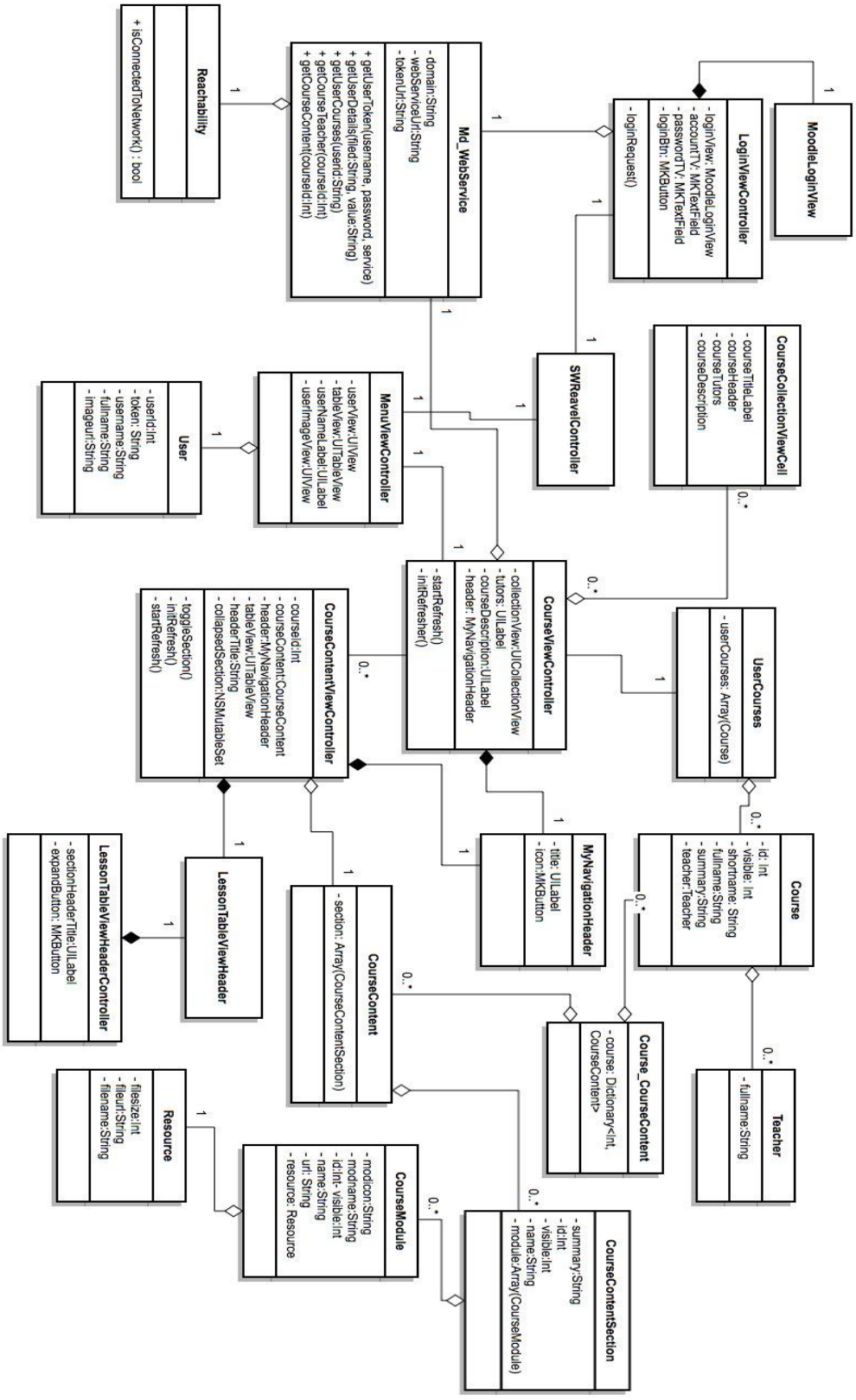
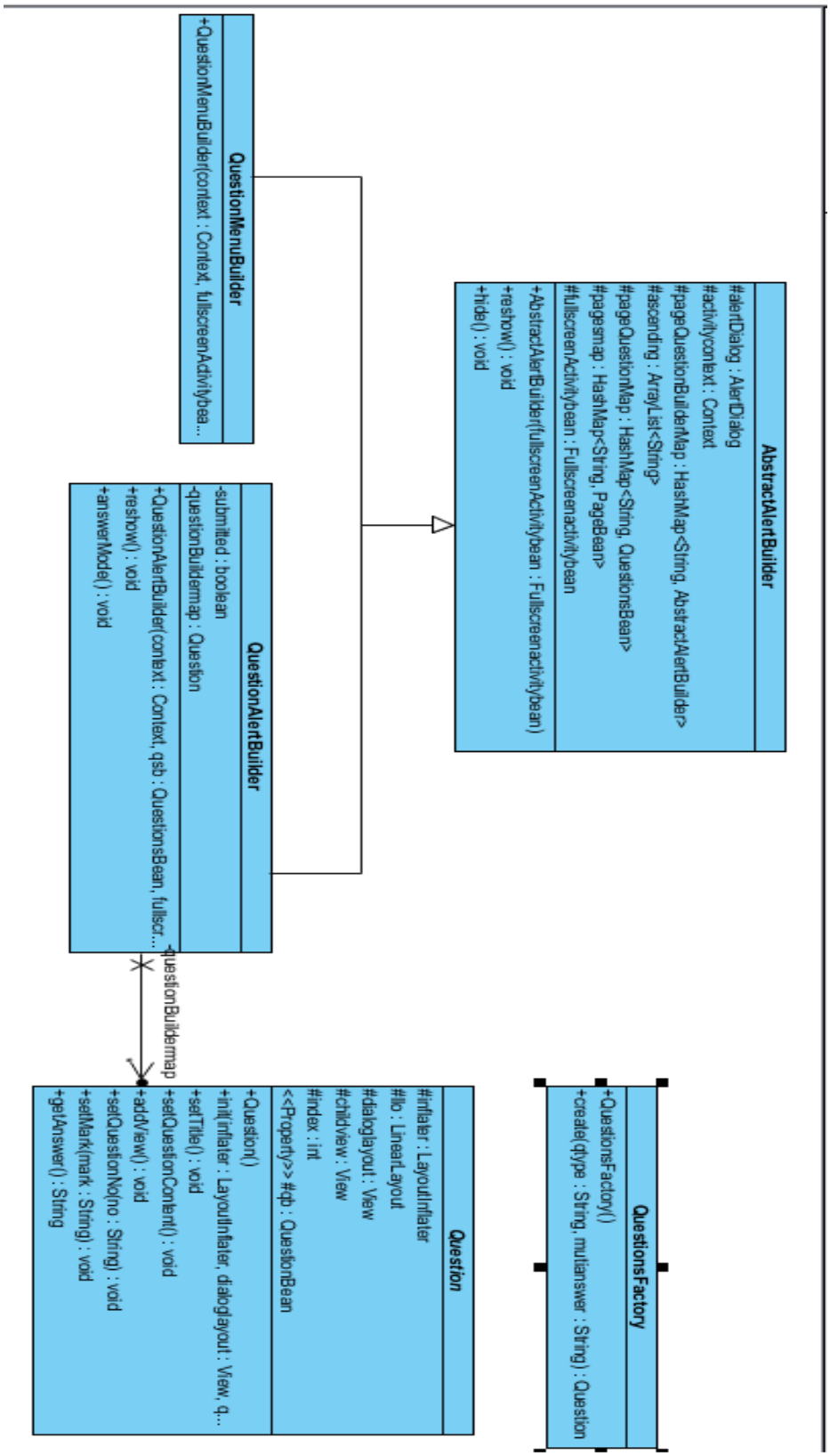


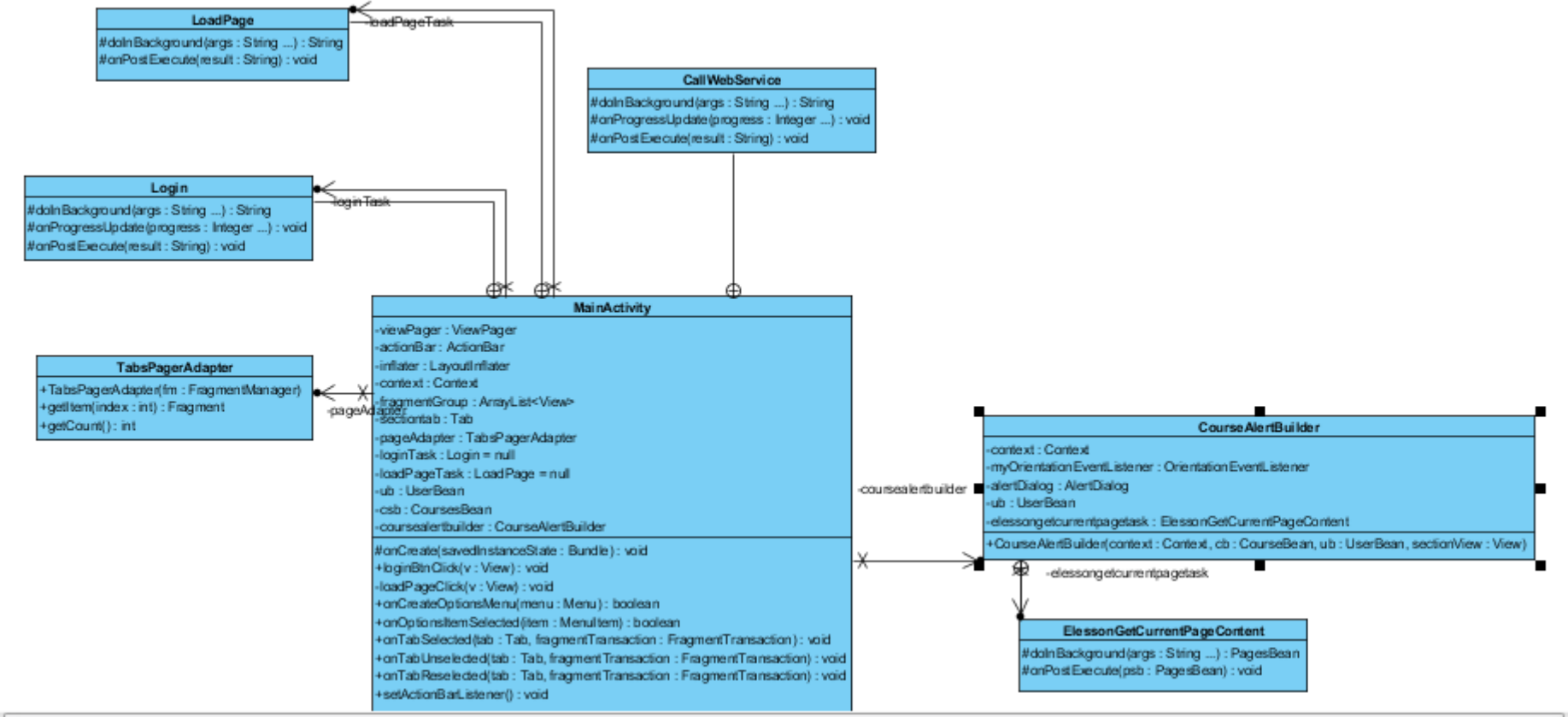
Figure 14Class Diagram for iOSMoodle Application

Class Diagram for Android Application

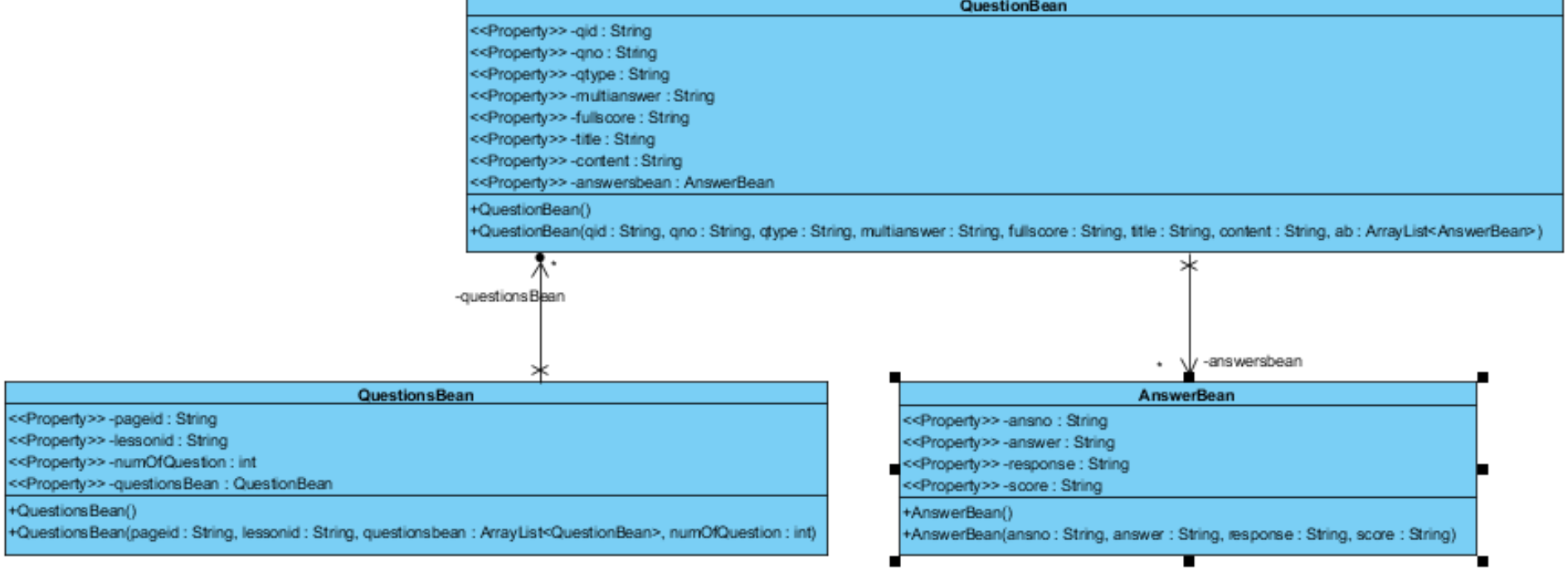


Question
#inflater : LayoutInflater
#lo : LinearLayout
#dialoglayout : View
#childview : View
#index : int
<<Property>> #qb : QuestionBean
+Question()
+init(inflater : LayoutInflater, dialoglayout : View, qb : QuestionBean, index : int) : void
+setTitle() : void
+setQuestionContent() : void
+addView() : vod
+setQuestionNo(no : String) : void
+setMark(mark : String) : void
+getAnswer() : String
-questionBuilderMap
#qb
ShortQuestion
-editText : EditText
+ShortQuestion()
+init(inflater : LayoutInflater, dialoglayout : Vie...
+getAnswer() : String
SingleMCQuestion
-radiogroup : RadioButton[]
+SingleMCQuestion()
+init(inflater : LayoutInflater, dialoglayout : View, qb : Que...
+setChoice() : void
+getAnswer() : String
TFMutiQuestion
-tfgroup : RadioButton[]
+TFMutiQuestion()
+init(inflater : LayoutInflater, dialoglayout : View, qb...
+getAnswer() : String
MutiMCQuestion
-checkboxgroup : CheckBox[]
+MutiMCQuestion()
+init(inflater : LayoutInflater, dial...
+setChoice() : void
+getAnswer() : String

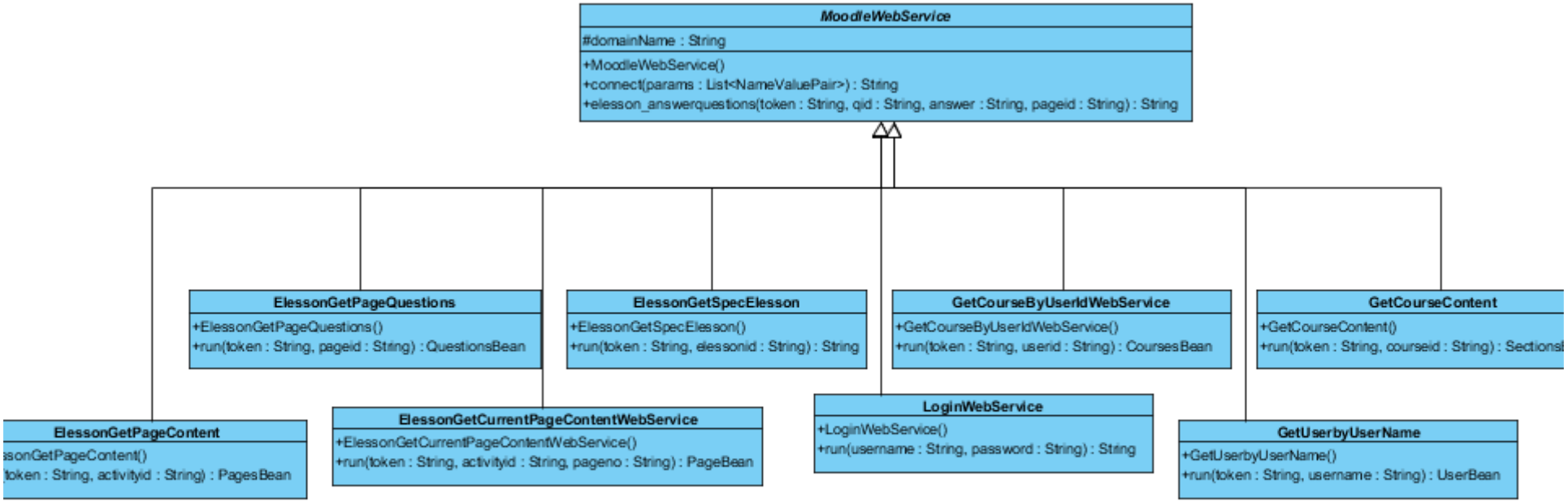
ElessonGetSpecElesson
+ElessonGetSpecElesson()
+runInBoken : String, elessonId : String) : String
<gpsp
WebAppInterface
~mContext : Context
~WebAppInterface(c : Context)
+showToast(toast : String) : void
+getPageContent(pageNum : String) : String
+getHeight() : String
+getWidth() : String
a
FullscreenActivity
-webview : WebView
-pagesmap : HashMap<String, PageBean>
-pageQuestionMap : HashMap<String, QuestionsBean>
-pageQuestionBuilderMap : HashMap<String, AbstractAlertBuilder>
-ascending : ArrayList<String>
-currentpage : String
-context : Context
~waiting Timer : Timer
~actionbarTimer : Timer
-actionbarindex : int
~actionbar : ActionBar
-fullscreenActivitybean : Fullscreenactivitybean
-changePageTask : ChangePage
-ub : UserBean
-psb : PageBean
-gpsp : ElessonGetSpecElesson = new ElessonGetSpecElesson()
#onCreate(savedInstanceState : Bundle) : void
+changePage(pageNum : String) : void
-initActionBar() : void
#onStop() : void
#onPause() : void
#onResume() : void
GetPageQuestions
#doInBackground(args : String ...) : String
#onPostExecute(result : String) : void
<changePageTask
ChangePage
#doInBackground(args : String ...) : String
#onPostExecute(result : String) : void
LoopTask
#doInBackground(args : String ...) : String
#onPostExecute(result : String) : void


LoadPage
#doInBackground(args : String ...) : String
#onPostExecute(result : String) : void
loadPageTask
CallWebService
#doInBackground(args : String ...) : String
#onProgressUpdate(progress : Integer ...) : void
#onPostExecute(result : String) : void
Login
#doInBackground(args : String ...) : String
#onProgressUpdate(progress : Integer ...) : void
#onPostExecute(result : String) : void
loginTask
MainActivity
-viewPager : ViewPager
-actionBar : ActionBar
-inflater : LayoutInflater
-context : Context
-fragmentGroup : ArrayList<View>
-sectiontab : Tab
-pageAdapter : TabsPagerAdapter
-loginTask : Login = null
-loadPageTask : LoadPage = null
-ub : UserBean
-csb : CoursesBean
-coursealertbuilder : CourseAlertBuilder
#onCreate(savedInstanceState : Bundle) : void
+loginBtnClick(v : View) : void
-loadPageClick(v : View) : void
+onCreateOptionsMenu(menu : Menu) : boolean
+onOptionsItemSelected(item : MenuItem) : boolean
+onTabSelected(tab : Tab, fragmentTransaction : FragmentTransaction) : void
+onTabUnselected(tab : Tab, fragmentTransaction : FragmentTransaction) : void
+onTabReselected(tab : Tab, fragmentTransaction : FragmentTransaction) : void
+setActionBarListener() : void
TabsPagerAdapter
+TabsPagerAdapter(fm : FragmentManager)
+getItem(index : int) : Fragment
+getCount() : int
pageAdapter
-coursealertbuilder
CourseAlertBuilder
-context : Context
-myOrientationEventListener : OrientationEventListener
-alertDialog : AlertDialog
-ub : UserBean
-elessongetcurrentpagetask : ElessonGetCurrentPageContent
+CourseAlertBuilder(context : Context, cb : CourseBean, ub : UserBean, sectionView : View)
-elessongetcurrentpagetask
ElessonGetCurrentPageContent
#doInBackground(args : String ...) : PagesBean
#onPostExecute(psb : PagesBean) : void

CoursesBean
<<Property>> -coursesbean : CourseBean
+CoursesBean()
coursesbean
CourseBean
<<Property>> -courseid : String
<<Property>> -shortname : String
<<Property>> -fullname : String
<<Property>> -idnumber : String
<<Property>> -visible : String
<<Property>> -summary : String
<<Property>> -ssb : SectionsBean = new SectionsBean()
+CourseBean(courseid : String, shortname : String, fullname : String, idnumber : String, visible : String, summary : String)
+CourseBean()
-ssb
SectionsBean
<<Property>> -sections : SectionBean
+SectionsBean()
+SectionsBean(sections : ArrayList<SectionBean>)
SectionBean
<<Property>> -sectionid : String
<<Property>> -sectionName : String
<<Property>> -sectionvisable : String
<<Property>> -summary : String
<<Property>> -summaryformat : String
<<Property>> -modules : ActivityBean
+SectionBean()
+SectionBean(sectionid : String, sectionName : String, sectionvisab...
-modules
ActivityBean
<<Property>> -activityid : String
<<Property>> -url : String
<<Property>> -activityname : String
<<Property>> -activityvisible : String
<<Property>> -modicon : String
<<Property>> -modname : String
<<Property>> -modplural : String
<<Property>> -indent : String
+ActivityBean(activityid : String, url : String, activityname : String, activityvisible : String, modi...
+ActivityBean()


QuestionBean
<<Property>> -qid : String
<<Property>> -qno : String
<<Property>> -qtype : String
<<Property>> -multianswer : String
<<Property>> -fullscore : String
<<Property>> -title : String
<<Property>> -content : String
<<Property>> -answersbean : AnswerBean
+QuestionBean()
+QuestionBean(qid : String, qno : String, qtype : String, multianswer : String, fullscore : String, title : String, content : String, ab : ArrayList<AnswerBean>)
-questionsBean
-answersbean
QuestionsBean
<<Property>> -pageid : String
<<Property>> -lessonid : String
<<Property>> -numOfQuestion : int
<<Property>> -questionsBean : QuestionBean
+QuestionsBean()
+QuestionsBean(pageid : String, lessonid : String, questionsbean : ArrayList<QuestionBean>, numOfQuestion : int)
AnswerBean
<<Property>> -ansno : String
<<Property>> -answer : String
<<Property>> -response : String
<<Property>> -score : String
+AnswerBean()
+AnswerBean(ansno : String, answer : String, response : String, score : String)

PageBean
+PageBean()
PageBean
+PageBean()
UserBean
+UserBean()
FullscreenactivityBean


MoodleWebService
#domainName : String
+MoodleWebService()
+connect(params : List<NameValuePair>) : String
+elesson_answerquestions(token : String, qid : String, answer : String, pageid : String) : String
ElessonGetPageQuestions
+ElessonGetPageQuestions()
+run(token : String, pageid : String) : QuestionsBean
ElessonGetSpecElesson
+ElessonGetSpecElesson()
+run(token : String, elessonid : String) : String
GetCourseByUserIdWebService
+GetCourseByUserIdWebService()
+run(token : String, userid : String) : CoursesBean
GetCourseContent
+GetCourseContent()
+run(token : String, courseid : String) : Sections
ElessonGetPageContent
sonGetPageContent()
token : String, activityid : String) : PagesBean
ElessonGetCurrentPageContentWebService
+ElessonGetCurrentPageContentWebService()
+run(token : String, activityid : String, pageno : String) : PageBean
LoginWebService
+LoginWebService()
+run(username : String, password : String) : String
GetUserbyUserName
+GetUserbyUserName()
+run(token : String, username : String) : UserBean

Class Diagram for PowerPoint Content Parser Program

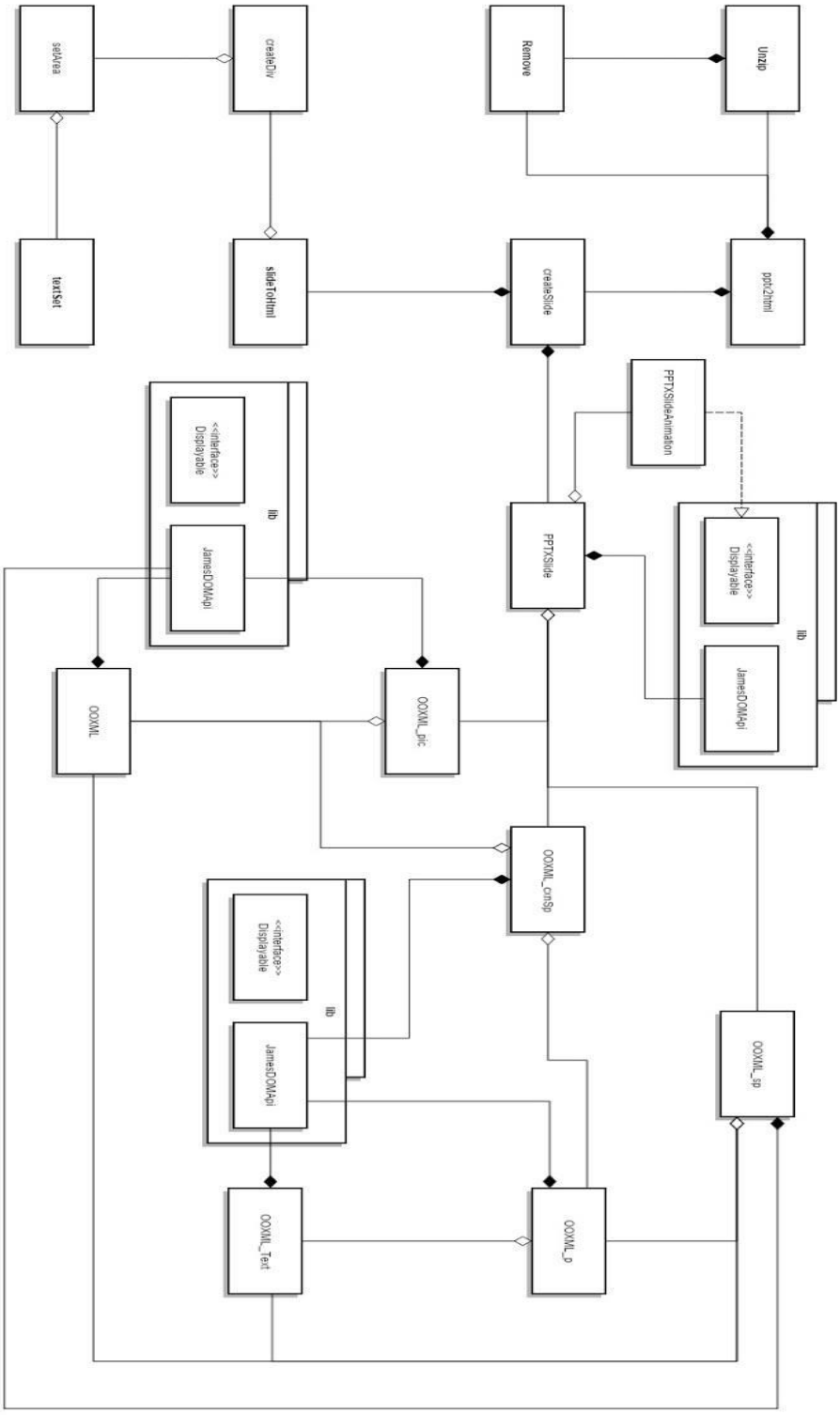
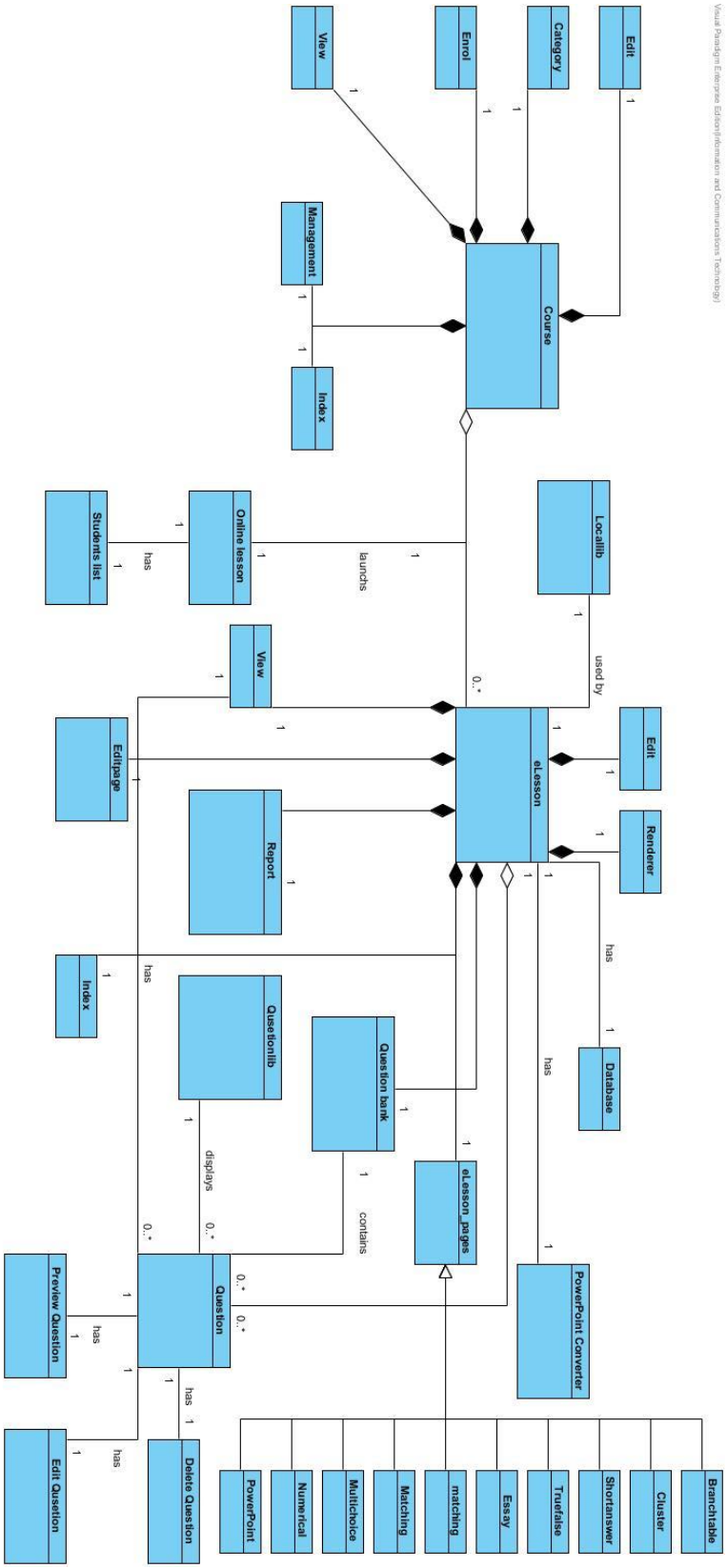


Figure 15PowerPoint Content Parser Program Class Diagram

Class Diagram for eLesson module



Context Diagram

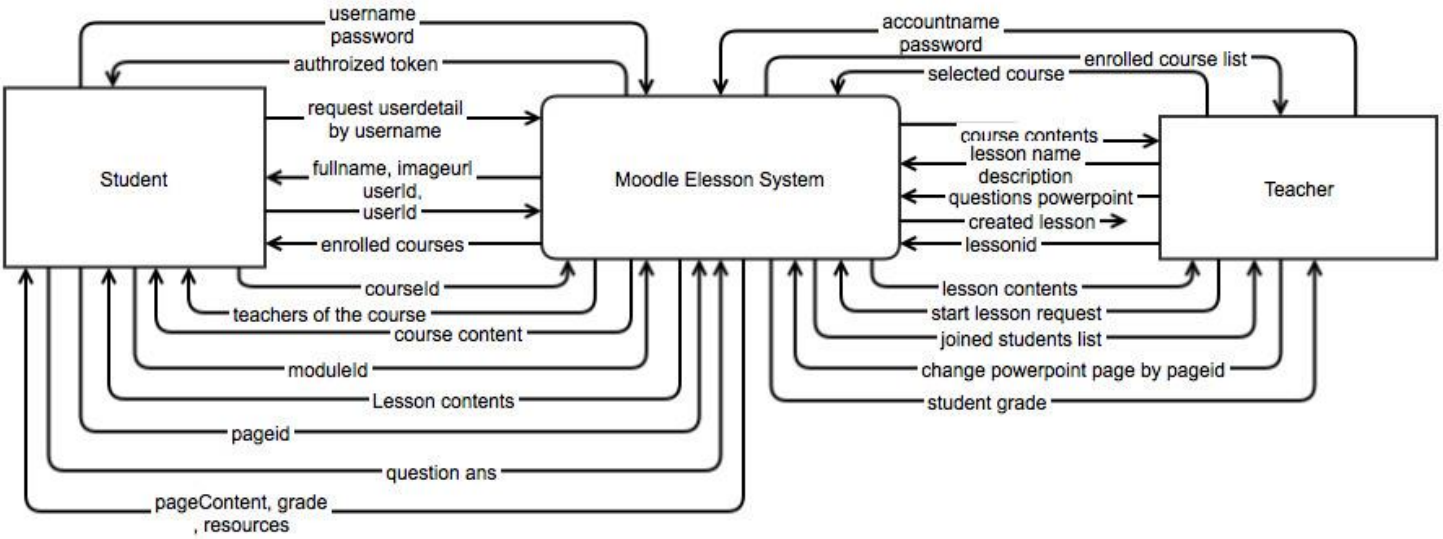
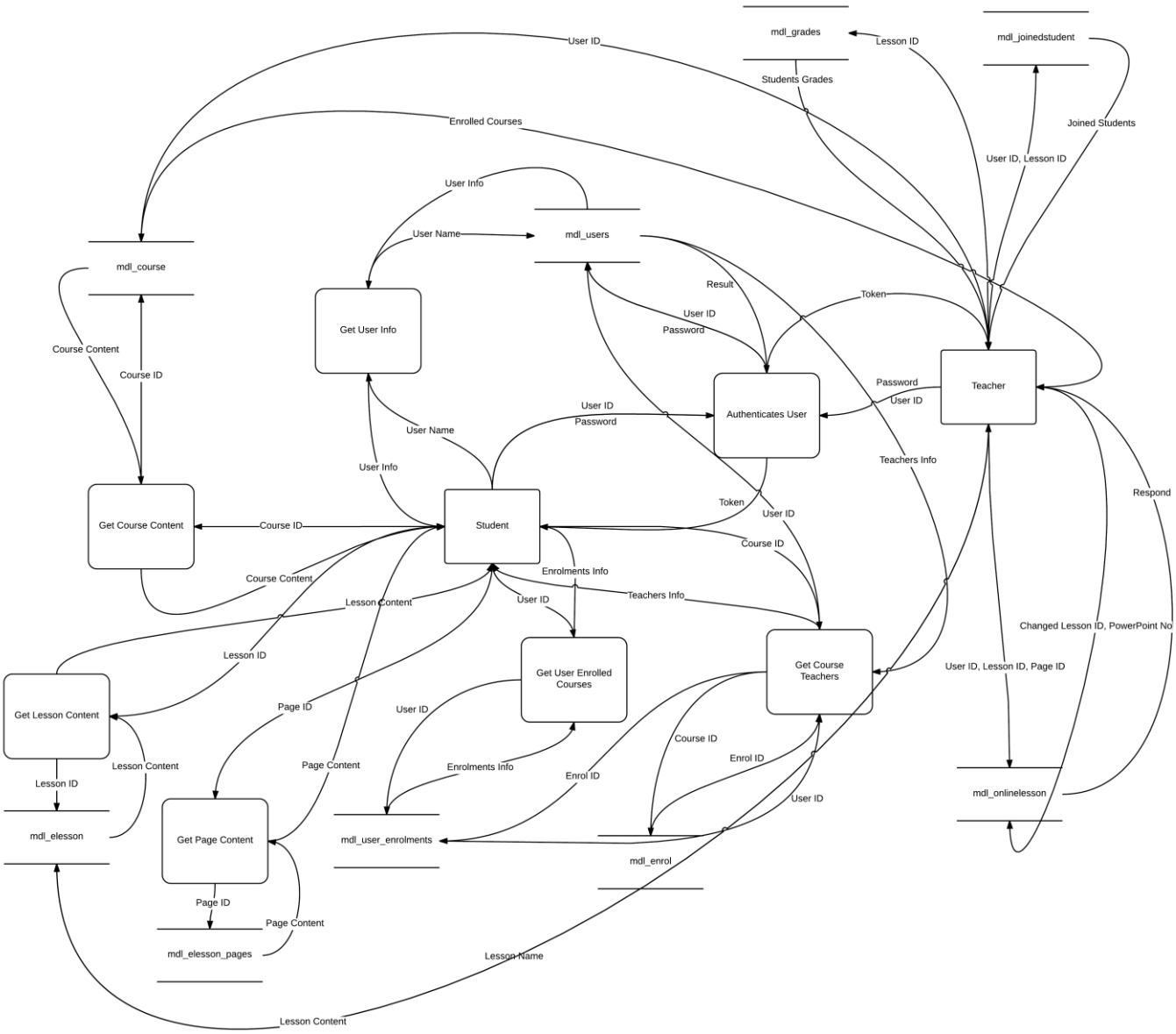


Figure 16 Context Diagram

The data flow of the system is between the Student to Moodle platform and teacher to Moodle platform.

DFD For Moodle System(Level 0)



Data Flow Diagram

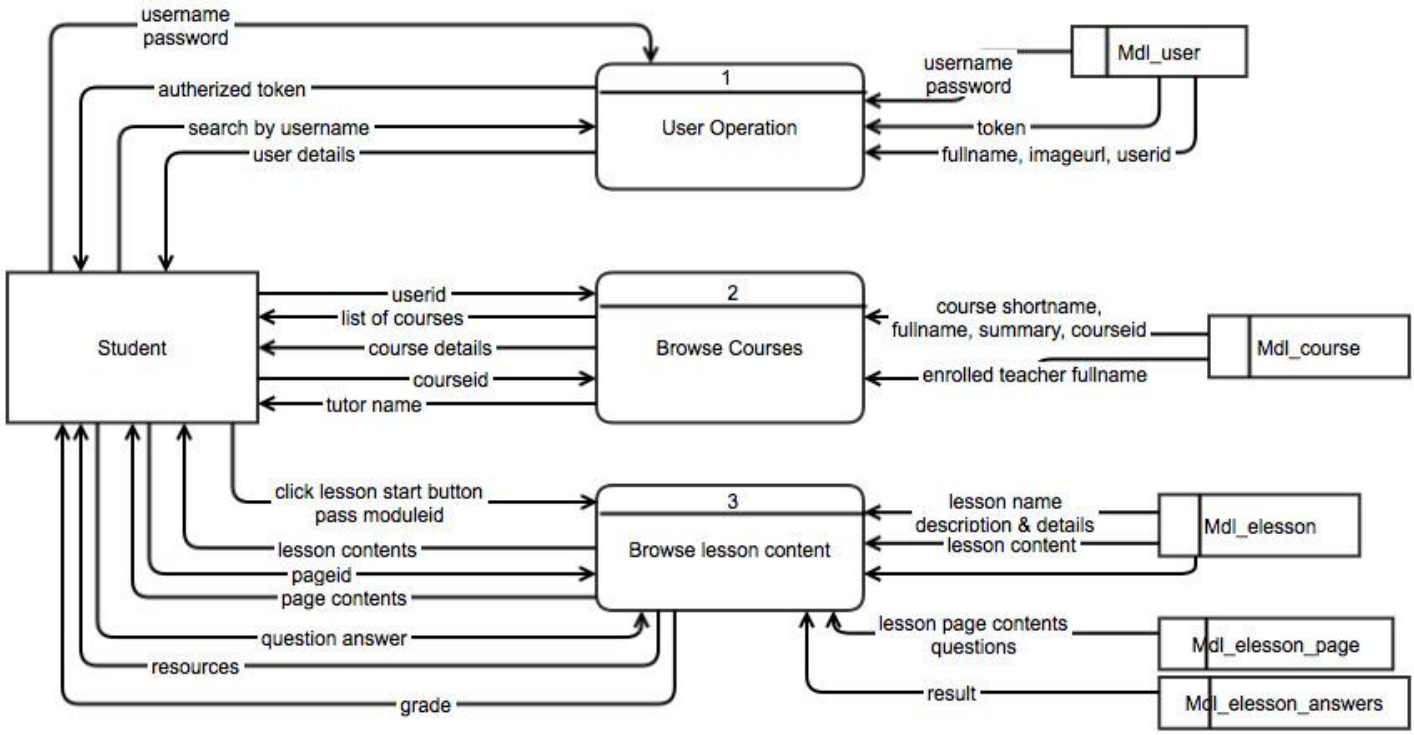


Figure 17 DFD level 1 for student

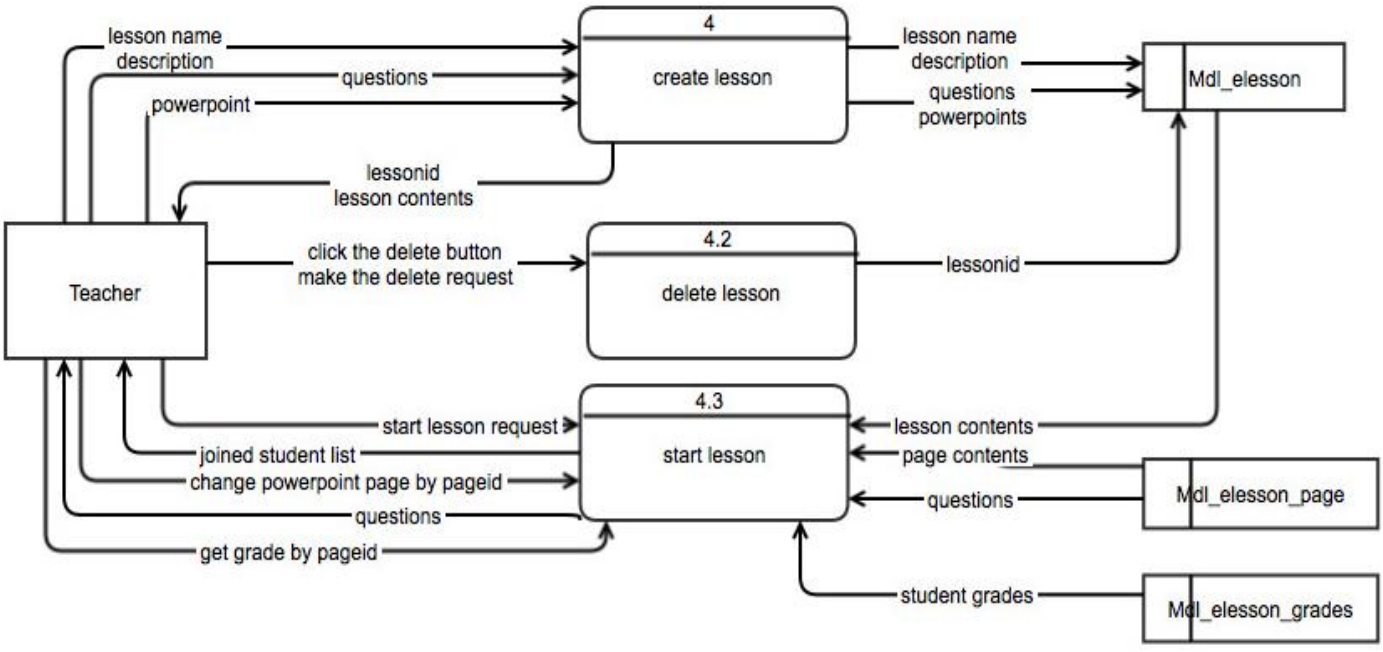


Figure 18 DFD level 1 for teacher

Activity Diagram ForPowerPoint HTML Parser

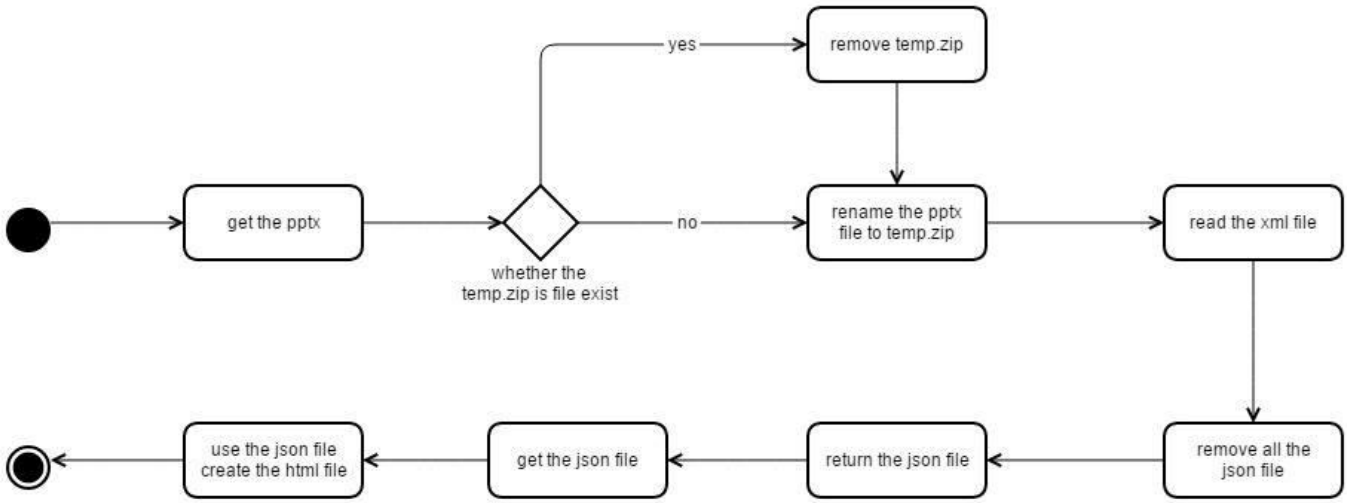


Figure 19Activity Diagram ForPowerPoint HTML Parser

Sequence Diagram ForPowerPoint HTML Parser

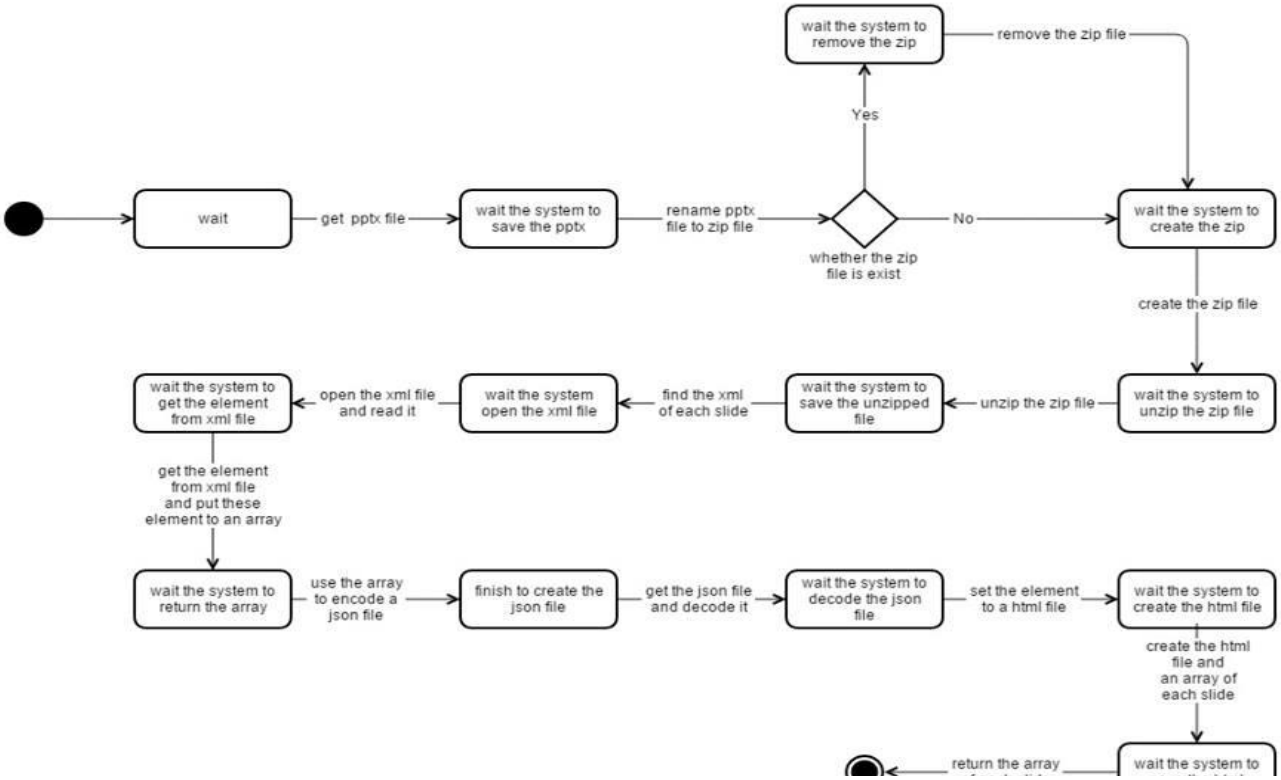


Figure 20Sequence Diagram for PowerPoint HTML Parser

Initial Design

User Analysis

User Characteristics

Three Areas

1. Physical differences -Age

Our main target of our user is teenager. Normally, they are between 16 and around 25. However children,adult and elder is able to use if they have a certain degree of knowledge in electronic device like mobile phone.

2. Primary user - Student

The primary user is the students which directly install the mobile app in their smart phone or tablet. Also they have to enter their privacy detail like userid and personal password in order to login to the system. The user should be expected a certain degree of knowledge in electronic device like smart phone.

3. Tool preferences - Required Hardware

The user is required the smart phone or tablet in order to use the app.

Techniques for observing and listening to users

1. Consideration

The app is the multi- platform mobile app. It is required stable internet connection when the student join the class .We can easily to find out the user face to face. The method of observing can held on everywhere which have a stable wifi connection, computer, computer, smart phone or tablet. It is more efficiency to collect the feedback.

2. Method

Role playing

One of the best techniques for observing and listening to user, the user can be easily found in the school. By observing what users will do during the using, it can easily found out the disability of the user and the bugs caused by the users. Moreover, after the role playing we can also receive the comment from the user directly.

Focus group

School is the good place which is easily to joint up the people together. However, it requires a skilled and experienced leader to lead the group in the discussion.

Environment Analysis

1. Where are they

Normally, the user should stay in the classroom and they are having the lesson when they are using the app. They can also use the app in everywhere like on the train or at home. For example, they can download the PowerPoint and element after lesson.

2. What is using

The user is using the smart phone or tablet for the lesson. They have to put their own electronic device on the desk. On point have to remind is that the students may be more easily discouraged by the message from other app during the lesson.

Task Analysis (HTA)

Login as user:

Open the app

Users open the app by clicking the app icon in the home page of their own electronic device.

Input the userid and userpassword

Users input their own userid and password in Moodle and click the login button.

Logout:

Click the logout button

User can click the logout button to logout from the system.

Confirm the logout

User click the confirm message.

Sending Message:

Select message

User clicks the message option in side bar.

Find the target user

User find the target.

Input the information of message

User input the message she want to send.

Join the lesson:

Select course list

User clicks the course option in side bar.

Select course

User selects the course that enrolled in the view.

Select lesson

After select the course, user can select the element in the course. Lesson can be joined.

Answer the question:

Join the lesson

User has to join the class first.

Answer the question

User can answer different type of question, like multiple choice etc.

Submit the answer

User clicks the answer and submits to the teacher.

Check the lesson Material:

Select lesson and Review the lesson

User selects the course that enrolled in the view. And after select the course, user

can select the element in the course. Lesson can be reviewed.

Switch the page

User can browse the lesson material and switch to different pages.

Check personal profile:

Select profile

User clicks the profile option in side bar.

Close the app:

Click the return or home button

User clicksreturn or home button to quit the app.

Moodle Server Side

Moodle is a learning platform that allows administrators, teachers and students to interact with each other like providing learning materials, quizzes for students to attempt or download so that the students would be benefited.

Moodle is constructed by various of plugins to perform different functionalities, for example, activities and resources, blocks are the most important plugin types in Moodle. In this project, an eLesson plugin would be built that is used in course to perform online viewing PowerPoint, and launching online lesson functions.

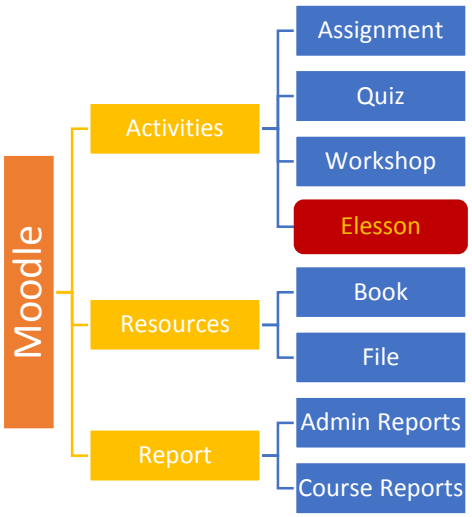


Figure 21Demonstration of Moodle system architecture, Moodle system consists of many kinds of plugins to create a learning environment. eLesson is one of the instances in Activities plugin

Functions of eLesson

eLesson takes lesson module as the prototype, it has a new page type which is "PowerPoint page" to provide web and mobile phones online, offlinePowerPoint functions, while it keeps all the original page types and features left so that the new page type could be integrated with existed page types to perform an interactive lesson.

The following paragraph would show how to use the eLesson in Moodle.

Firstly, you need to login to Moodle, make sure your account has the permission to create and manage course.

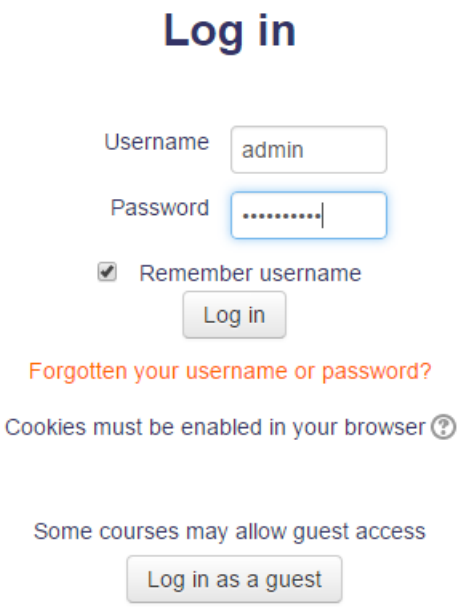


Figure 22Login to Moodle

Then, on the left side of the website, there will be a navigation menu and an administration menu, select “site administration”, and then select “Courses”, finally, choose “Manage Courses and categories”.

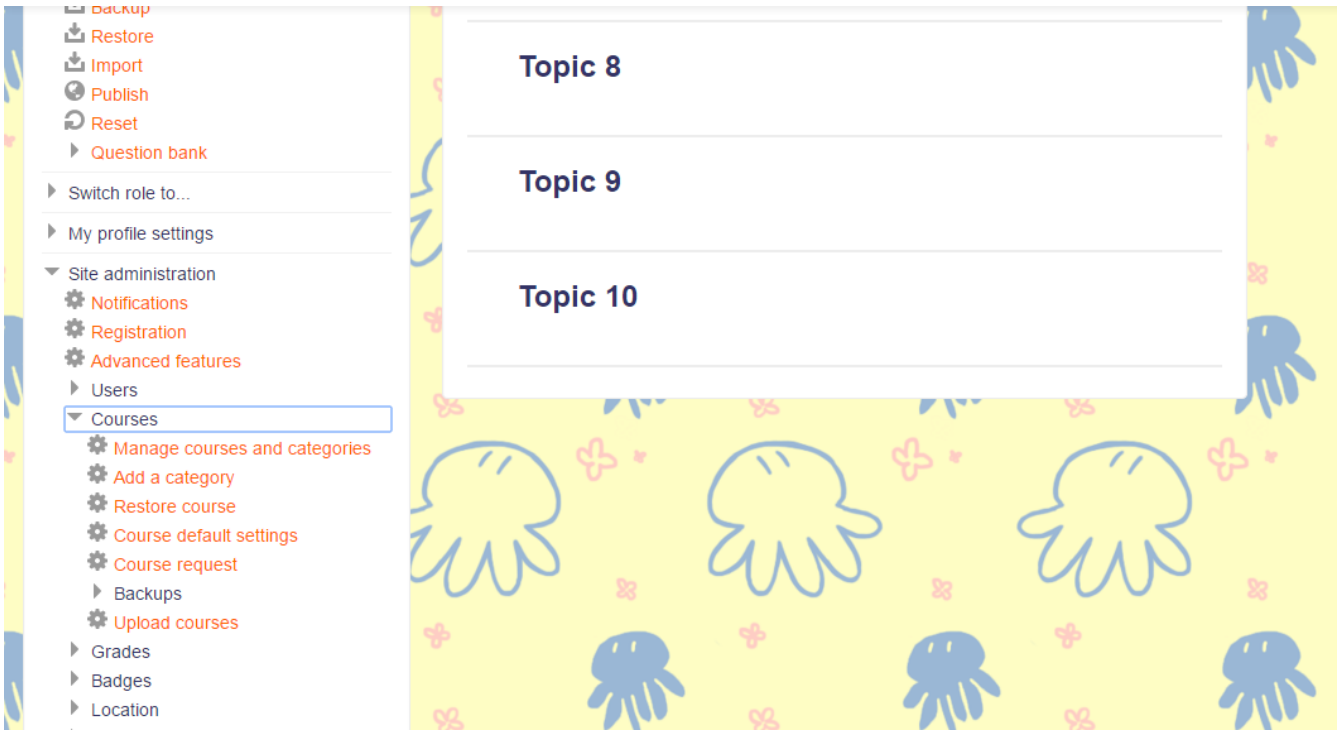


Figure 23Choose “Manage courses and categories” in the administration menu on the left side of the website

In course and category management, there are two things that could be added, the first one is course category, and the second one is course. Each course belongs to one course category. You could click “Create new category” or “Create new course” to create those things.

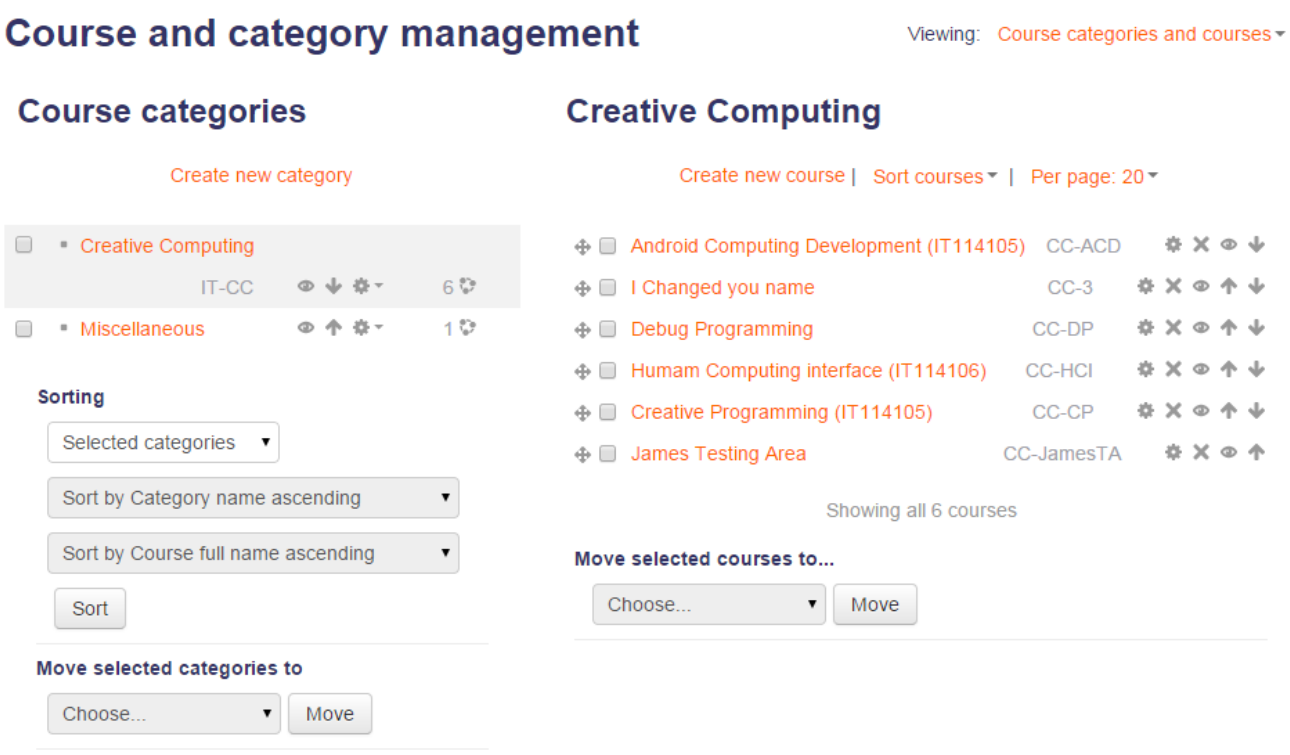


Figure 24Course and category management page, course categories and courses creation are done through this page

After creating a course, the course would be found in “My courses” in the navigation menu.

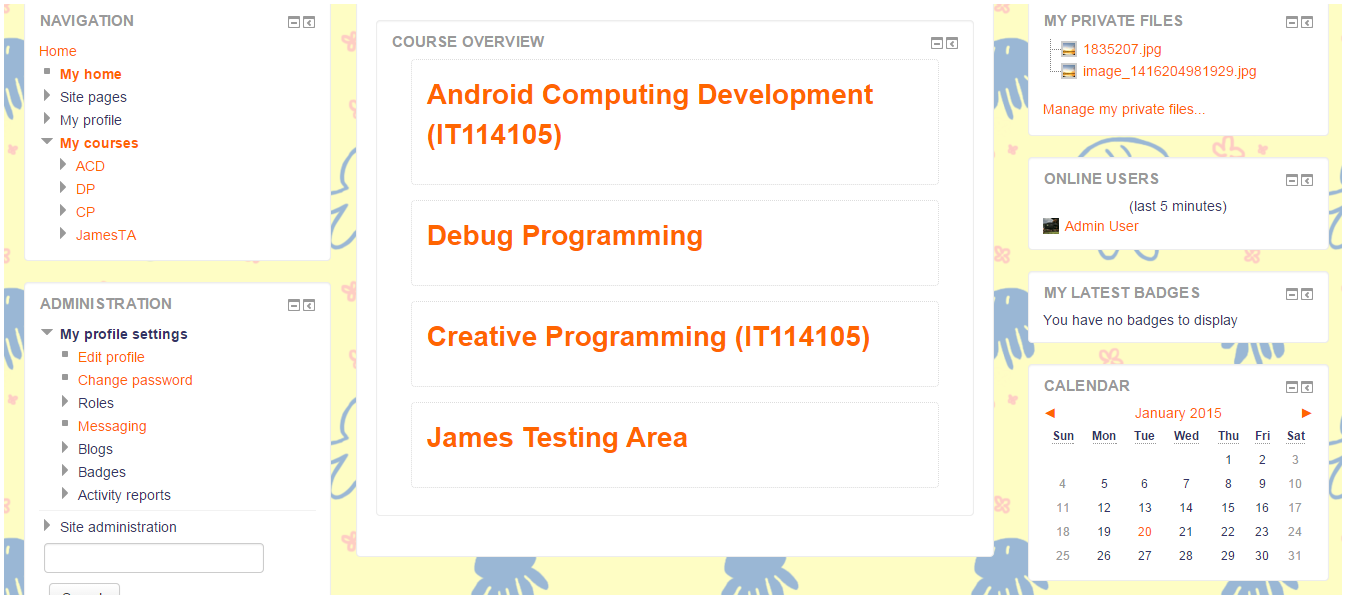


Figure 25Course overview, it shown all of the courses the account joined

If the account has the permission to create an activity or resource, there should be a "Turn editing on" button on the top right corner. Click the button and the edit mode would be enabled.

Figure 26Course page without edit mode enabled

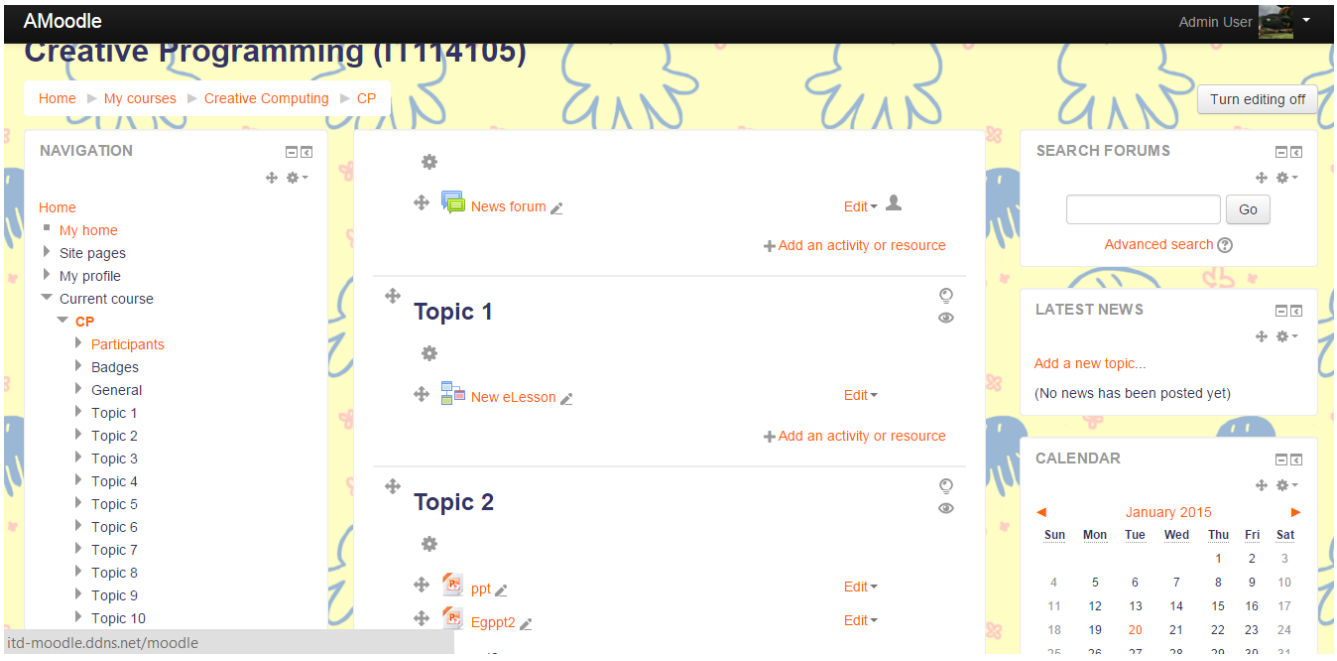


Figure 27Course page with edit mode enabled, you could now add or edit stuff in the course

To add an eLesson to the course, click "Add an activity or resource" and a window

would pop up, there are many activities or resources could be added, select “eLesson” and click “Add”. Then an eLesson would be added.

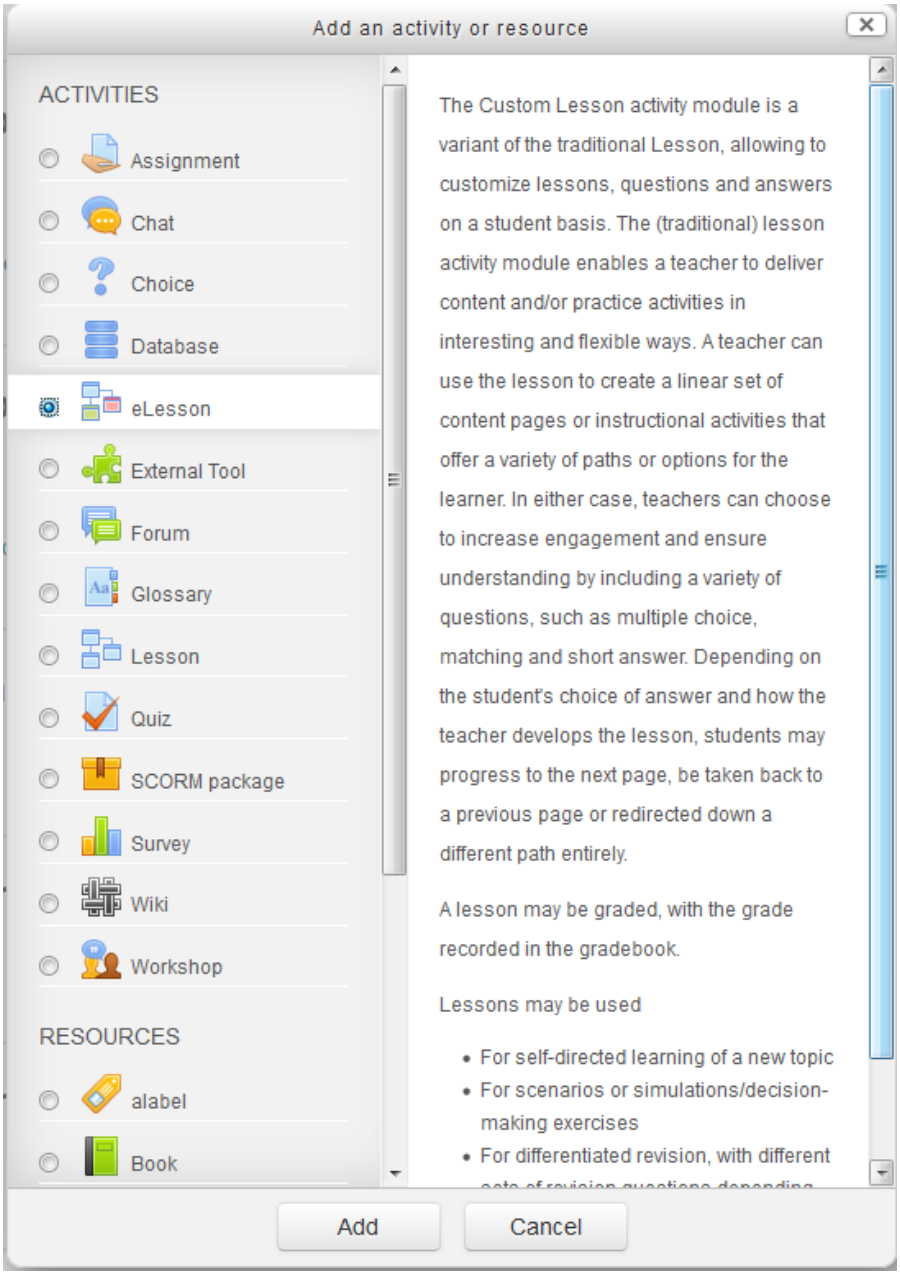


Figure 28Adding the eLesson module into a course, select “eLesson” and click “add”

After adding an eLesson, you could click into the elsson and add pages into it. If there is no page in the eLesson, system would ask you to add the first page.

Figure 29If the elsson has no page, system would ask you to add the first page

If there are already some pages in the eLesson, you could add new page in the edit tab.

Figure 30If pages existed in eLesson, add page function could be found in “Actions”

To add a PowerPoint page, you should choose “Add a PowerPoint”, and it will redirect to the PowerPoint page form.

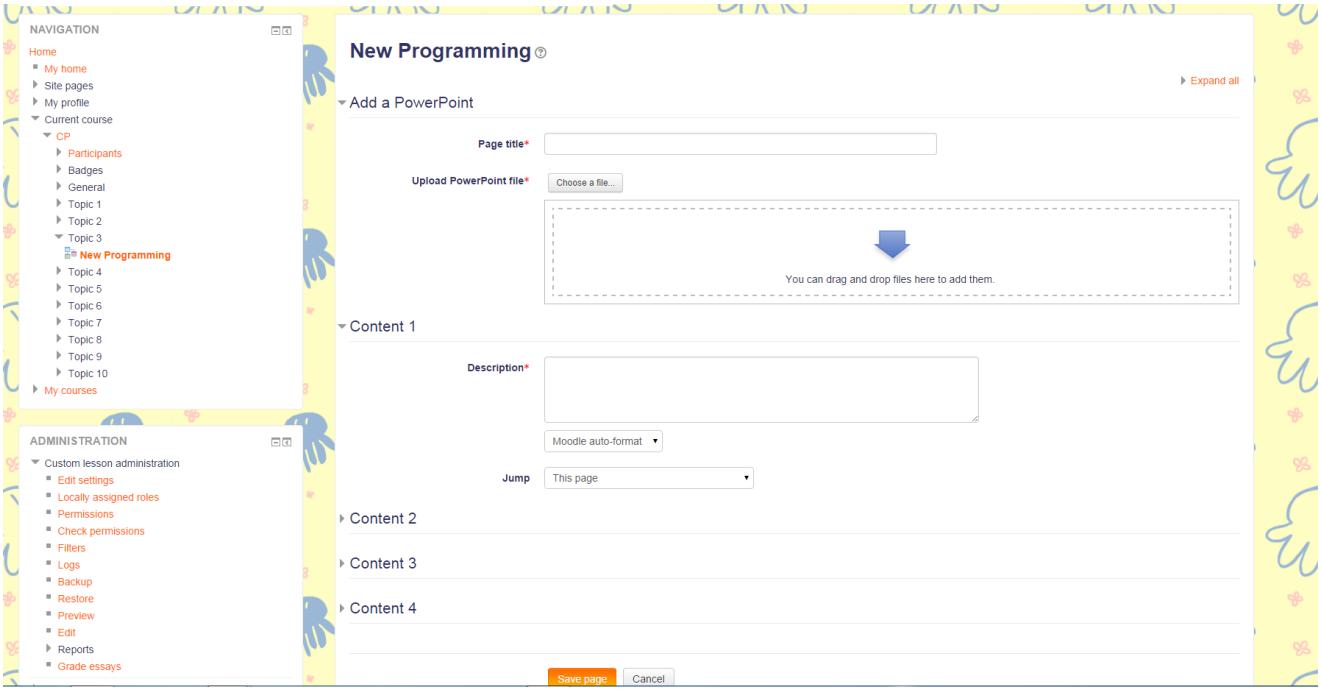


Figure 31The PowerPoint page form

To perform a PowerPoint page, teachers just need to upload a PowerPoint file and customize the PowerPoint page like setting content, which page would be jumped to etc. and the rest would be automatically completed. In the PowerPoint page form, there are three columns that must be filled. That is Page title, the PowerPoint file that to be uploaded, and Content(at least one). In uploading the PowerPoint, you could just drag the PowerPoint file from your local computer, or select the file in server using file picker.

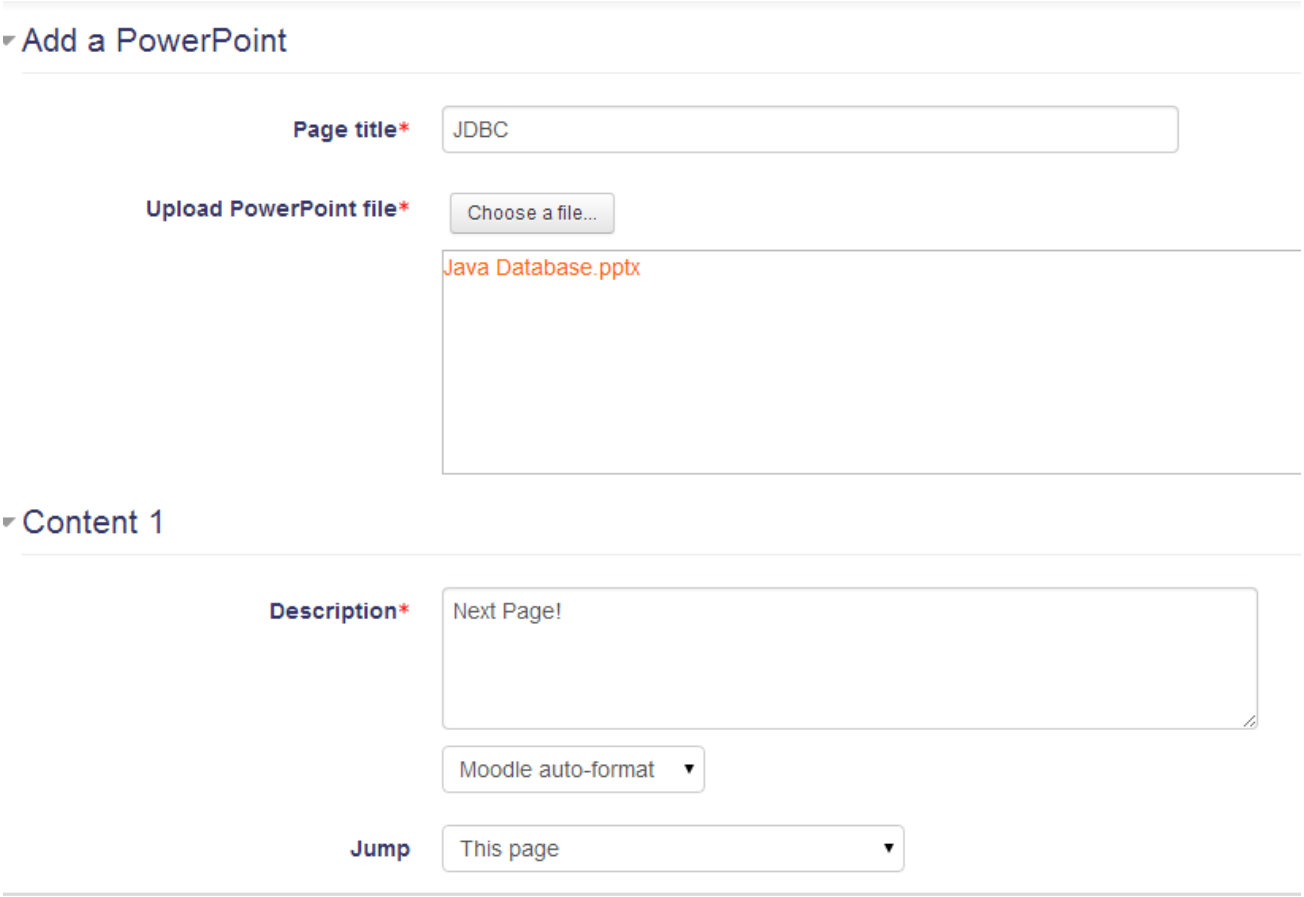
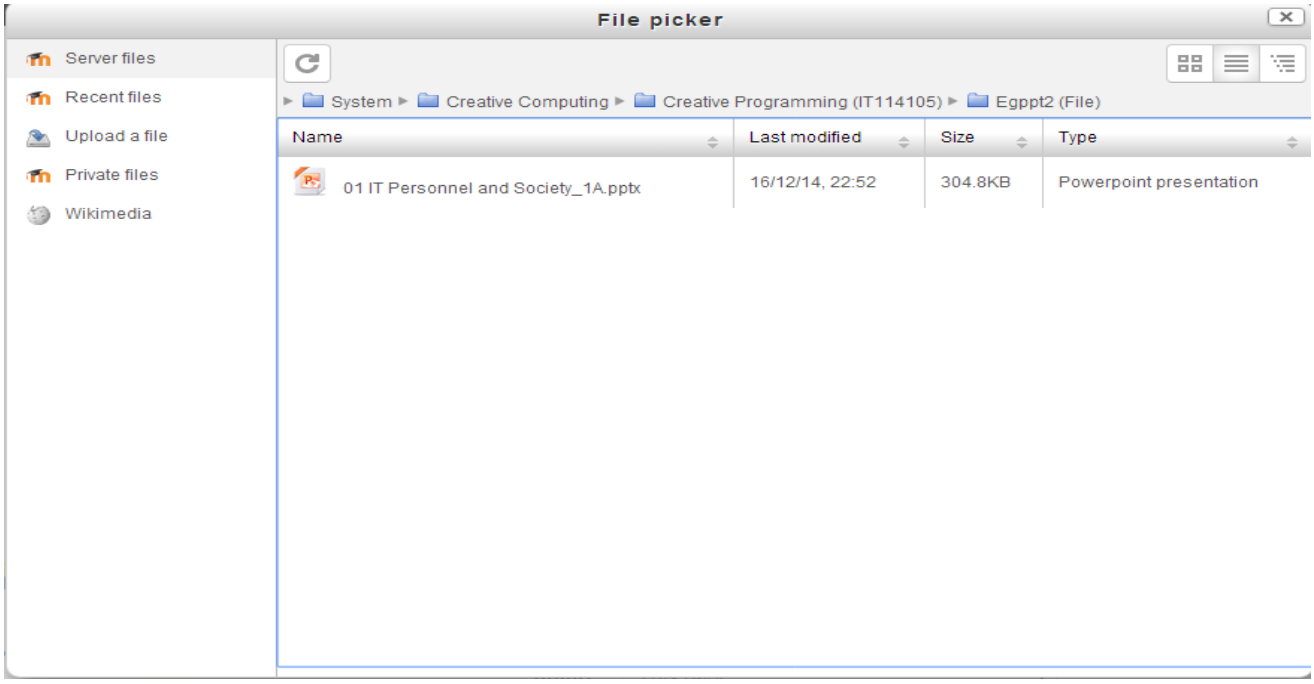


Figure 32Fill up the three necessary columns (title, PowerPointfile,content), and the rest would be



auto-completed

Figure 33If you want to upload a file but the file is not in the local computer, you could use the file picker to find the file in Moodle server

After adding PowerPoint page, the PowerPoint page could be found in “Preview” tab. You could see the page you just added. If the page is a PowerPoint page, you could change the slide of the PowerPoint by pressing “.”(next slide) and “,”(previous slide).

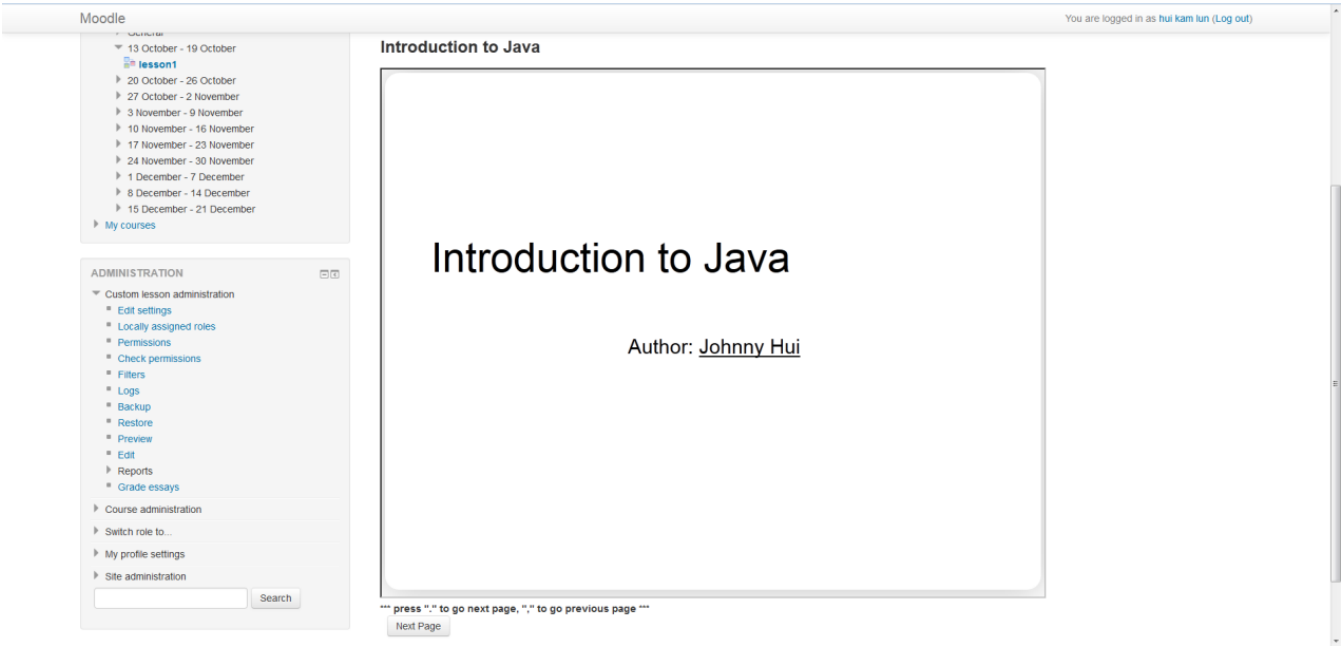


Figure 34Users could view the whole PowerPoint in Moodle website after inserting the page. If the page is a PowerPointpage,you could change the slide of the PowerPoint using “.” And “,” keys

If a PowerPoint page needs to be modified, there is no need to delete the original PowerPoint page, and add a new one. You could click the button with text “Edit page cotents” in the top right corner of “Preview” tab, or click the wheel next to that page.

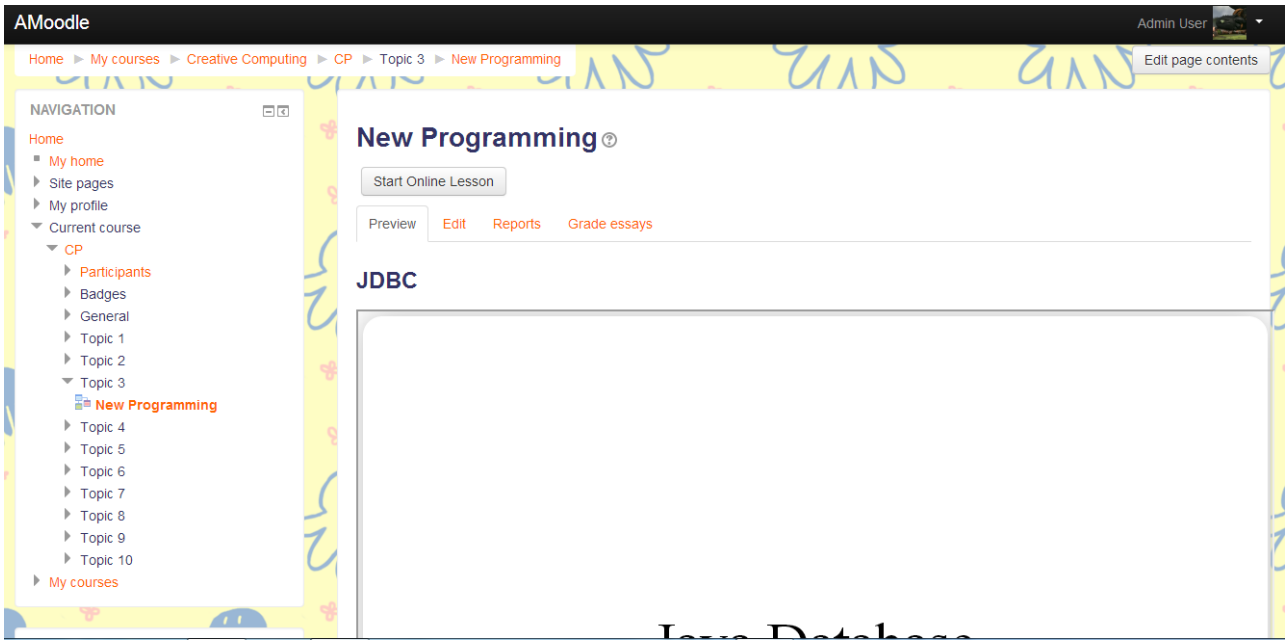


Figure 35Edit button in “Preview” tab(top right corner)

Figure 36Edit button in “Edit” tab(wheel)

Through adding different type of pages or questions, an interactive eLesson could be performed.

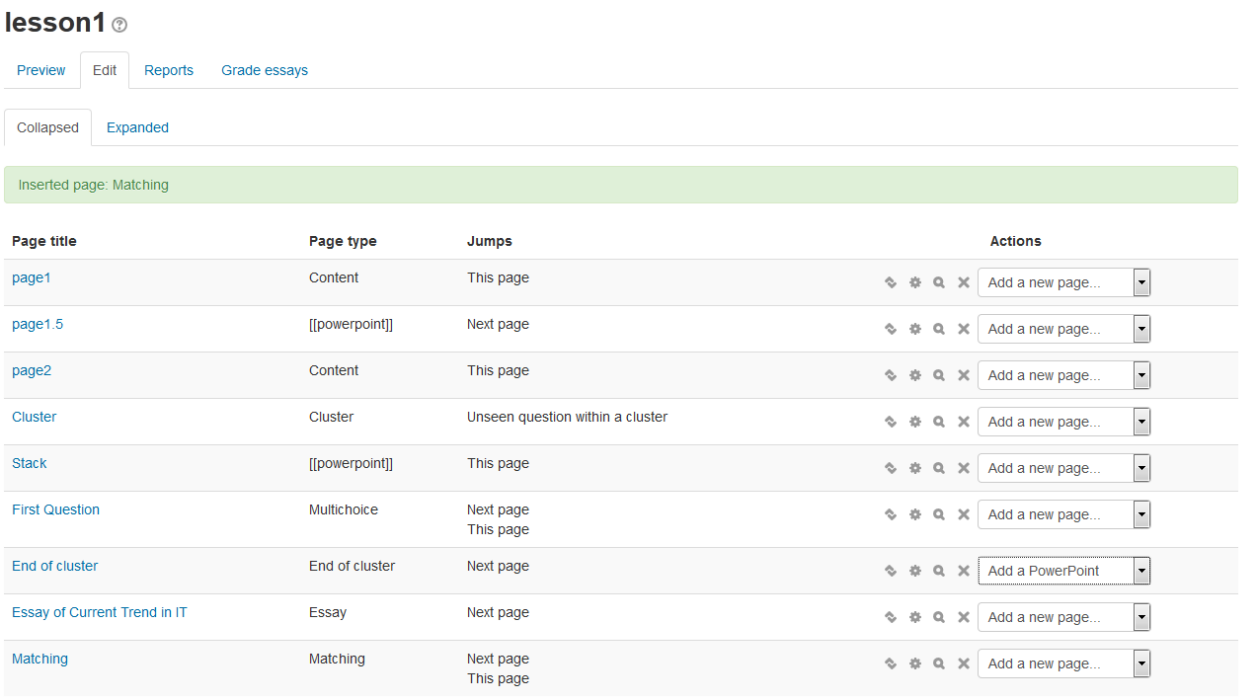


Figure 37Different pages could be cooperated together to perform an interesting lesson

Web Services in Moodle

In order to send data from Moodle server to mobile phone, it is necessary to use web services. Although there are already some web services existed provided by Moodle to get some basic information, it is still not enough for supporting eLesson module to perform eLesson apps, and online eLesson function. Hence, there are sets of custom web services were added in order to send specific data to mobile clients.

Getpagecontent_getpage:

getpagecontent_getpage

Get eLesson page content by ActivityID.

Arguments

activityid (Default to "1")

activity id of the module

```
General structure

int  Default to "1" //activity id of the module
```

```
XML-RPC (PHP structure)

[activityid] => int
```

```
REST (POST parameters)

activityid= int
```

Response

```
General structure

list of (
object {
id string   //page id
lessontitle string   //page title
lessonpagecontents string   //page contents
previouspageid string   //previous page id
nextpageid string   //next page id
qtype string   //question type
}
)
```

```
XML-RPC (PHP structure)


    Array
        (
        [0] =>
            Array
                (
                [id] => string
                [lessontitle] => string
                [lessonpagecontents] => string
                [previouspageid] => string
                [nextpageid] => string
                [qtype] => string
                )
```

Sample response:

[{"id":"5","lessontitle":"First step of programming","lessonpagecontents":"<p>echo \"Hello World!\";<\/p>","previouspageid":"0","nextpageid":"6","qtype":"20"},
{"id":"6","lessontitle":"Second step","lessonpagecontents":"<p>for($array in $key=>$value){<\/p><p>   echo $array[$key];<\/p><p>}
<\/p>","previouspageid":"5","nextpageid":"7","qtype":"20"},{"id":"7","lessontitle":"Is it true?","lessonpagecontents":"<p>Is Alan 18 years old?
<\/p>","previouspageid":"6","nextpageid":"0","qtype":"2"}]

Figure 18: This web services takes activity ID of the eLesson as parameter, return the information of all pages in that eLesson.

Getcoursedescription_getcoursedesp:

getcoursedescription_getcoursedesp

Get courses data by userid

Arguments

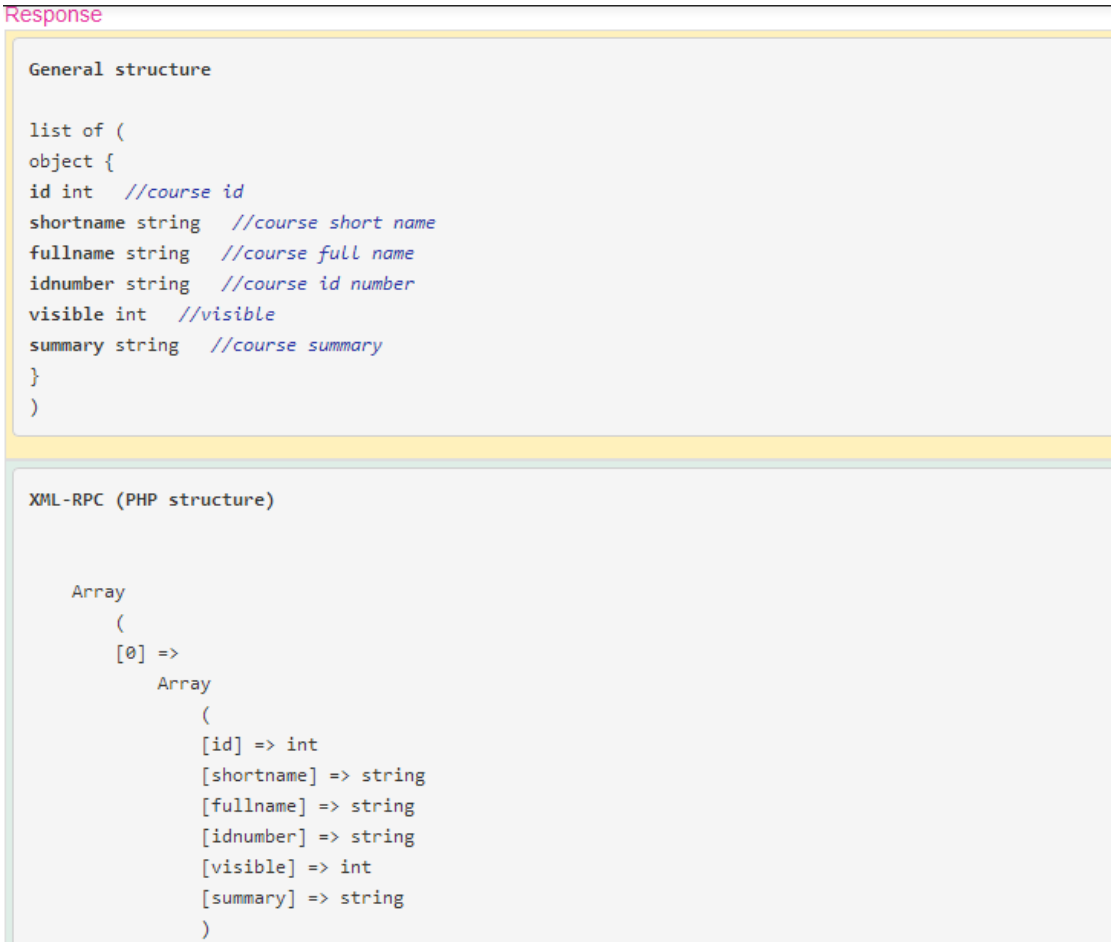
userid (Default to "-1")

user id

```
General structure

int  Default to "-1" //user id
```

```
XML-RPC (PHP structure)

[userid] => int
```

```
REST (POST parameters)

userid= int
```



[{"id":5,"shortname":"ACD","fullname":"Android Computing Development (IT114105)","idnumber":"CC-ACD","visible":1,"summary":"<span style=\"-webkit-text-size-adjust:
100%; background-color: rgba(255, 255, 255, 0);\"><b style=\"margin: 0px; padding: 0px; border: 0px; vertical-align: baseline; background-image: none; box-sizing:
border-box;\">Android<\/b> is a <a href=\"http:\/\/en.m.wikipedia.org\/wiki\/Mobile_operating_system\" title=\"Mobile operating system\" style=\"margin: 0px;
padding: 0px; border: 0px; vertical-align: baseline; background-image: none; box-sizing: border-box;\">mobile operating system<\/a> (OS) based on the <a
href=\"http:\/\/en.m.wikipedia.org\/wiki\/Linux_kernel\" title=\"Linux kernel\" style=\"margin: 0px; padding: 0px; border: 0px; vertical-align: baseline; background-
image: none; box-sizing: border-box;\">Linux kernel<\/a> and currently developed by <a href=\"http:\/\/en.m.wikipedia.org\/wiki\/Google\" title=\"Google\"
style=\"margin: 0px; padding: 0px; border: 0px; vertical-align: baseline; background-image: none; box-sizing: border-box;\">Google<\/a><\/span>
"},
{"id":8,"shortname":"DP","fullname":"Debug Programming","idnumber":"CC-DP","visible":1,"summary":"<h3>Debug Programming<\/h3>"},
{"id":3,"shortname":"CP","fullname":"Creative Programming (IT114105)","idnumber":"CC-CP","visible":1,"summary":"<h3>Creative Programming<\/h3><h3>|\u3002\ufe3f\u3002|
<\/h3>"},{"id":4,"shortname":"JamesTA","fullname":"James Testing Area","idnumber":"CC-JamesTA","visible":1,"summary":"<h3>Creative Object-Oriented Programming<\/h3>"}]

Sample response:

Figure 19: This webservices takes userid as parameter, return all the course information of that user enrolled.

Getanswer_getans:

getanswer_getans

Get eLesson answers of a page by Page ID

Arguments

pageid (Default to "1")

page id

```
General structure

int  Default to "1" //page id
```

```
XML-RPC (PHP structure)

[pageid] => int
```

```
REST (POST parameters)

pageid= int
```

Response

```
General structure

list of (
object {
id string   //answer id
pageid string   //page id
jumpto string   //jumpto option, -40=previous page, -1=next page, 0=this page
grade string   //grade
score string   //score
flags string   //flags
answer string   //text of the button, it could be answers for a question
response string   //response
}
)
```

```
XML-RPC (PHP structure)


    Array
        (
        [0] =>
            Array
                (
                [id] => string
                [pageid] => string
                [jumpto] => string
                [grade] => string
                [score] => string
```

```
                [flags] => string
                [answer] => string
                [response] => string
                )
        )
```

Sample response:

[{"id":"1","pageid":"1","jumpto":"-1","grade":"0","score":"0","flags":"0","answer":"Next Page","response":null}]

Figure 20: This web services takes pageid as parameter, return the page answer data of that page.

Entity Relationship Diagram (ERD) For Moodle

Since eLesson takes the reference from lesson module, database architecture of eLesson is almost the same comparing with lesson, so tables that are completely the same would be shown in this report.

In “eLesson_pages” table, in order to display the online PowerPoint in Moodle, there is a column called “htmlpath” was added to store the path of the html file.

Furthermore, there are two new tables are added in the database for supporting online eLesson, including “eLesson_joinedstudent” and “eLesson_onlineLesson”.

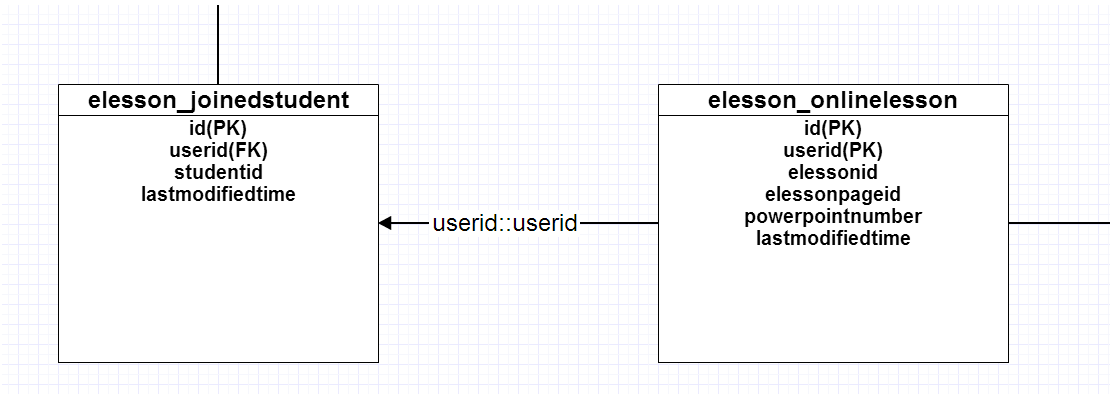


Figure 38Two new tables are added in eLesson, “userid” in “eLesson_joinedstudent” table takes the reference from “userid” in “eLesson_onlineLesson” table

Data Dictionary

eLesson_joinedstudent:

Column	Description
Id(Primary Key)	Id of the table(auto_increment)
Userid(Foreign Key→eLesson_onlineLesson)	Userid of the people who hold the online lesson.
Studentid	Userid of the people who attended the online lesson
Lastmodifiedtime	Last time that participants update their status.

eLesson_onlineLesson:

Column	Description
Id(Primary Key)	Id of the table(auto_increment)
Userid	Userid of the people who hold the online lesson.
eLessonid	Id of the eLesson that is launching.
eLessonpageid	Id of the current eLesson page.
PowerPointnumber	If the eLesson page is a PowerPoint page, this will contain the current PowerPoint slide number, otherwise it will always be zero.
Lastmodifiedtime	Last time that participants update their status.

IOS Application

The iOSMoodle application is developed in iOS8.1 version and xCode6.1, the application would be work well in this version. Because it is not tested in iOS7 or other version devices, so there may not be run successfully in those device.

The application use an orange color as the main theme of Moodleplatform.To makes the application more interactive and fun. We custom designed some image, icon and animation. That will let user be more tend to use our app instead of the Moodle website because we built a better user interface and more convenience to them. So that, user would not think bored or unlike to use our app even this is a e-learning purpose app.

Figure 39 App Icon

This is the ourself-designed application icon. It use the main theme of orange color and long shadow design style. The application name “MoodleDemo” is just for demonstration purpose, it will be change in the final version.

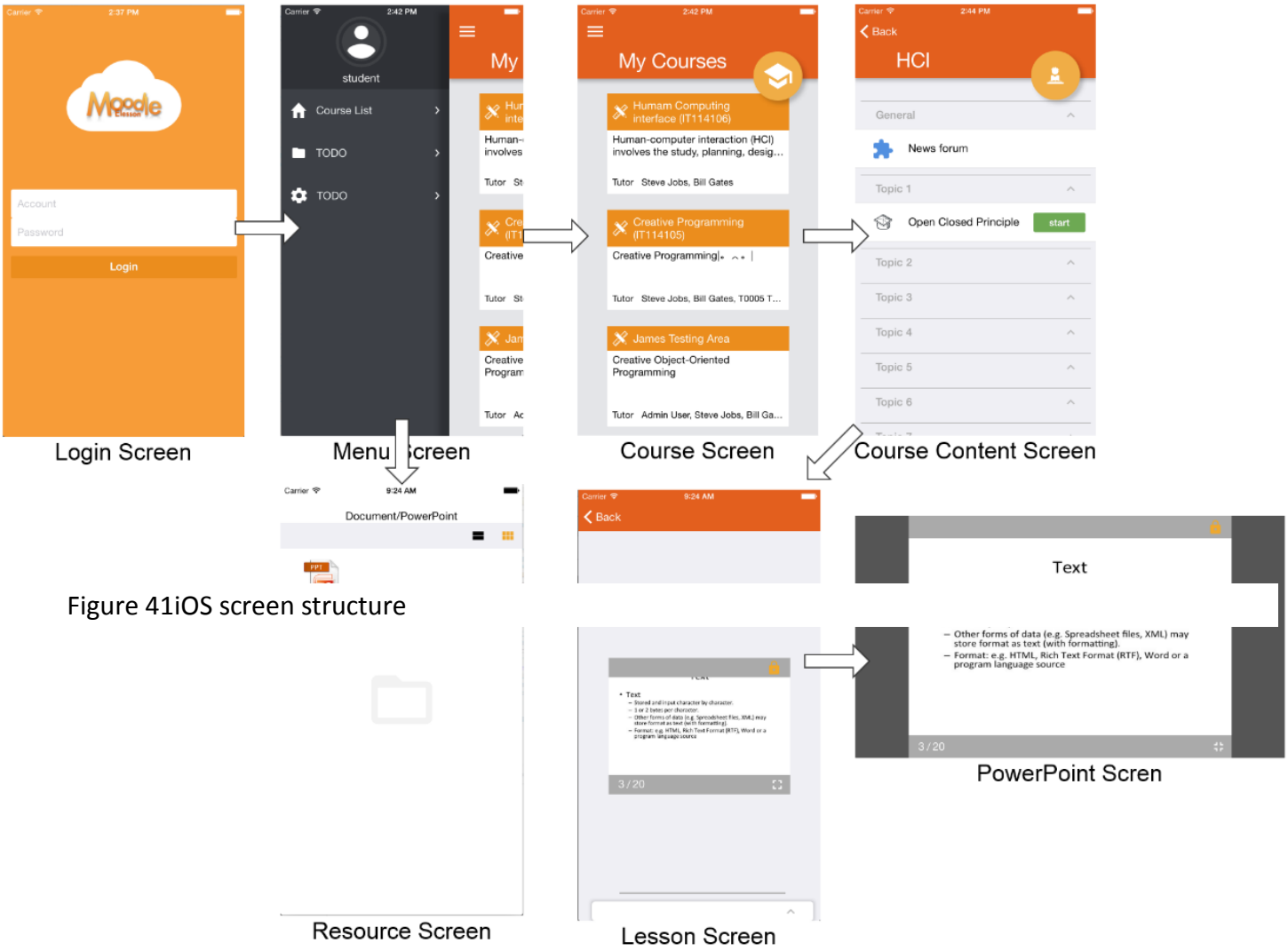


Figure 41iOS screen structure

Figure 40iOS screen structure

Application Structure

Here is the structure and navigation of the iOS application. The first screen is a login screen, user are require to login in order to use our app. The main screen navigation is controlled by the side bar. That will direct user for different function pages like the course menu, file menu and setting menu. So far, the application is just on the progress of implementing functions of course menu. When we tap the course list menu, the application will direct us to the course screen. Finally, when we tap once of the course, we can see all contents of the application. Lesson screen shows the eLesson details. PowerPoint screen shows a full screen PowerPoint in landscape mode.

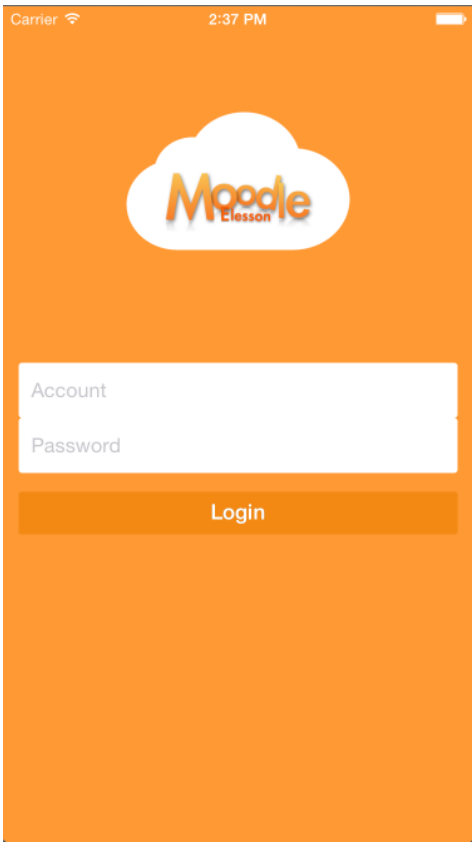


Figure 42 iPhone6 login screen

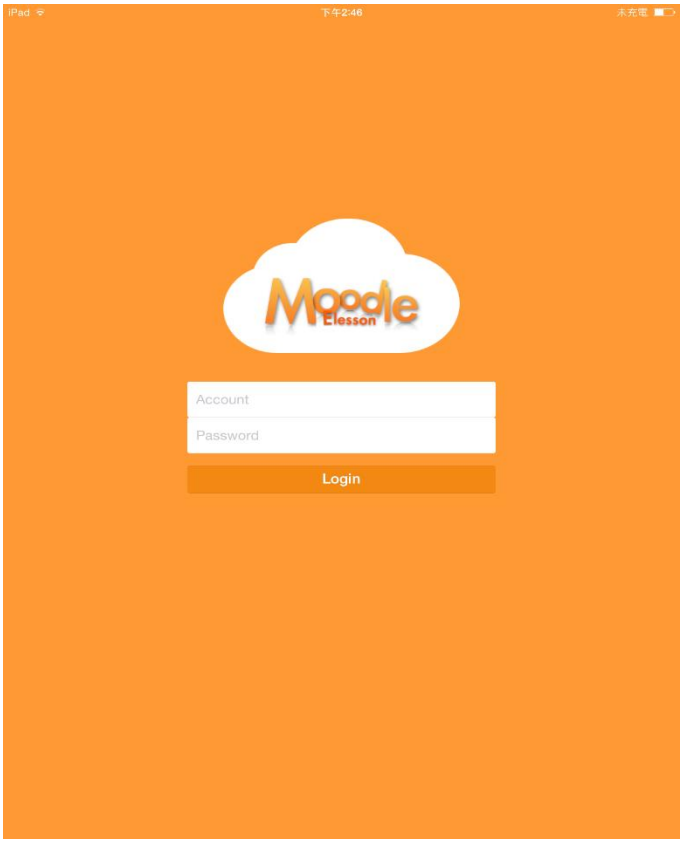


Figure 43iPad login screen portrait

Login Screen

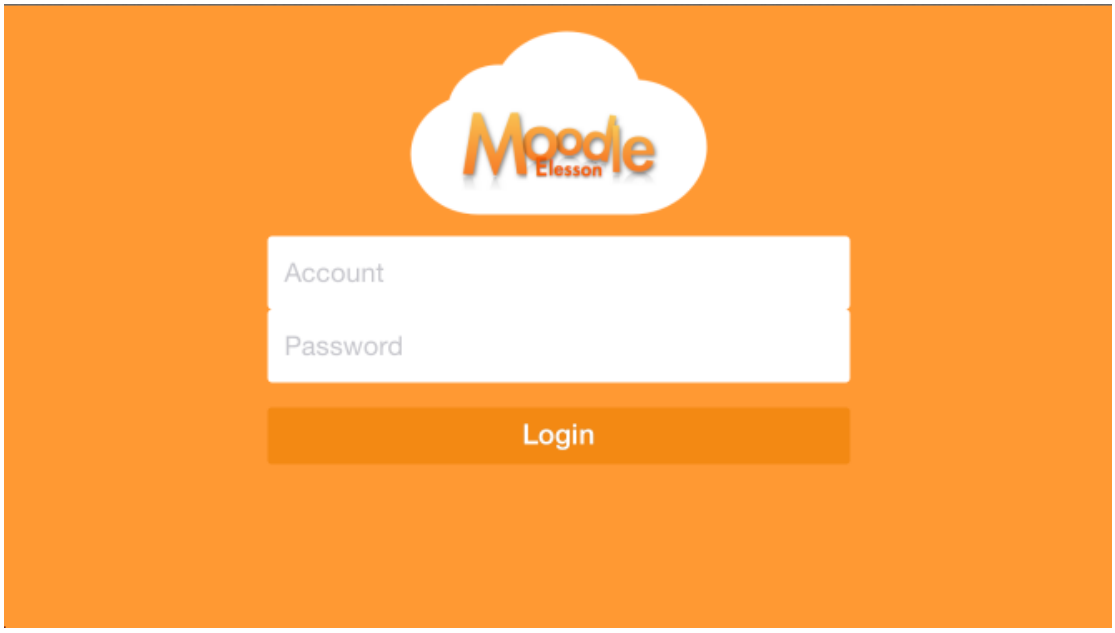


Figure 44 iPhone6 login screen landscape

As you can see above, it is the login screen of our application. There include a self-designed logo, a cloud with the text “MoodleeLesson” and a login form. It is a clear and simple design.

The application is also support for both iPhone and iPad. Components would be auto layout themselves in different size of screen. It use the layout technologies from Apple which called Auto Layout and Size Classes. So that, one source can develop for both iPhone and iPad, there is no need to re-write codes for different device.

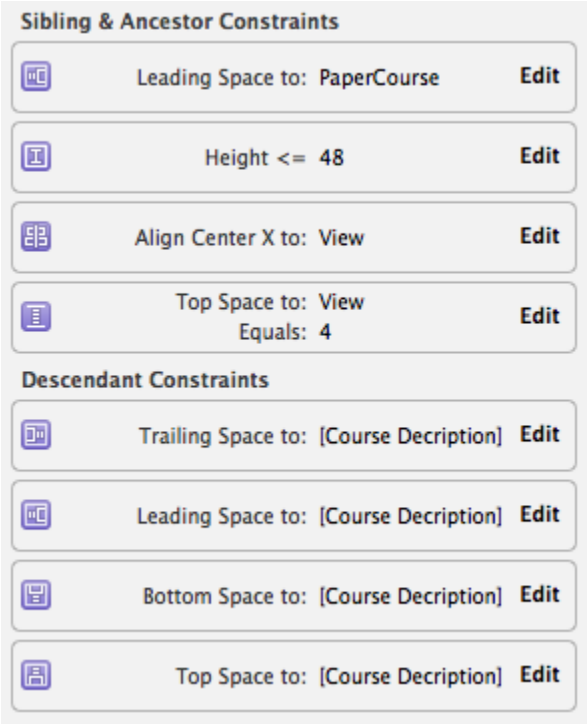


Figure 45 auto layout

Auto Layout is a layout technology developed by Apple. It cans layout things based on setting constraints. For example, setting the maximum width and height, the property that the constraint will go to conformed, the spacing of two components, the scale and alignment. The application can be automatically calculate the best

layout for different screen. Figure 5 is the example of constraint set in the course label.

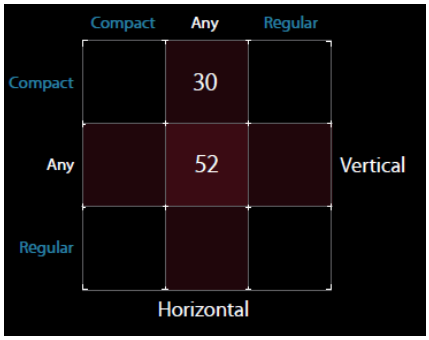
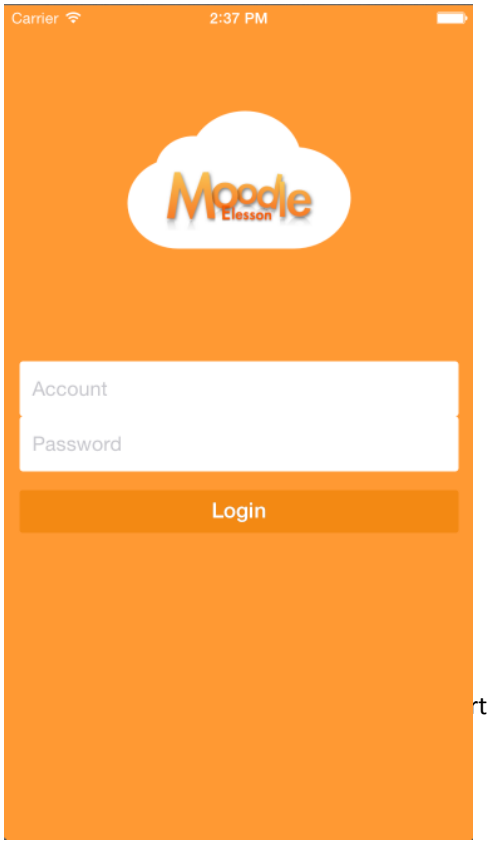


Figure 46 Size Class reference from apple WWDC 2014

The application also uses the concept of Size Classes. Size Classes allow developers to set up different constraint on different Size Classes. There's three size classes, they are Compact, Any and Regular.They are representing different device. A constraint that set in Any, Any that represents the width and height of the Size Classes would be considered by the application in any devices.

With auto layout and size class the application can be fitting into different devices in dimensions. That is convenient for the development and testing. Therefore, our app uses this layout technology.



Login screen

When user clicks the login button without entering account or password. The whole login form would be shaking in horizontal way a few

times. This design is reference from the mac login screen. It is a great way to inform user that there is something wrong.

Figure 47 login screen

When user entered required details and hit login button, the application would show an activity indicator replacing the origin login text to inform user the login process is now in progress.

Figure 48 login button

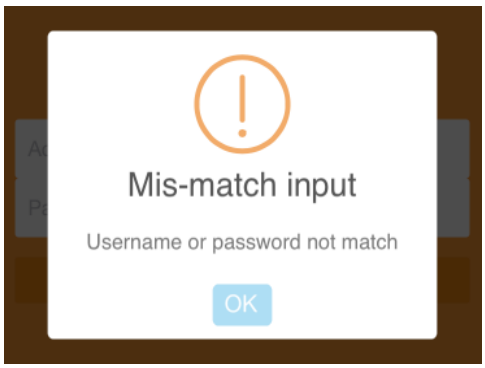


Figure 49 Alert box - warning

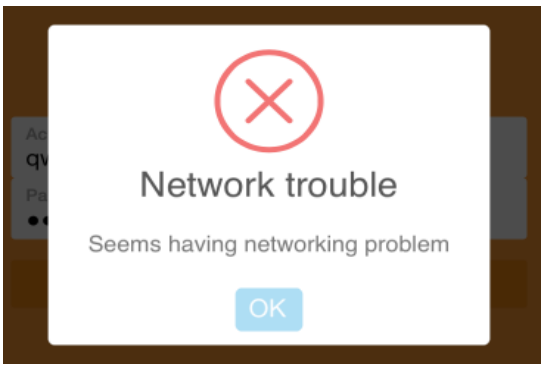


Figure 50 Alert box - error

When user entering wrong password, the networking status is not available or any other possible failure in server, there will be an alert box showing the error.That is an open source module implemented by developer in github.

Figure 51 Login form

The text field of login form has a floating function. The placeholder will be float upward when user is editing.

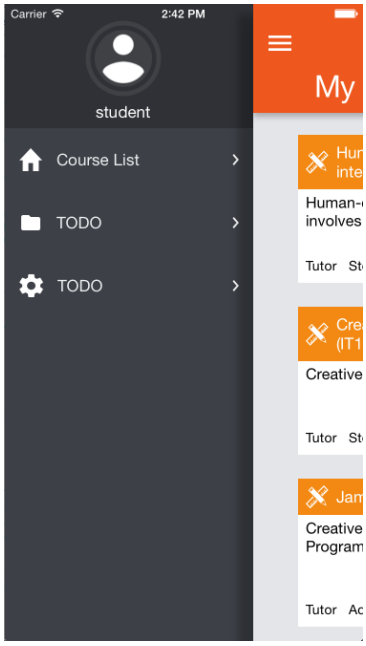


Figure 53 side bar portrait

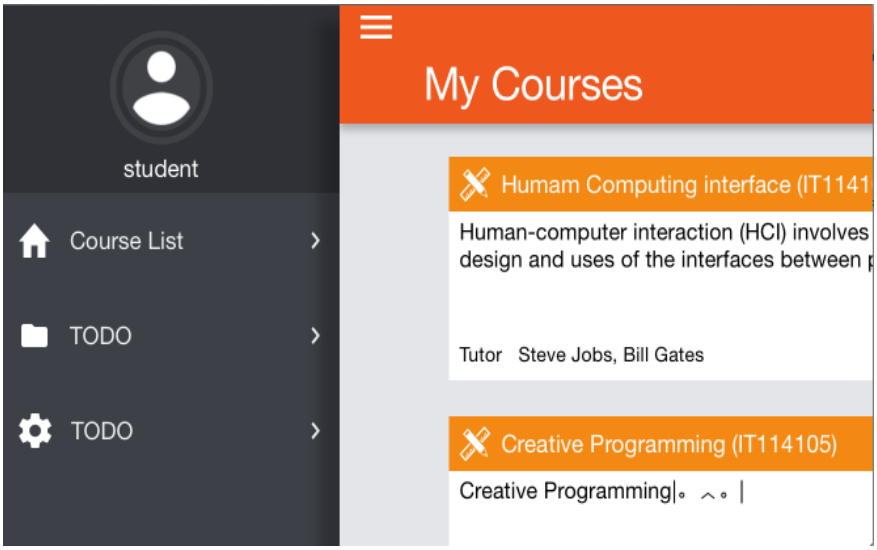


Figure 52 side bar landscape

Side bar

The side bar will be the main navigation of the whole application, on top of the side bar, there is a user icon and user name showing the details of current user. The user icon and user name is gained from server by web service.

Under the user information area is a table view showing the menu. The file menu and the setting menu are not implemented. If user taps the menu button, the right view will be change to corresponding detail page.

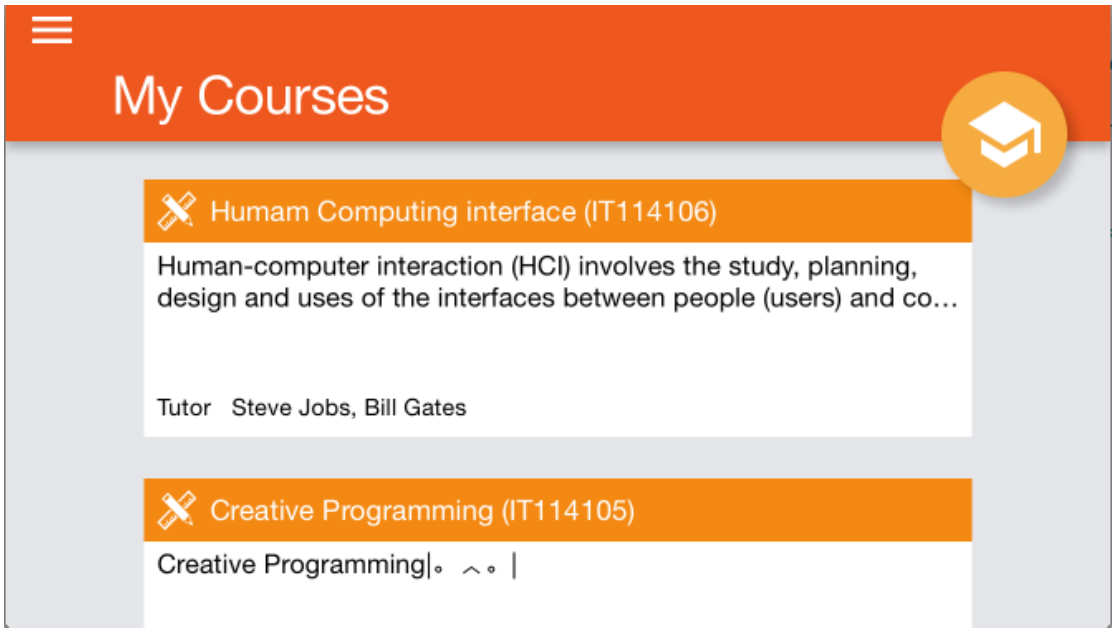
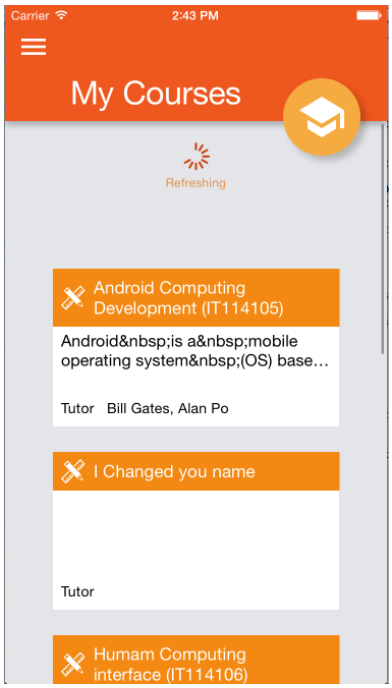


Figure 54 My Courses table

Figure above is the course view showing available courses that user enrolled in to the user. Each course is displayed in a paper material style. There is the course description and the tutors of that course.



The Course table also implement with a pull to refresh function, if there is something change in server side, pulling the course list would be re-fetching the data from server and update to the view.

Figure 55 pull-to-refresh

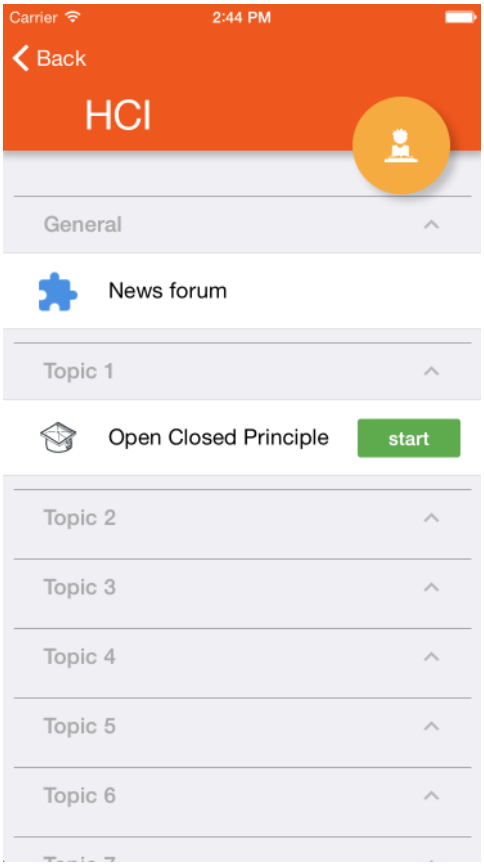


Figure 56 Course contents View

The figure above will appear when user tap one of a user course. The title HCI is a short name of human computer interaction. There is also a table showing the all course details as the website shown. The VTCeLesson module is shown is header Topic 1. This is the main module that we are working on. User can join into the lesson when they press the start button. Other modules shown as normal, but we are not focusing to interact with them in this moment.

Figure 57 expandable table header

The table header can be expand or collapse the all modules which is the contents of that section header by tapping the arrow at the right side. At the moment, the arrow will be rotated 180 degrees indicating the user whether it is expanding state or collapsing state.

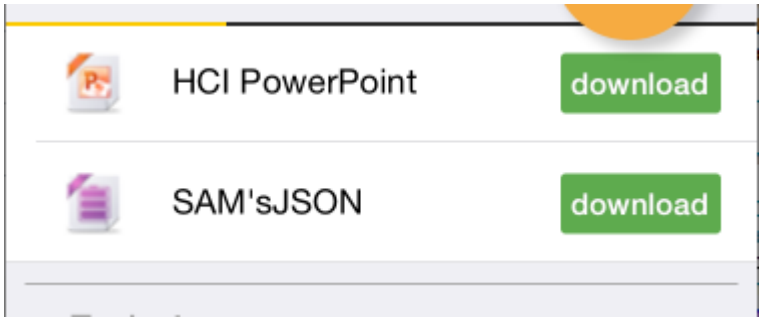


Figure 58 download resource

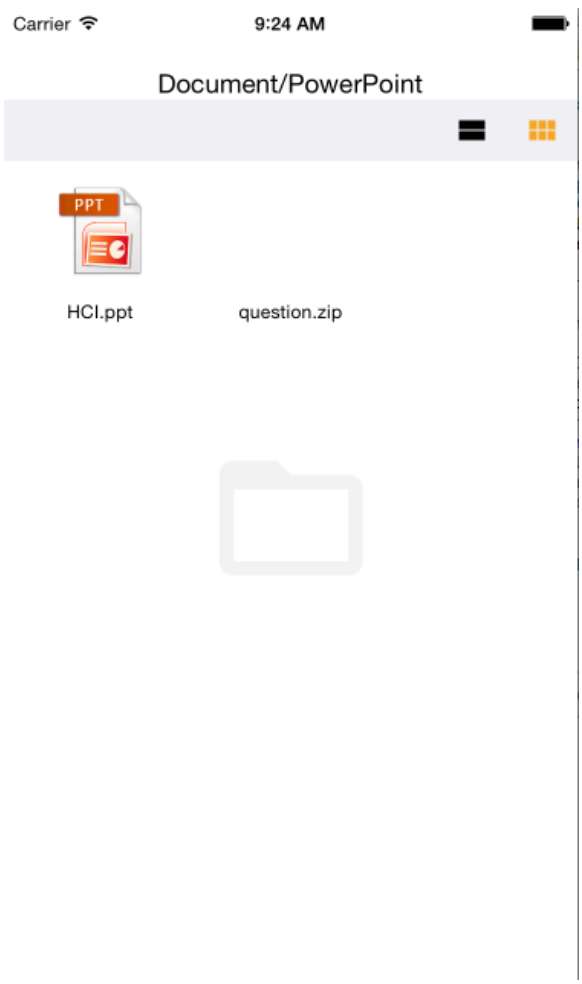


Figure 59 Resource Screen

The figure above shows thatuser can download the resource and the download process in shown upward the table view cell.

The figure on the left shows the resource screen. Resources downloaded would be shown in here.

By default it is in grid view.

It can also be stream view by clicking the button on the top-right corner.

Figure on the left shows the Lesson Screen, the lesson have a PowerPoint on the middle. The questions can be drag upward on the bottom at the screen.

Figure 60 Lesson Screen

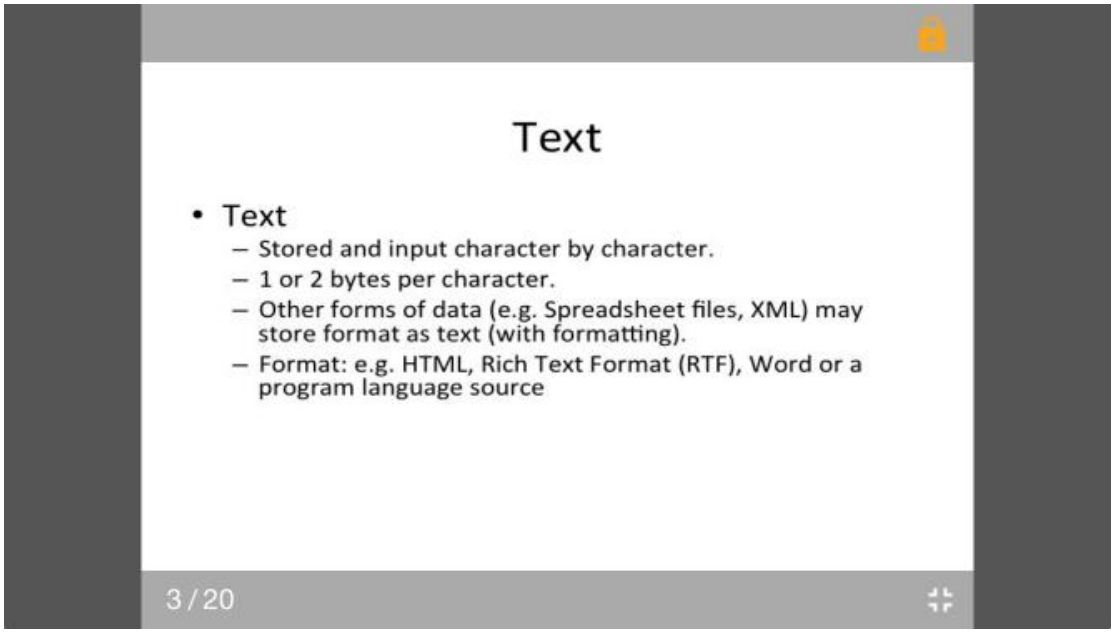


Figure 61 Full Screen PowerPoint

Figure above shows the PowerPoint in full screen mode. It contains two tool bar on the top and bottom which provide functionality of shows the page number, zoom out or in, tracking to teachers page.

How the mobile application communicate with server side

The mobile application is communicatingwith server by web service. The server has been written some web service which enables multiple protocols like REST, XML-RPC. Mobile application calls the web service by a request and passing in required parameters. For example, a login request will requires user name and password as parameters.

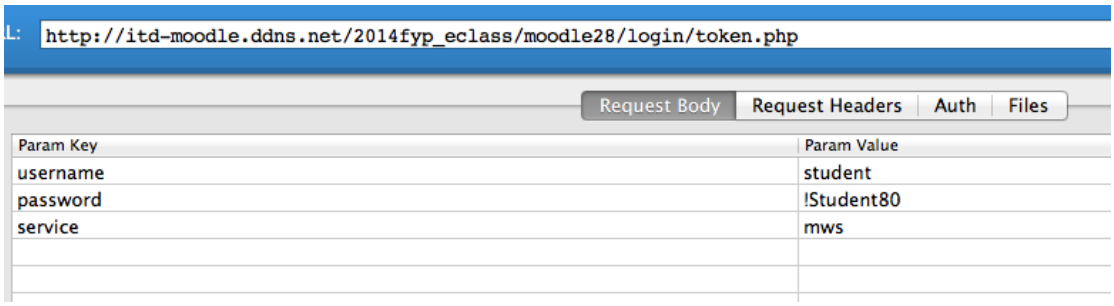


Figure 62 login request

After that, the server will return an authorized token allowing the application to call other web services and passing in the request as an authorization.


```
{
  "token" : "280cdb818740f0696297a35f1ea730f7"
}
```


Figure 63 returned Token

The figure below is the core web service built in Moodle.

Figure 64 web services

In iOS (Figure 63), we use Alamofire which is an open source HTTP request module to handle the web service request. In the onCompletion callback, the request is done and passing back four parameters, request, response, JSON and error. If there are errors occur during the request, the error parameter will contain the request status code and message of a HTTP request header, like a 200 status code is a success request. By checking the JSON data and the error parameter, it can be checking the status of the request and make suitable response.

```
Alamofire.request(.POST, SubStruct.domain + SubStruct.tokenUrl, parameters: ["username": "\(username.text)",
    "password": "\(password.text)",
    "service":"\(service)"]
).responseSwiftyJSON { (_, response, JSON, error) in
    if let t = JSON["token"].string{
        System.token = t
        var user :User = User.sharedInstance
        user.username = username.text
        user.token = t
        onCompletion(nil, nil, true, [])
    }else{
        if error != nil{
            SweetAlert().showAlert("Network trouble", subTitle: "Seems having networking problem", style: AlertStyle.Error)
        }else if let e = JSON["errorcode"].string{
            SweetAlert().showAlert("Server Error", subTitle: e, style: AlertStyle.Error)
        }else{
            SweetAlert().showAlert("Mis-match input", subTitle: "Username or password not match", style: AlertStyle.Warning)
            username.text = ""
            password.text = ""
        }
        onCompletion(nil, nil, false, [])
    }
}
```

Figure 65Alamofire Request

Figure 20 is the source code of using Alamofire to calling the web service and handling the JSON data from the block callback.

When the request is made, the application can get the JSON format response. Figure 64 is an example of returned JSON data. The application is now retrieving required data from the JSON and saving the JSON data into the modal class. When the view controller needs the data to display the contents, it can be got from the modal class and build the interface. It use the Modal View Controller(MVC) programming structure as normal as an usual iOS application did.


```
[
  {
    "enrolledusercount" : 63,
    "idnumber" : "CC-ACD",
    "id" : 5,
    "visible" : 1,
    "shortname" : "ACD",
    "fullname" : "Android Computing Development (IT114105)"
  },
  {
    "enrolledusercount" : 1,
    "idnumber" : "CC-3",
    "id" : 7,
    "visible" : 1,
    "shortname" : "TIANC",
    "fullname" : "I Changed you name"
  },
  {
    "enrolledusercount" : 3,
    "idnumber" : "CC-HCI",
    "id" : 6,
    "visible" : 1,
```


Figure 66 JSON data of course list

Here is an overall diagram of how mobile and server communications.The mobile client and Moodle service are communicated with the web services that built in the Moodle database and setting enable to the mobile device. In doing so, mobile client can call the web service function, basically PHP function and retrieve needed data from the database.

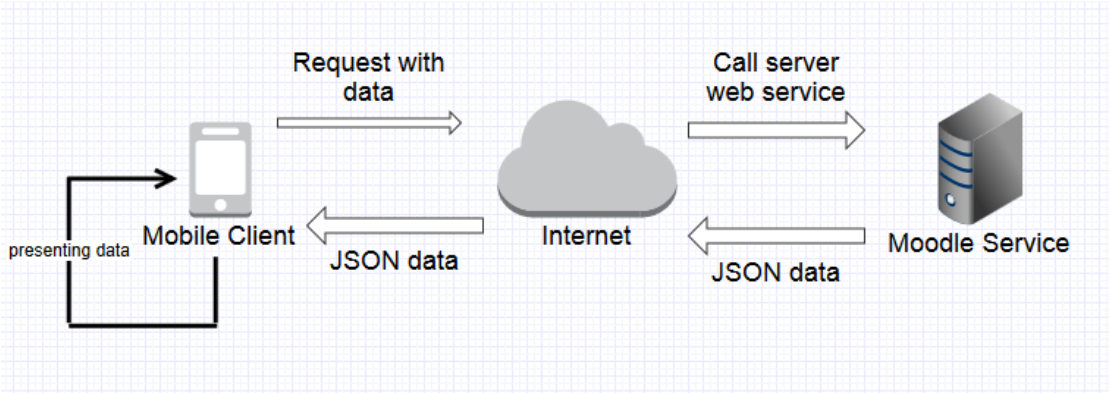


Figure 67 Mobile Server Communication

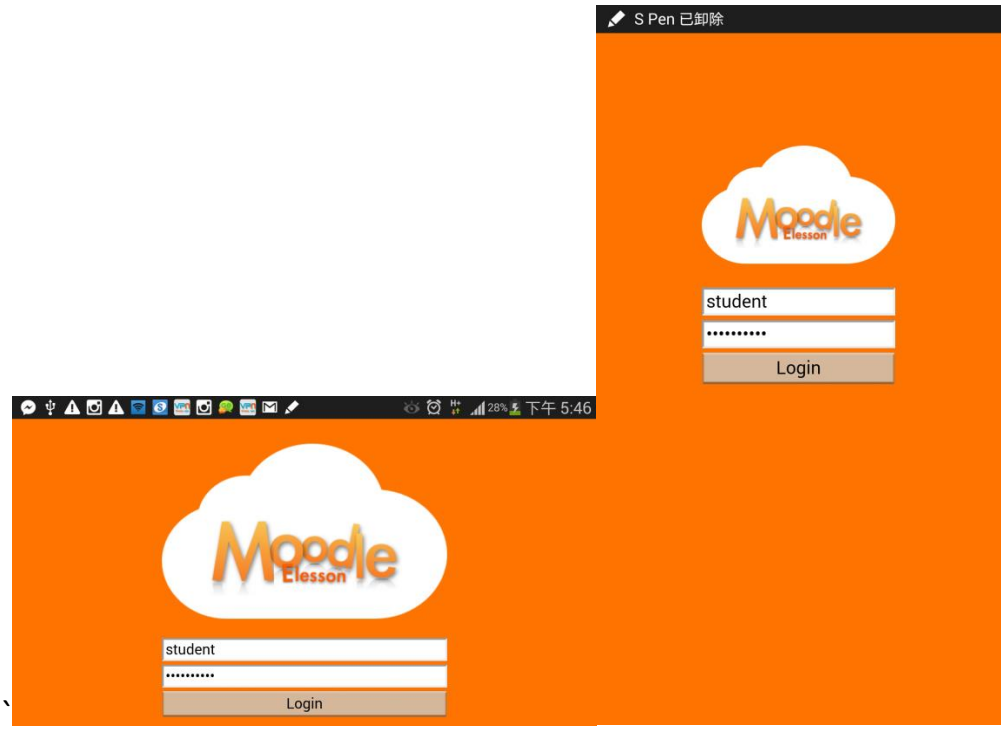
Android Application

Introduction

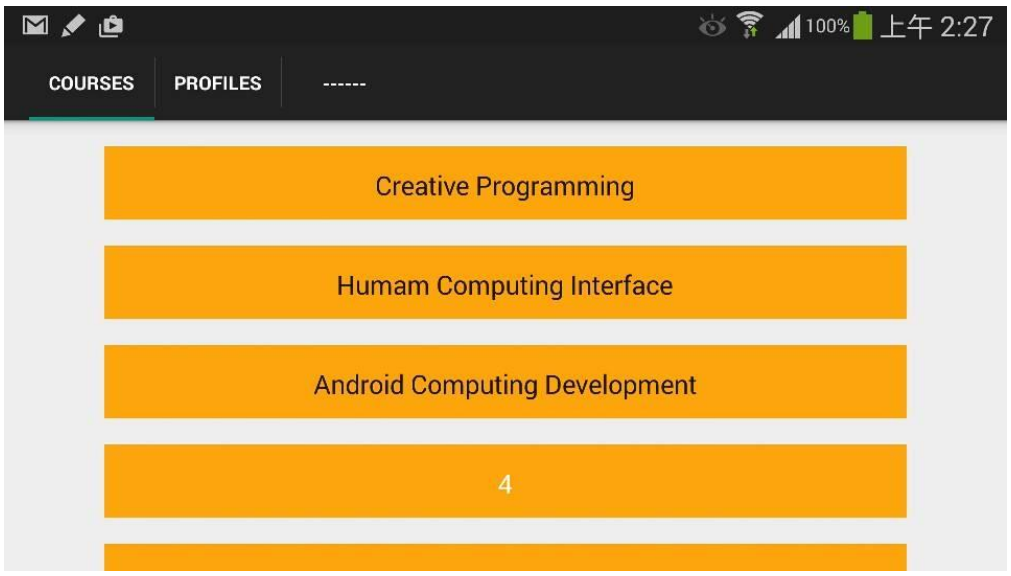
The android application of Moodle allow user to usethe function of the Moodle in android system. The project mainly focus on theeLesson , which is the Moodle plugin for the teaching . Mainly the project focus the interaction between student and teacher.Teachers are allowed to create the class in the website Moodle and the students receive the material. Android application is the client side that calling the web service of Moodle and reply the required data for the website Moodle. Moreover it is expected to fulfill the needs of students.

Android UI design

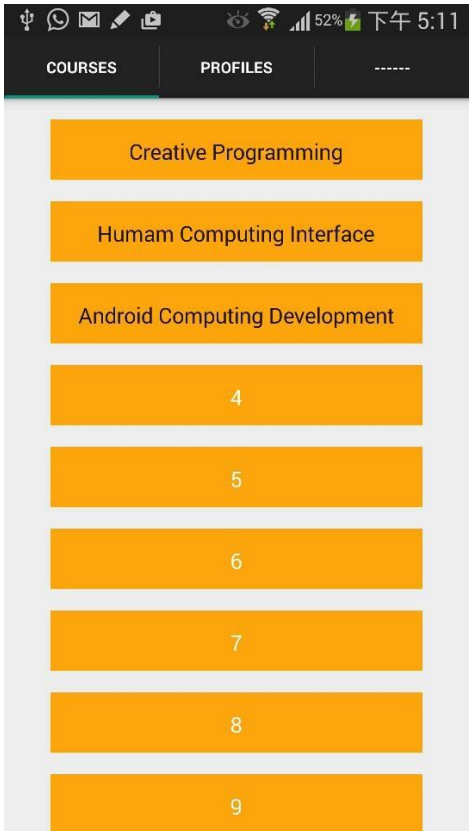
Main menu



Course list

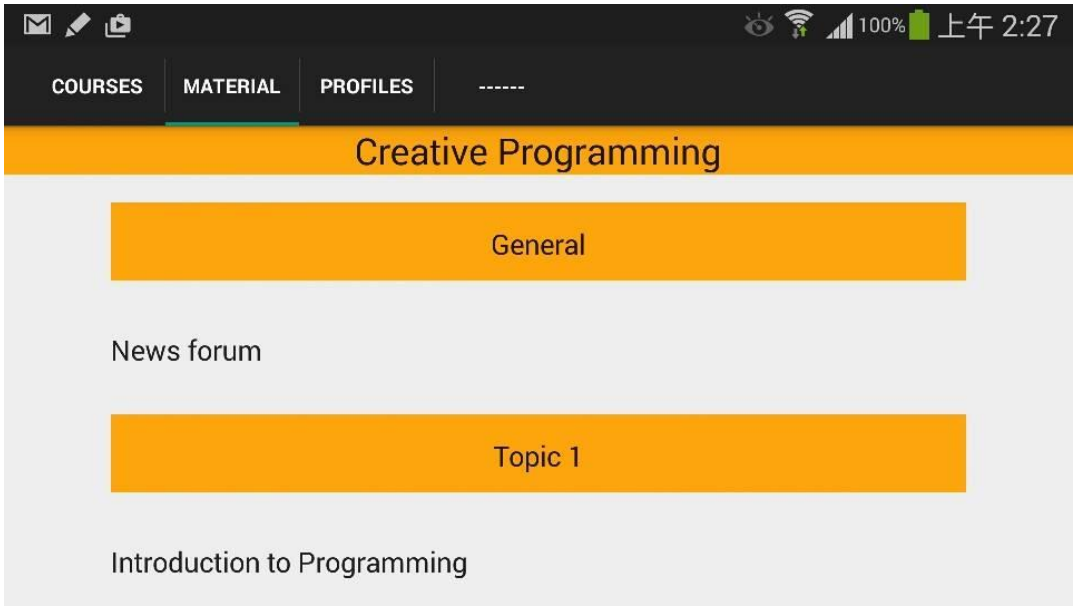

100% 上午 2:27
COURSES
PROFILES

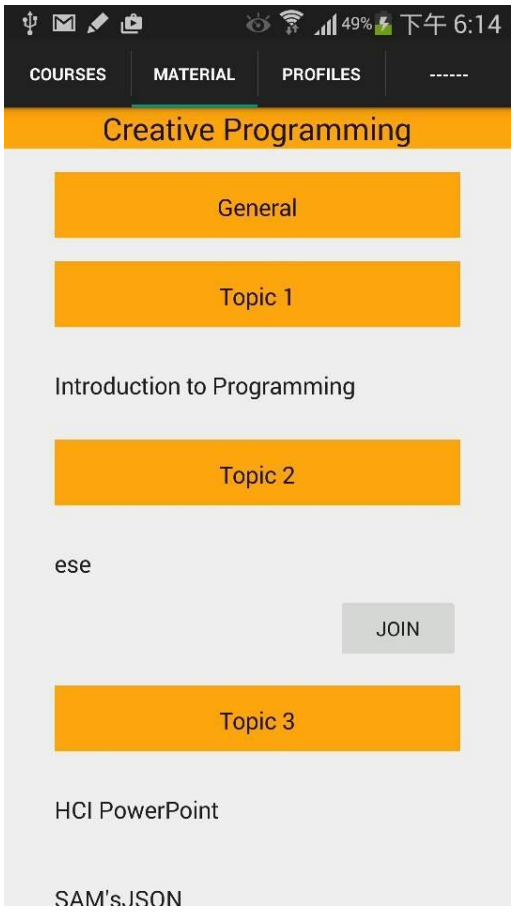
Creative Programming
Humam Computing Interface
Android Computing Development
4


52% 下午 5:11
COURSES
PROFILES

Creative Programming
Humam Computing Interface
Android Computing Development
4
5
6
7
8
9

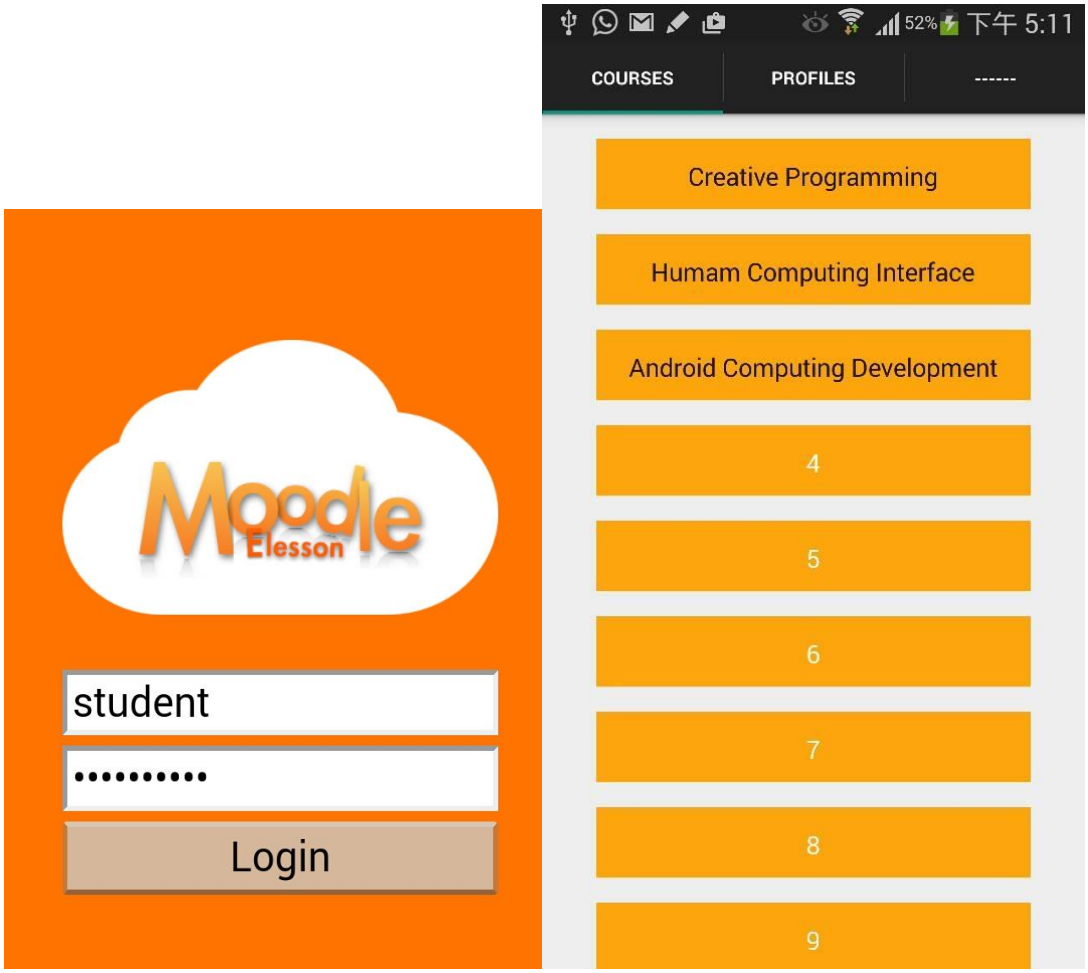
Course's lesson





Functions for Android app

Login & Logout



User (Student) can input their Moodleuserid and user password to login to system.

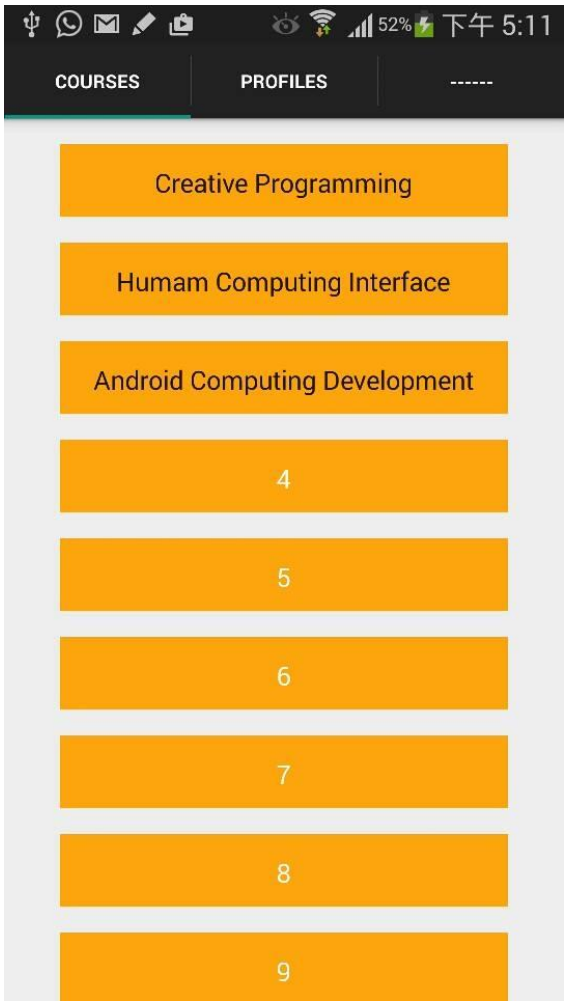
If the user input the incorrect id or password, the application will show the “Error Message” in the screen.

If the user types the correct information, the application will switch to the home page.And the user can open the side bar and pick the functions below.

ActionBar

Get Courses of the user

User can press the option “COURSES” in theactionbarAnd thenthe view will change to the course page. It show all the course that the students enrolled. User can click the courses and browses the inner element in the course.



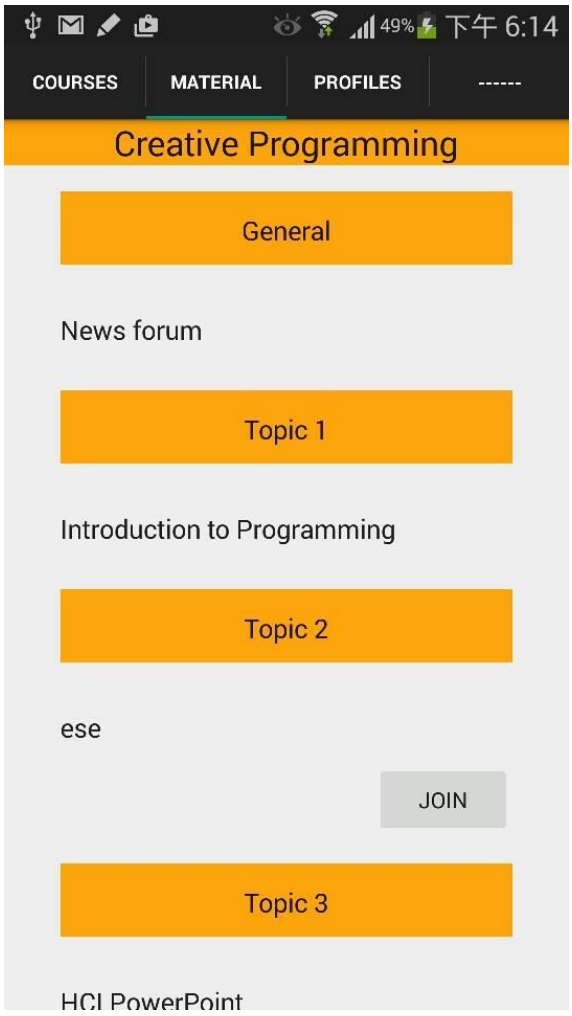
Get Course Content

Below shown the inner element of the course, and there are different material inside the course. Eg, eLessonmodule,PowerPointetc.

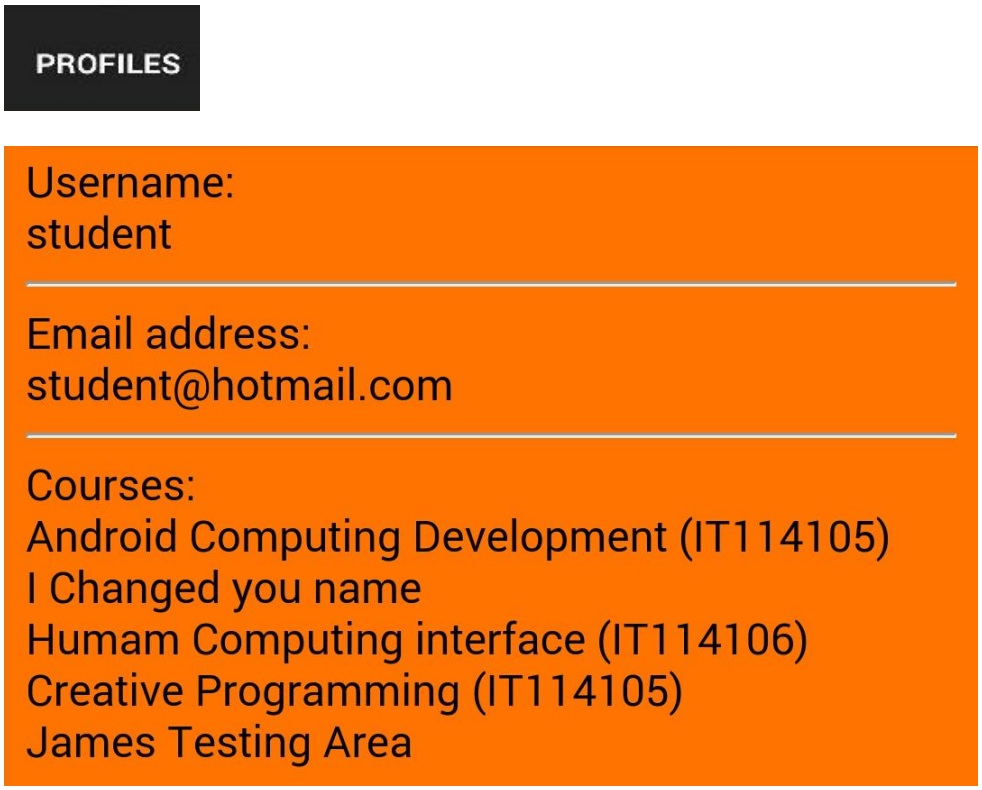
The user can click the module and browse the material inside the module.

The material inside the eLessonincludesthe question and PowerPoint.

However, we are developing the eLesson plugin .The function of browsingmaterial.



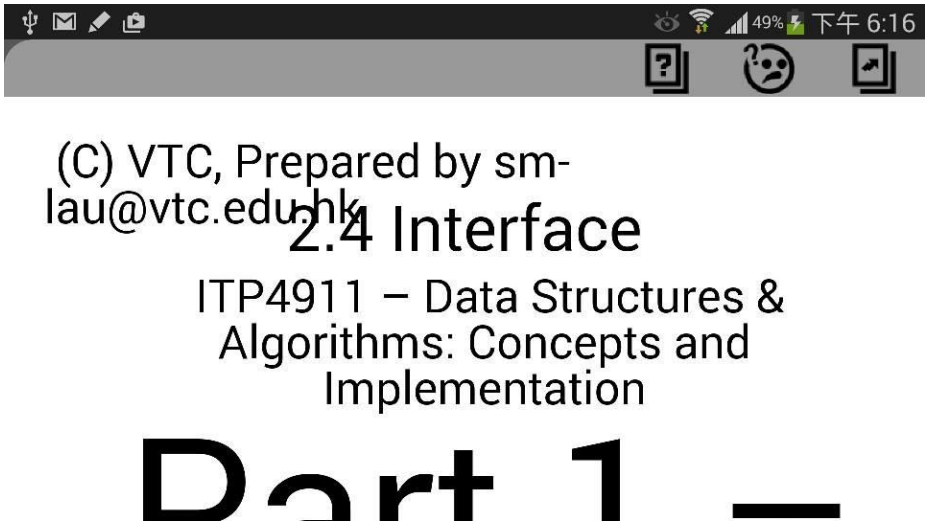
Get User information

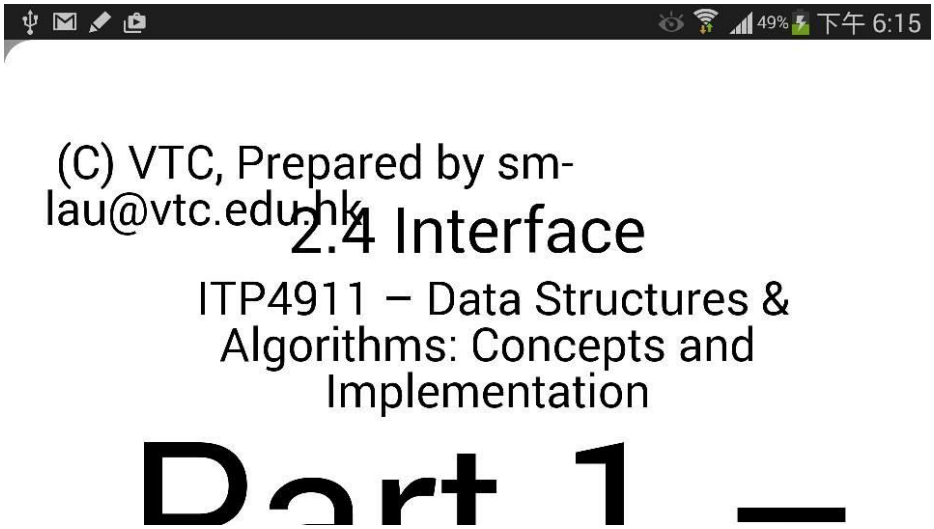


Elesson

Click the Join button to join Elesson created by teacher.

If the elesson class existed , the app will automatic show the current powerpoint page of the elesson.





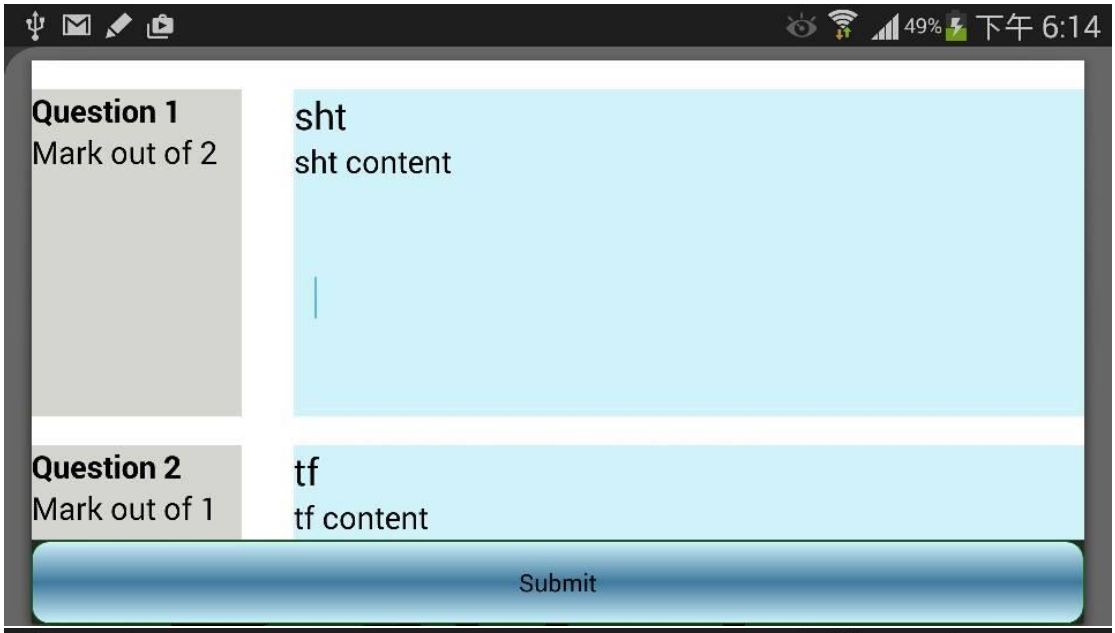
Notice:

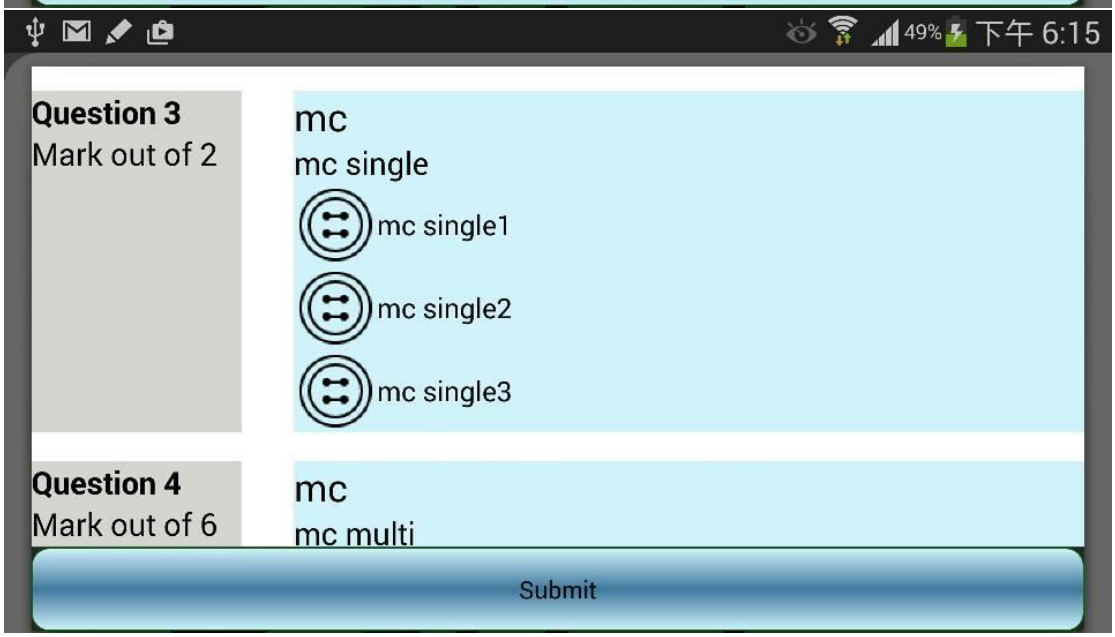
If the user stay on the page , the app will notice the teacher that the user joined the lesson.

Else if the user leave the page , the app will notice the teacher that the user left the lesson.

Elesson-Question

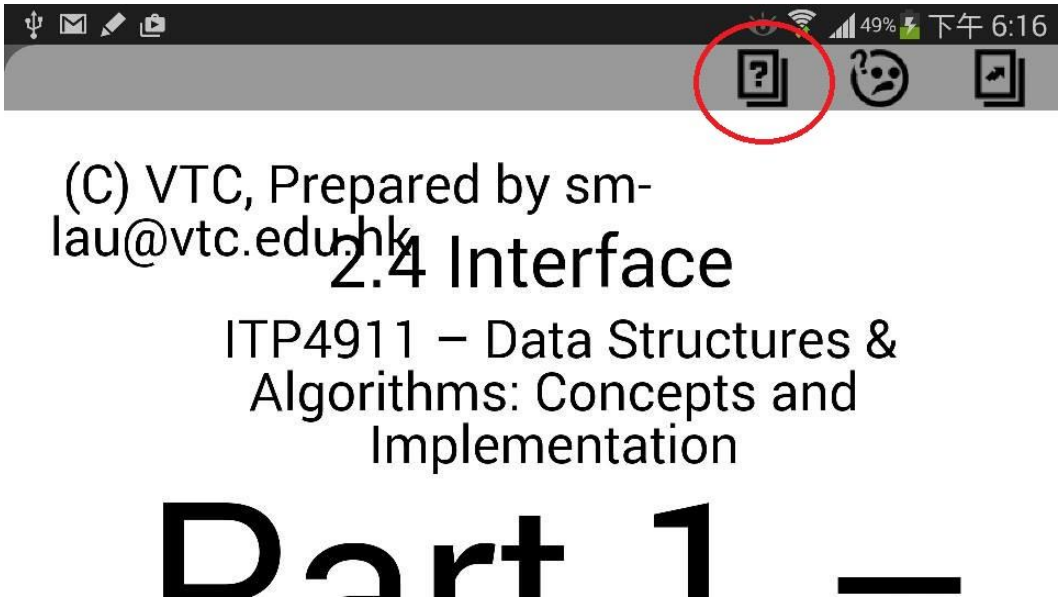
When the teacher broadcast the question, the question will cover the screen.

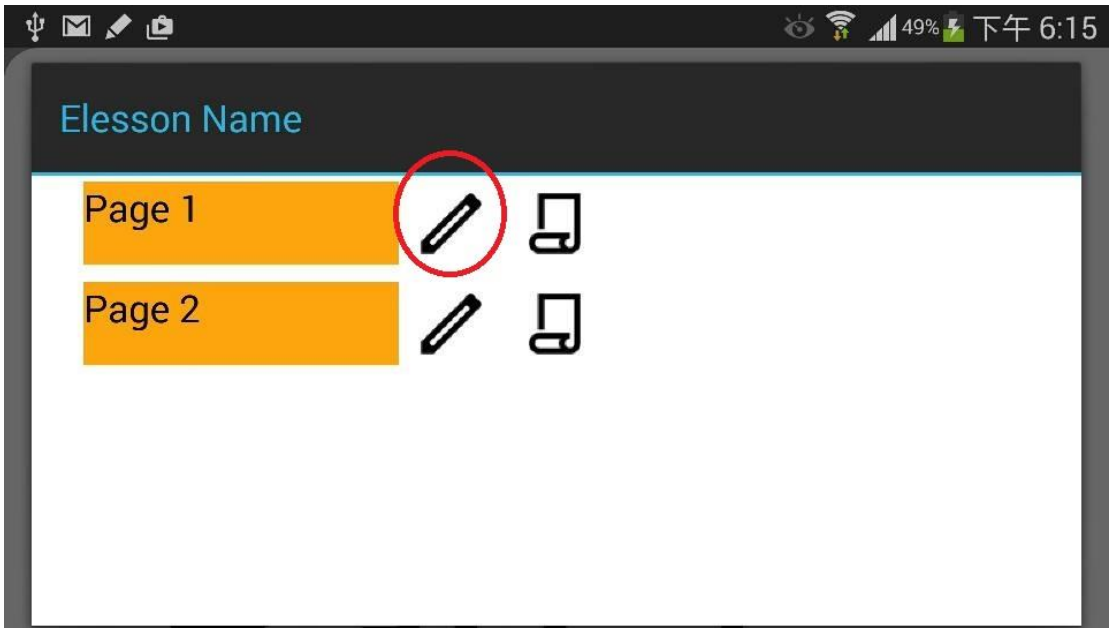


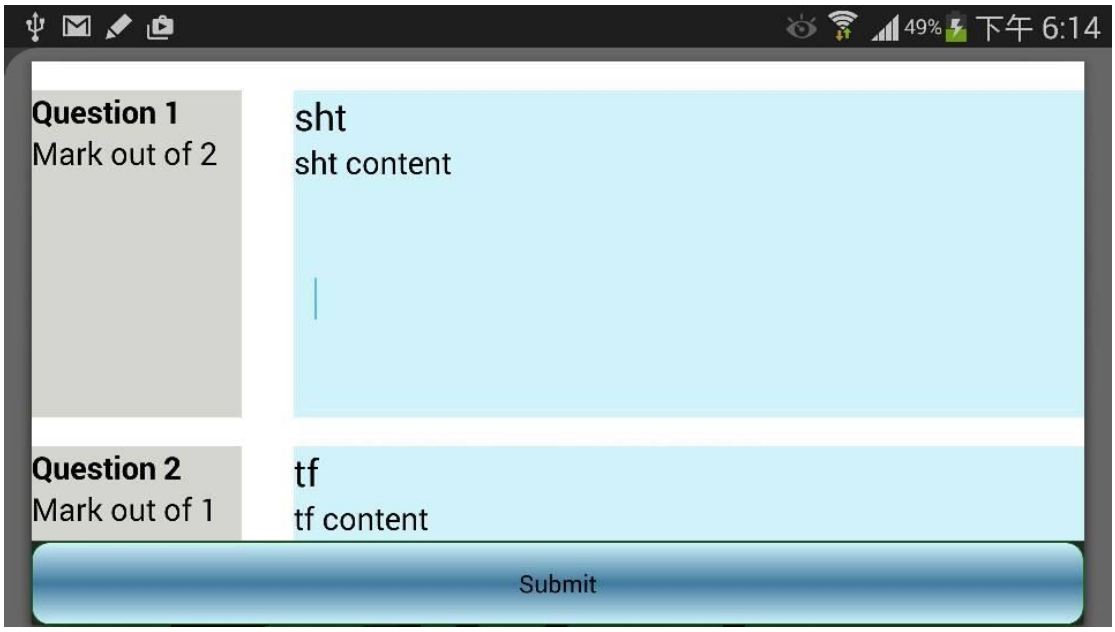


If student miss out the chance to answer question.

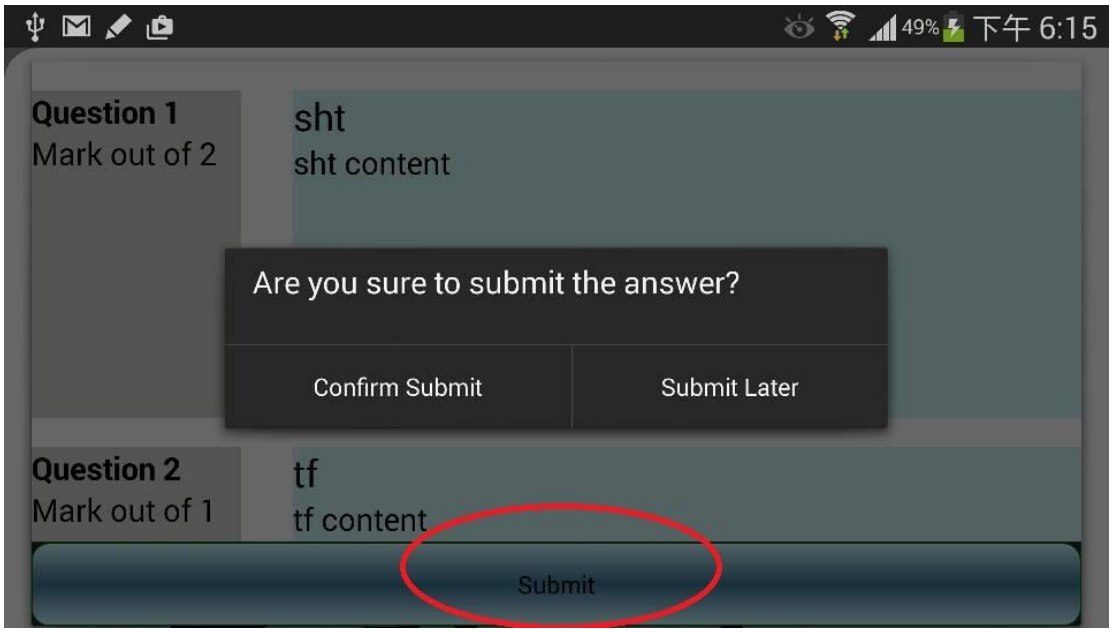
Option can be chose for reopen the question dialog.







Click the Submit button submit the answer



Notice:

If the student answered the question, no resubmit is available.

PowerPointHTML Parser Application

Tool to convert the pptx file to html file

PHP

PHP not only is a server-side scripting language designed for web but also used as a general-purpose programming language.

JSON

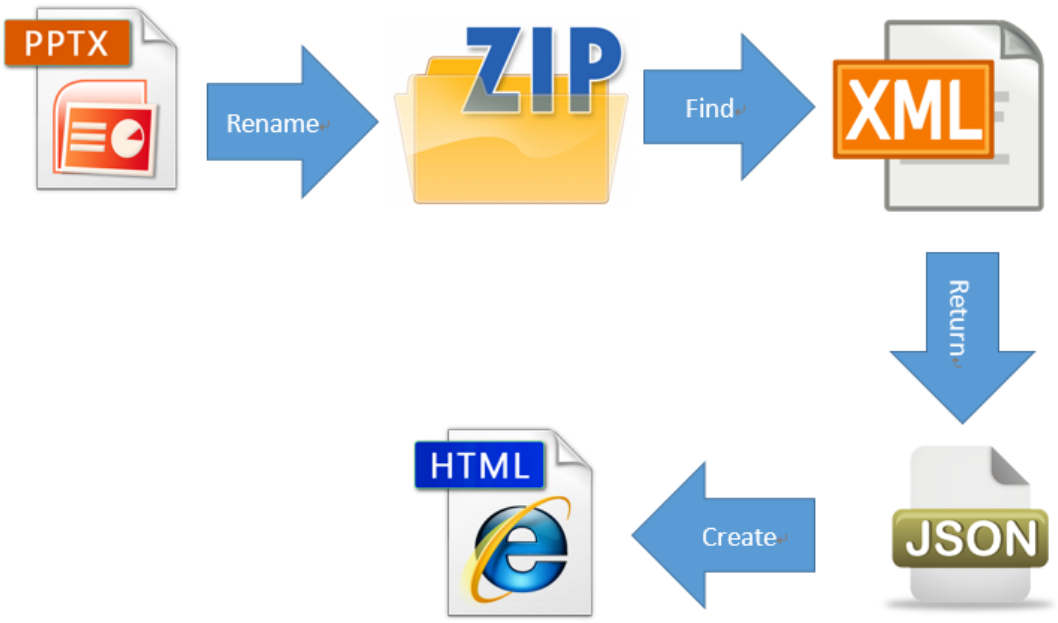
JSON is an open standard language which is derived from the JavaScript scripting language.

DOMDocument

DOMDocument is a php class which can read, write and execute the html file and xml file.

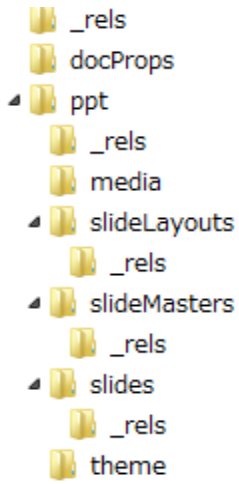
How the system convert the pptx file to html file?

Frist, the system will rename the pptx file to convert it to a zip file. Then unzip the file and find the xml file which save the detail of each slide. Next, the system will read these xml file and get the element of the each element in pptx file, for example, the word of text, the align of text and the color of text. After that, the system will encode these elements to a json file. Finally, the system will use the json file to create a html file like the pptx file.

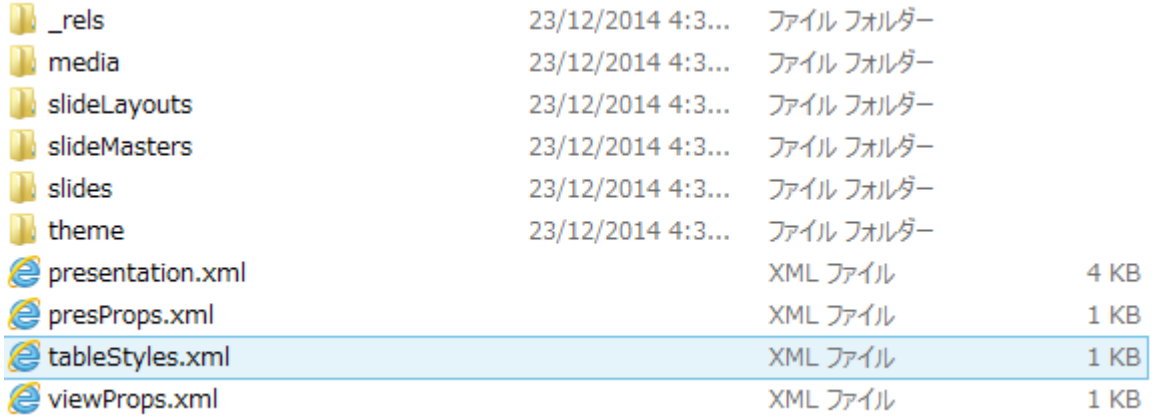


The detail of zip file, which is the pptx before the system convert it

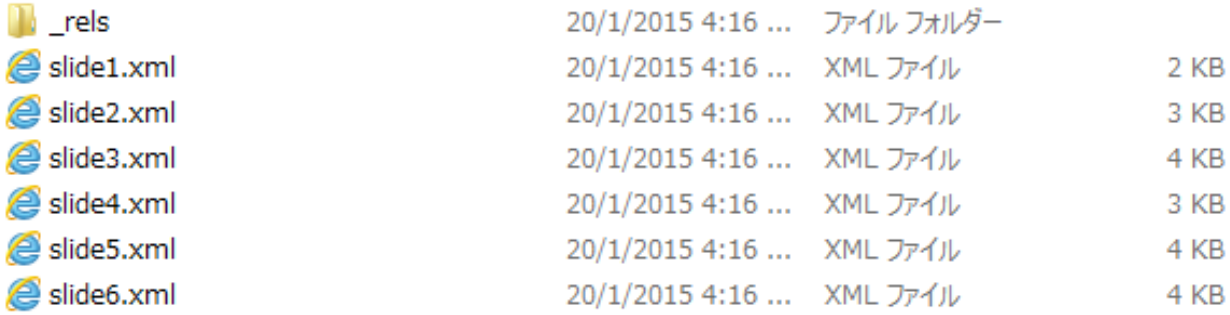
Unzipped file:



The items, like images, in Power Point is saved in /ppt/media

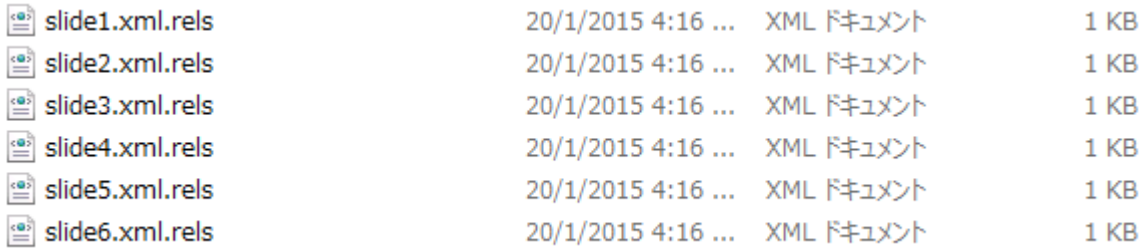


The xml of each slide we need is saved in "/ppt/slide"



The slide?.xml is saved the slide detail.

The path of item in Power Point is saved in “\ppt\slides_rels”



How to use the xml file to create a json file

Before this step, the system needs to rename the pptx file to a zip file and unzip it.

Then, the xml file is in “pptx\ppt\slides” directory.

```
- <p:sp>
  - <p:nvSpPr>
      <p:cNvPr name="副標題 2" id="3"/>
    - <p:cNvSpPr>
        <a:spLocks noGrp="1"/>
      </p:cNvSpPr>
    - <p:nvPr>
        <p:ph type="subTitle" idx="1"/>
      </p:nvPr>
    </p:nvSpPr>
    <p:spPr/>
  - <p:txBody>
      <a:bodyPr/>
      <a:lstStyle/>
    - <a:p>
      - <a:r>
          <a:rPr lang="en-US" smtClean="0" dirty="0" altLang="zh-HK" err="1"/>
          <a:t>subTitle</a:t>
        </a:r>
        <a:endParaRPr lang="zh-HK" dirty="0" altLang="en-US"/>
      </a:p>
    </p:txBody>
  </p:sp>
```

After that, we use the DOMDocument in php to get the element we need in the xml file. Next, we put these elements to an array and encode the array to a json file.

```
{
  "id": "3",
  "name": "\u526f\u6a19\u984c 2",
  "type": "subTitle",
  "array_p": [
    {
      "textContent": [
        [
          {
            "text": "subTitle",
            "size": "",
            "color": null,
            "bold": "",
            "italics": "",
            "underline": ""
          }
        ]
      ],
      "algn": null,
      "marL": null,
      "indent": null,
      "char": null,
      "type": null
    }
  ],
  "x": null,
  "y": null,
  "cx": null,
  "cy": null,
  "rot": null
}
```

How to create the html file by using json file

After the json file is created, the system will set the default element.

```
$head="<!doctype html><html><head><meta
name=\"viewport\" content=\"width=1200\" />";
$head.="<link href=\"./present/fathom.sample.css\"";
$head.="rel=\"stylesheet\" type=\"text/css\" /><link
href=\"./present/pptx.css\" rel=\"stylesheet\">";
$head.="<script
src=\"./present/jquery-1.11.2.min.js\"></script>";
```

Then the system will call each json by using while loop to get the element.

```
while($loop==true){
  $check=false;
  for($i=0;$i<count($files);$i++){
    if($files[$i]=="slide".$index.".xml"){
      $check=true;
    }
  }
  if($check!=true){
    $loop=false;
  }else{
    $html1= new slideToHtml;
    PPTXSlide::slide_api($path."/slide".$index.".xml"
    ,"slide".$index);
    $slide=$html1->toHtml("slide".$index, $index);
    $body.=$slide;
    $this->pptx[$index-1]=$slide;
    $index++;
  }
}
```

After that, the system will decode the json file and set these element, which gotten from the json file.

```
$data = json_decode($line_of_text, true);
```

```
$x;
if((!empty($div['x']))){
  $x=(int)((int)$div['x'])/12700;
}
$y;
if((!empty($div['y']))){
  $y=((int)((int)$div['y'])/12700);
}
$w;
if((!empty($div['x']))){
  $w=((int)((int)$div['cx'])/12700)-$x;
}
$h;
if((!empty($div['y']))){
  $h=((int)((int)$div['cy'])/12700)-$y;
}
$rot;
if((!empty($div['rot']))){
  $rot=((int)((int)$div['rot'])/60000);
}
```

```
$area=new setArea($textTetail[$i],$tType);
$this->in.="<pre style=\" text-align: ".$area->
getAlgn()."; \">".$area->getArea()." </pre>";
```

Finally, the system will provide a function to return the array which saved the detail on each slides.

```
function getHtml(){
  return $this->pptx;
}
```

Problem Encountered & Solutions

Problems in traditional classes

In traditional classes, the teaching process is in one-dimensional. The teacher teaching based on the textbook, PowerPoint or prepared paper materials. The student is just listening to the teacher and interacts with the papers. That would last for 40 minutes or even more. Sometimes, students would feel that the lesson is quite boring and could not last for a long time focusing on the lesson and teachers' talk.

That brings to multiple problems. First, the lesson is boring. There is less chance to let teacher and student having more interactive teaching in the lesson. Second, the teacher teaching the knowledge, but they cannot know whether the student is understood to the things just teach or not. There is no immediate result. Third, teacher would not know a student is really participating in the lesson or not.

Therefore, we start focusing on this project. That use IT technology and device interacting with the lesson and make the lesson more fun. Teachers and student can both join to the lesson by their device. Teacher can get immediate result and feedback from the student, so that teacher can make improvement. Student can get the lesson resources by the internet instead of distributing the papers in 3-5 minutes. Students can also answering the questions and getting the grading to keep tracking with the learning process. They can also communicate through the platform. So that, shy student can also express their opinions through the platform.

Problem Encountered of iOS Application

Learning Swift

Time consumingon learning Swift. In swift, there having many new concepts. Tuple, Closure,a range data type, optional bounding. It gets times to learning this language.

Many Bugs when programming with Swift

When developing with Swift, there having many confusing bugs. As it is a new programming language, there have some bugs and some are even not easy to fix as there are not many answers on the internet. Also, the error prompt from XCode is not clear enough, it may also leading us to debug the program in the wrong way. Figure 28 is the error that XCode always prompt to me and that is a bug in XCode.

Figure 68xcode bug

Consuming times on investigationof Web app

We have been considering developing the application in a Web app way or as a Hybrid app to developing a better UI. So, we have searching and learning some of the

technologies of it.We investigate for some of the Hybrid App framework like Ionic, Famo.us, F7, augularJS and Phonegap.

Figure 71 ionic framework

Figure 70 famo.us framework

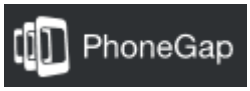


Figure 69phonegap framework

Lastly, we are not developing the app in this manner becausethere have too much unpreditible situations development and AngularJS requires more experience on programming with javascript. Also, a web app will be store the html, js in local, that meaning the source is visible for other people. However, we use quite a lot time on the way, this is also the reason that the native application prototype is presenting less things.

Difficulty in displaying PowerPoint in iOS Application

The PowerPoint in requesting to the server and the PowerPoint will be send by JSON format. ThePowerPoint retrieved from JSON in a String of div HTML tag element. It will be displaying in the device in a web view format. The html string putting into the UIWebView element.

The problem is that the server need to send the stylesheet(.css) and javascript(.js) to impact to the PowerPoint page in the JSON web service request also. After that, the iOS application should be mainipulating the returned String and append the String as the original PowerPoint. That may involves multiple work of String mainipulation in both client and server slide. Beside this, The UIWebView should also displaying the PowerPoint in a correct scale. As the device is in both poritait and landscape mode, It is requires some calculation to let the PowerPoint fit into the UIWebView and displaying in the right scale. Finally, the UIWebView should disable scrolling and enable zoom to full screen size to display. This process involves much work.

Reason for using Swift for Development of iOS Application

The iOS application is developing by Apple new programming language Swift. Because swift can compatible with Objective-C code in project development. Objective-C developing for a long time, there are many great sources to focus on like AFNetworking. With those source, would make the application development process more easier.

Figure 72 Swift Icon

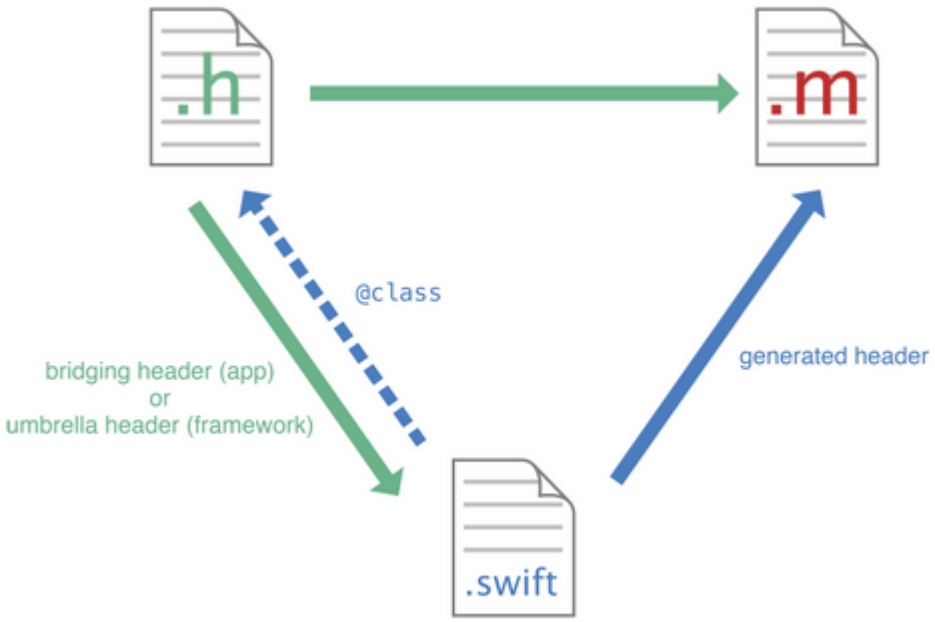


Figure 73 compile process with .h .m

Easy Compatible with Objective-C

In compatible with Objective-C in Xcode6, we just need to come with the bridging header name as [Project-name]-Bridging-Header.h and import the Objective-C libraries in this file. XCode would then automatically read the bridging-header and

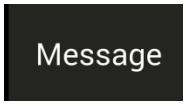
import the library for swift.

Cleaner Code & future extensible

The other reason is Swift making iOSdevelopmentmore easier in coding perspective. Some function implemented with Swift would make more clear and less code. Also, swift may be the programming language of iOS future version development. For further upgrade and implementing more functions in the application. Therefore, we use the Swift in this project.

Problem Encountered of Android Application

Message



The message function is not yet implemented.Weare studying push notification technology and there are some data collisions when the program call the web service.

Hard code

Some function have to hard code in this moment .We have no yet decide what should we do in the function.For example, the Home page is a static web page in this moment. We are going to implement some new function on the home page depend on the non-functional requirement. However, first of all we have to finish the core function of our application first rather non-functional requirement.

Repeating Code

Some coding is repeated in the program. For example, the AsyncTask in Android application. There are too many repeating code on each time calling the web service. From the increasing of implement functions, we have to redesign the data structure and program structure. For example, it is suitable to Implementthe Object-oriented concept into our program. Open for extension and closed for modification can help us to avoid the error in the future development.

```
if (datatask == null || datatask.getStatus().equals(AsyncTask.Status.FINISHED))
{


   datatask = new DataAsync();
   datatask.execute();
   Toast.makeText(mContext," Welcome to moodle ", Toast.LENGTH_LONG).show();
}
```

```
if (msgtask == null || msgtask.getStatus().equals(AsyncTask.Status.FINISHED))
  {


    msgtask = new MessageAsync();
    msgtask.execute();
  }
```

Unformatted Main Activity

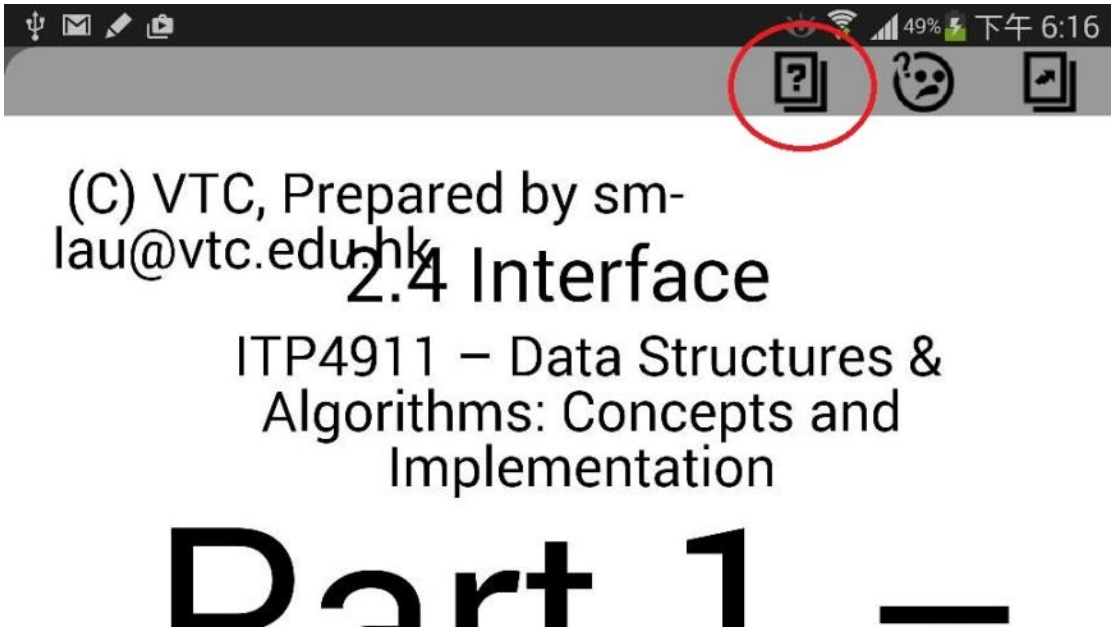
In this moment, the coding of the Application is quite messy, it is because there are too many attribute created in the main activity. And the program has implements a lot of inner class like AsyncTask and Thread.We will redesign the program structure like using Command pattern.

```
public class timerTask extends TimerTask
{
  public void run(){
      Log.d("result", "hi");
  }

};

//Javascript
public class WebAppInterface {
//Javascript
```

PowerPoint implementation

We have to Implement the function that allows teacher upload the PowerPoint from computer and show it in student mobile application. In this moment we are still work at the resize the PowerPoint into html. We have done something but the program code of conversion is too unreliable currently. We have to solve the problem that the hard code pixels in html and the fixed size of font and element in the html. In the mobile, we have to convert the fixed pixels again and resize the PowerPoint one more time.

However, it is not a suitable and reliable method. We have to design another way like convert it is to percentage format or transform the scale of different height or width.



Problem Encountered of PowerPoint HTML Parser

The system cannot set all the shape into the html file because there are too many type of shape. Some of the shape can not only use the html, CSS or JavaScript the create.

Moreover, there are many default elements in Power Point but the system cannot find the default element because they do not save in the xml file, which save the detail of the slide, of the Power Point file.

Also, there are some different between the Power Point file and the html file,forexample,the position of the items.

Furthermore, if the pptx file is created form the Power Point which language is Chinese, the system will not detect some of the element, for example, unordered list, in that pptx file.

Finally, the html file cannot use the “←”key and ”→”key in the keyboard to change the slide, because we cannot find the code to let the computer know the user enter these key.

Solution

By solve the problem of images, we cannot find the solution new, but we try to use another way to get the path of image, for example, rename the file which save the path of images to a html file and try to get the path.

By solve the problem of the default element, we need to find thedefault element which user always use in Power Point and set them to a default element in the system.

By solve the problem of slide, we use the "," key and "."key to change the slide because we find some code can let the computer know that user enter these two key.

Problem Encountered of Web Moodle

Moodle platform compatibility

Moodle package is an on-going application, Moodle developers keep on updating a new version of moodle. For example, when the project is in the beginning stage, the latest moodle package version is up to 2.7+ but now the latest moodle package version is up to 2.9beta+. This will possibly cause some technical problems as the new version of Moodle may support old method or they modified the method for some purpose, this will make the eLesson crashed.

OOPHP

Moodle uses OOPHP to construct their system, which OOPHP is new for us. OOPHP is used for higher flexibilities and future extensibilities but it will cause the syntax of statements become much more complicated. It is hard to understand all the codes as one php file is connected with other php files and those php files will connect with other php files and so on, which means we may even need to watch above ten php files in order to know what is going on in one php file.

Existing architecture

Since our eLesson module is a plugin that is embedded into Moodle system, there are lots of rules and standard syntax we have to follow. For example, there are at least twenty files that must exist in order to let moodle system to read the plugin, and those files have a special naming system, if the file name contains any uppercase letter, this will cause the whole plugin cannot be installed. It is hard to find out what is wrong among these files as there are so many files that are not developed by me

and they usually have thousand lines of code, this causes me the same difficulties in combining moodle standard code and eLesson code.

Web Moodle Implementation

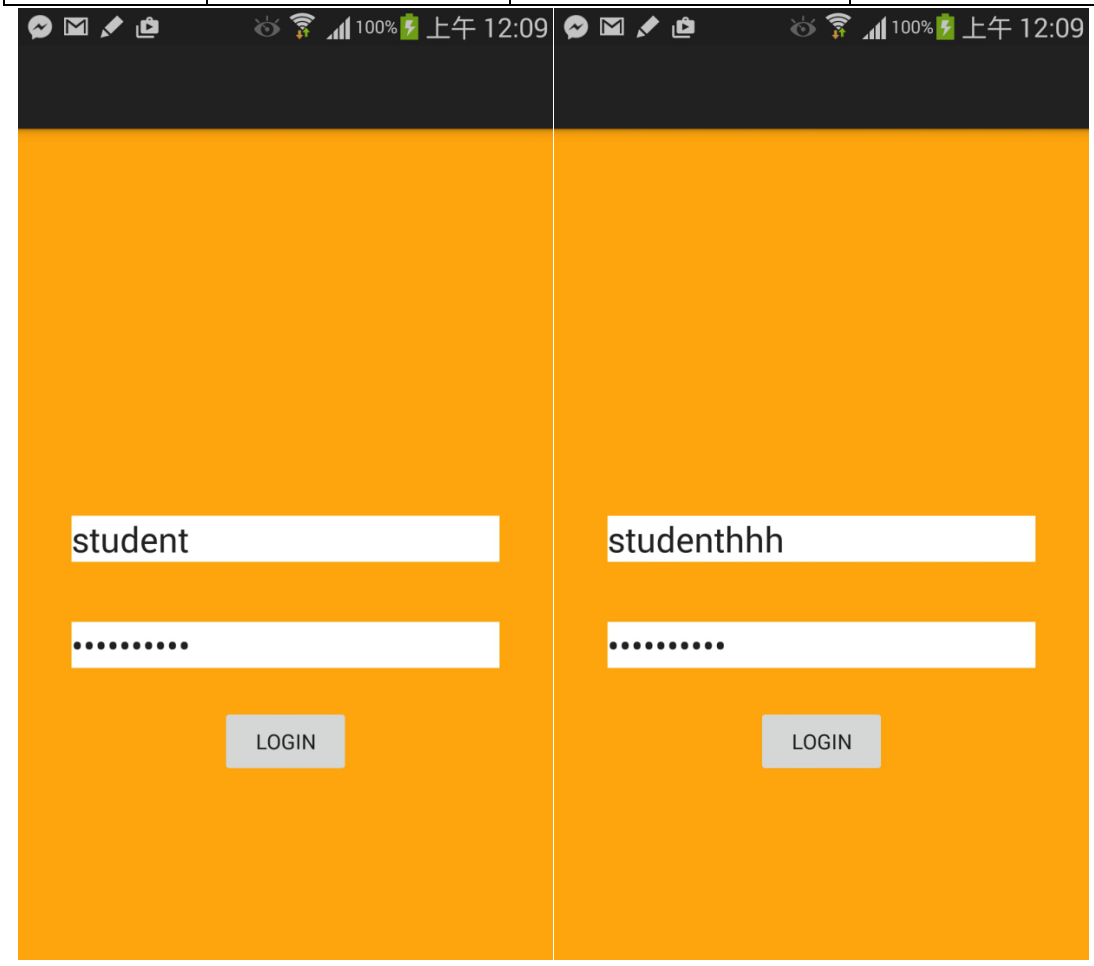
In order to solve the compatibility the different version of moodle, we could only test the functionalities of the moodle frequently to see if it goes wrong or not. Once there are some errors happen, we need to find out why and fix it quickly.

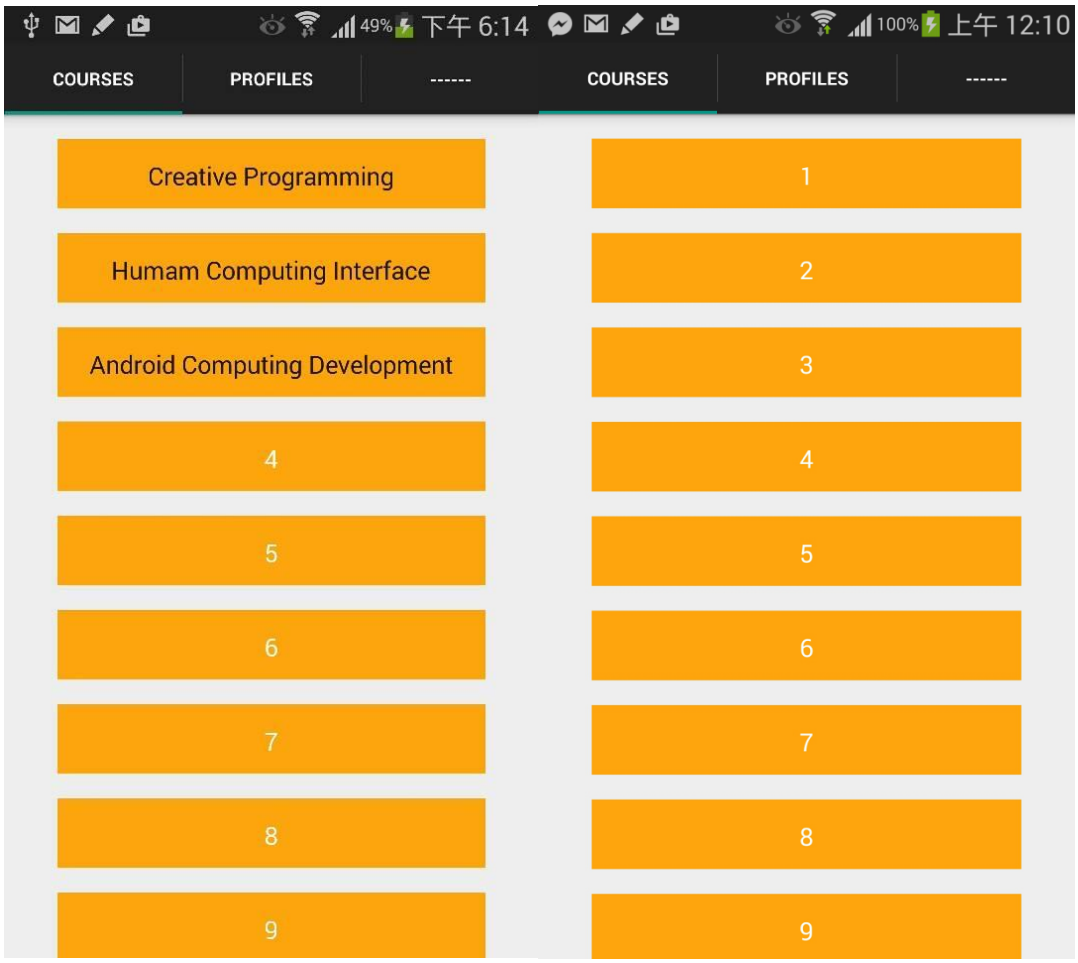
Dealing with the issue of OOPHP, we decided to learn to OOPHP, we went lots of website such as the official site of php, stackoverflow to understand the logic of moodle files. Also, we installed and enabled xdebug in moodle server so that we could investigate the program flow when the moodle is launching.

Besides, we visited official moodle forum and documentation site for moodle developers, to know more about what could be modified, what must be remained and not allow to be changed.

Test Case for Android app

Test Case	Description	Expected Result	Result
1	Login with correct username and paswword	Login success	Login success
2	Login with incorrect username and paswword	Login Failed	Login Failed
3	Login with incorrect username and password in chinese word	Login Failed	Login success without courses

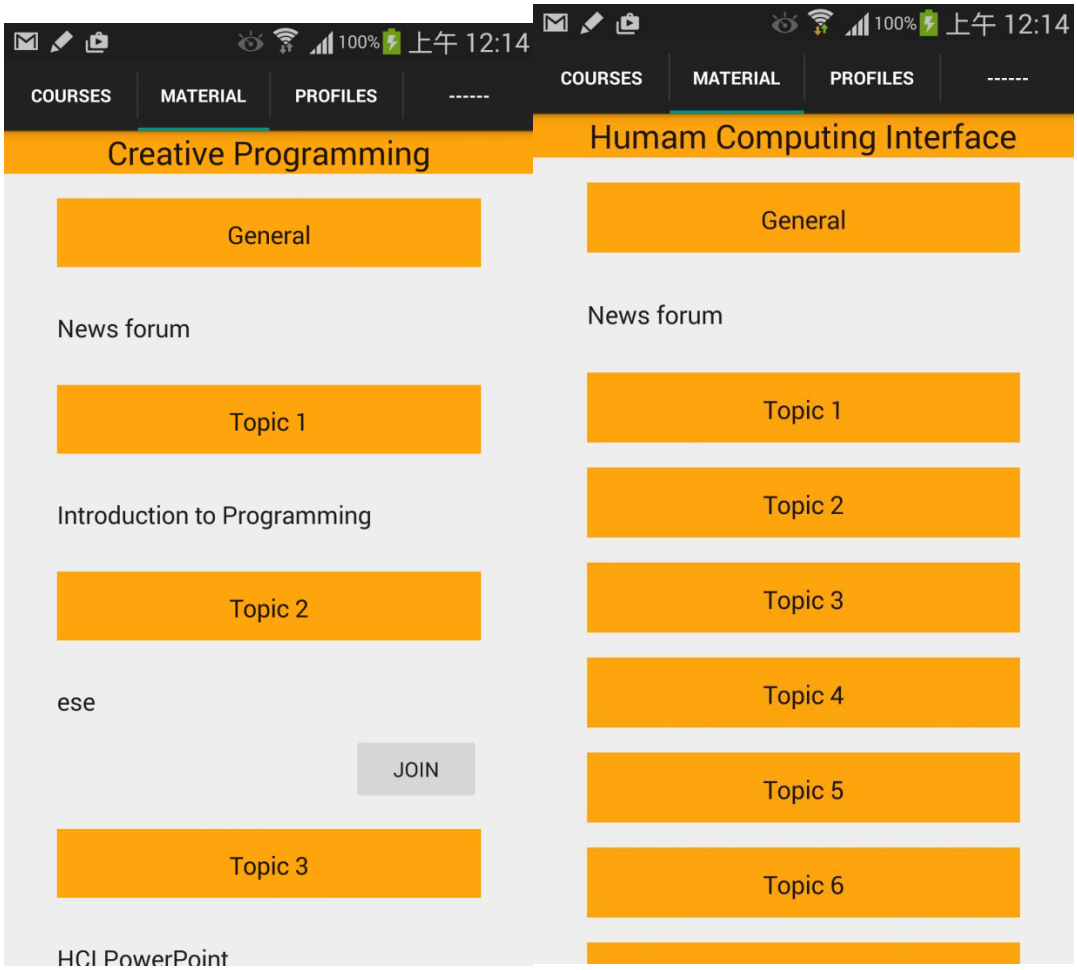



49% 下午 6:14
COURSES
PROFILES

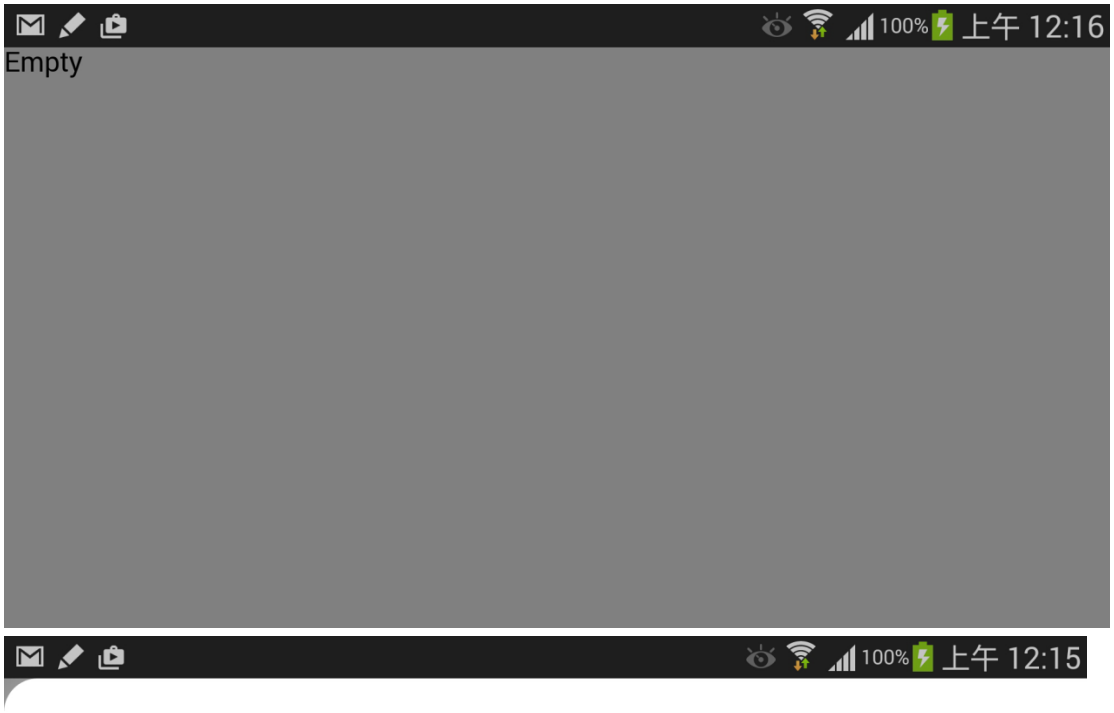
Creative Programming
Humam Computing Interface
Android Computing Development
4
5
6
7
8
9
100% 上午 12:10
COURSES
PROFILES

1
2
3
4
5
6
7
8
9

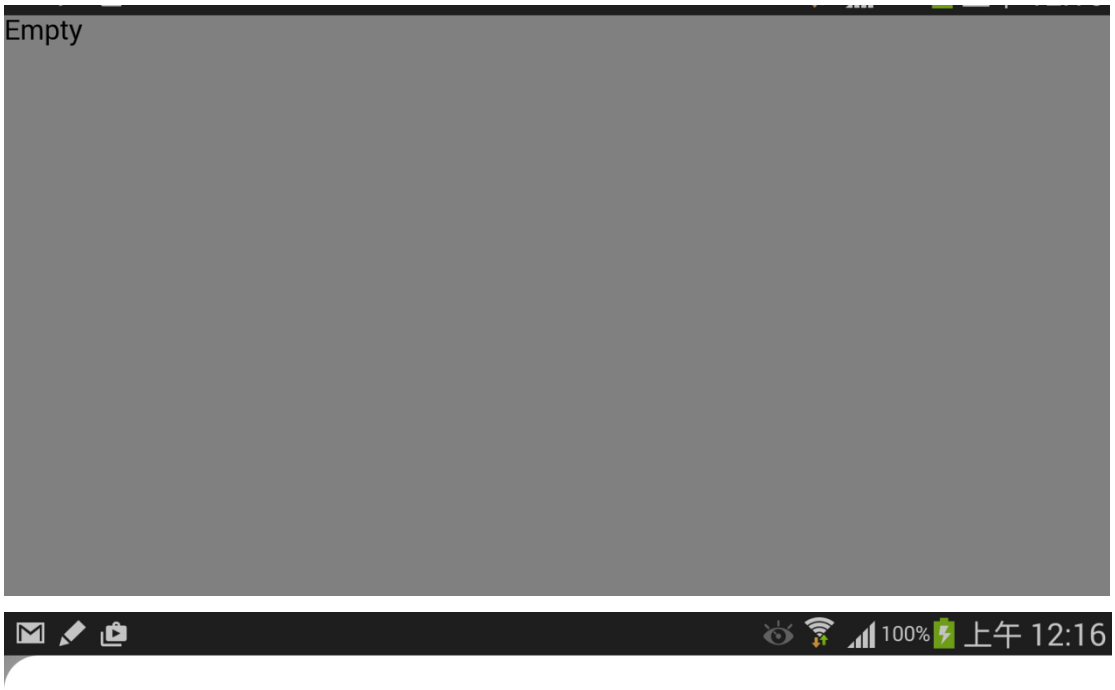
Test Case	Description	Expected Result	Result
4	Select First Enrolled courses to student	Show enrolled courses content	Shown success
5	Select Second Enrolled courses to student	Show enrolled courses content	Shown success

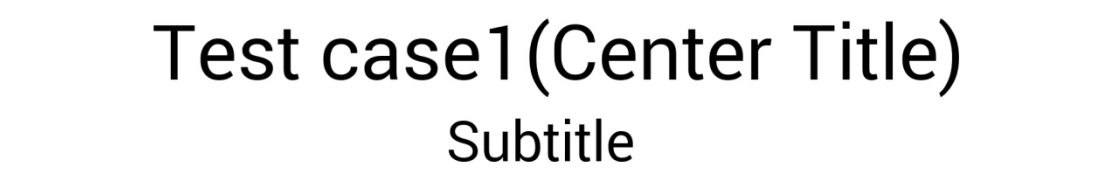


Test Case	Description	Expected Result	Result
6	Select one of the elesson class (Class Started)	Join success with powerpoint shown	Join success with powerpoint shown
7	Select one of the elesson class (Class not-Started)	Join not success without powerpoint shown	Join success without powerpoint shown

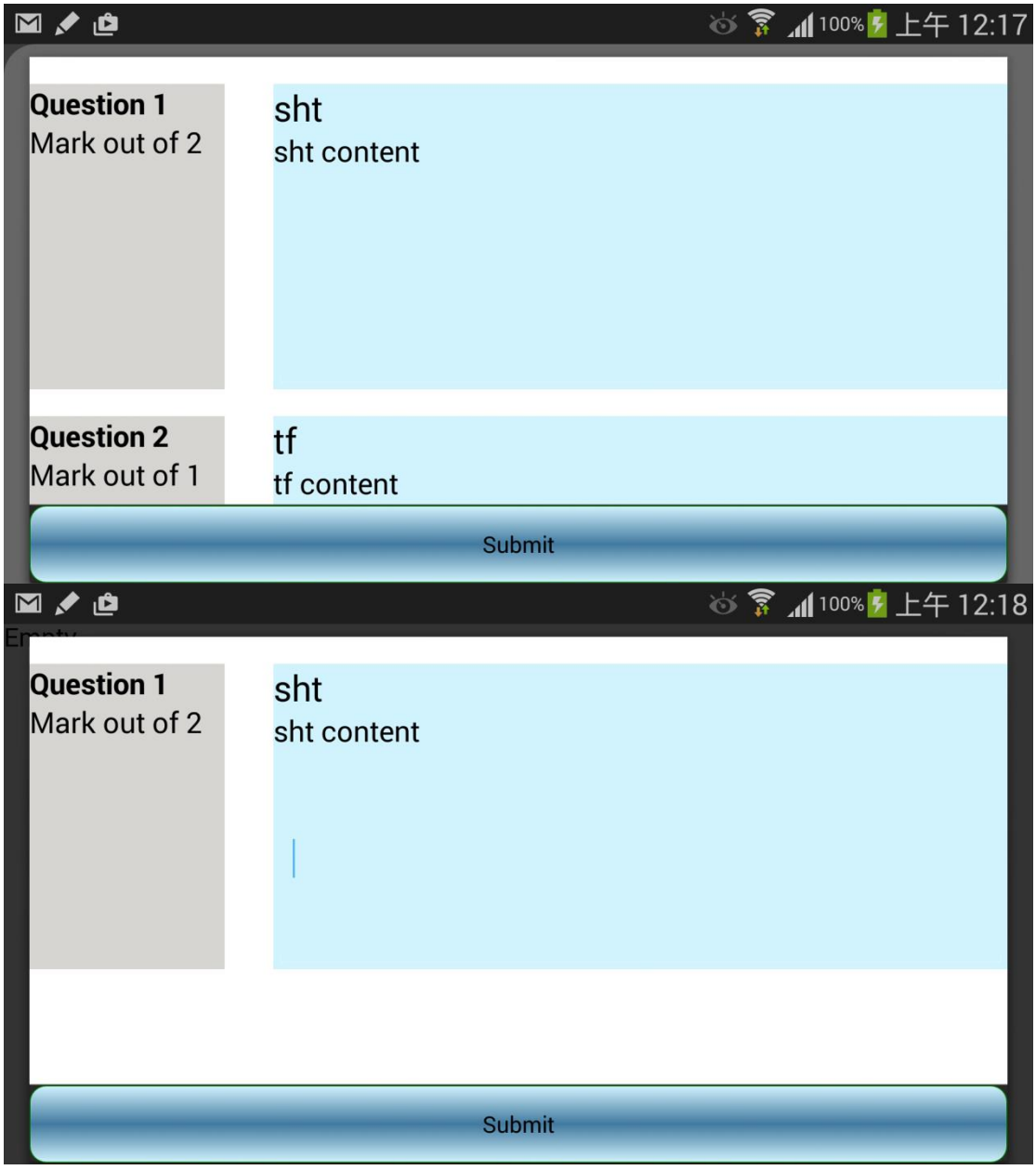


Test Case	Description	Expected Result	Result
8	Teacher control change page	change success	change success
9	Teacher stop elesson	Automatic to empty powerpoint page	Change sucess

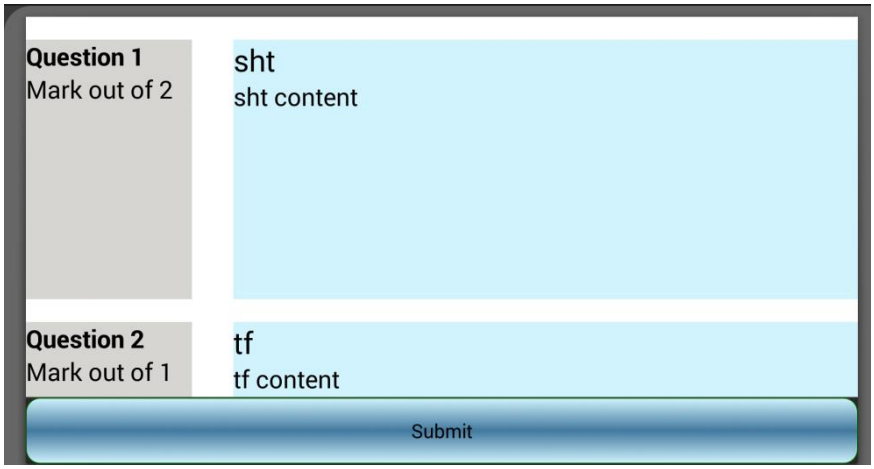


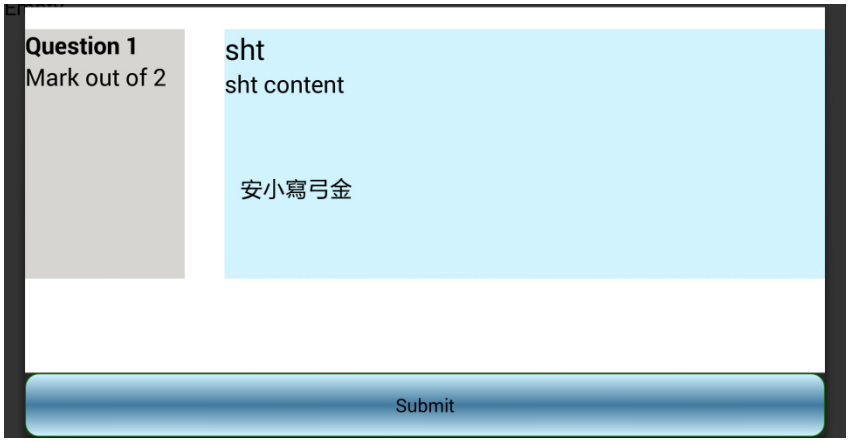


Test Case	Description	Expected Result	Result
10	Teacher broadcast the question	Instant shown in the app	Success
11	Student answering question, teacher broadcast another question	Replace the screen by the new dialog	Replaced screen by the new question

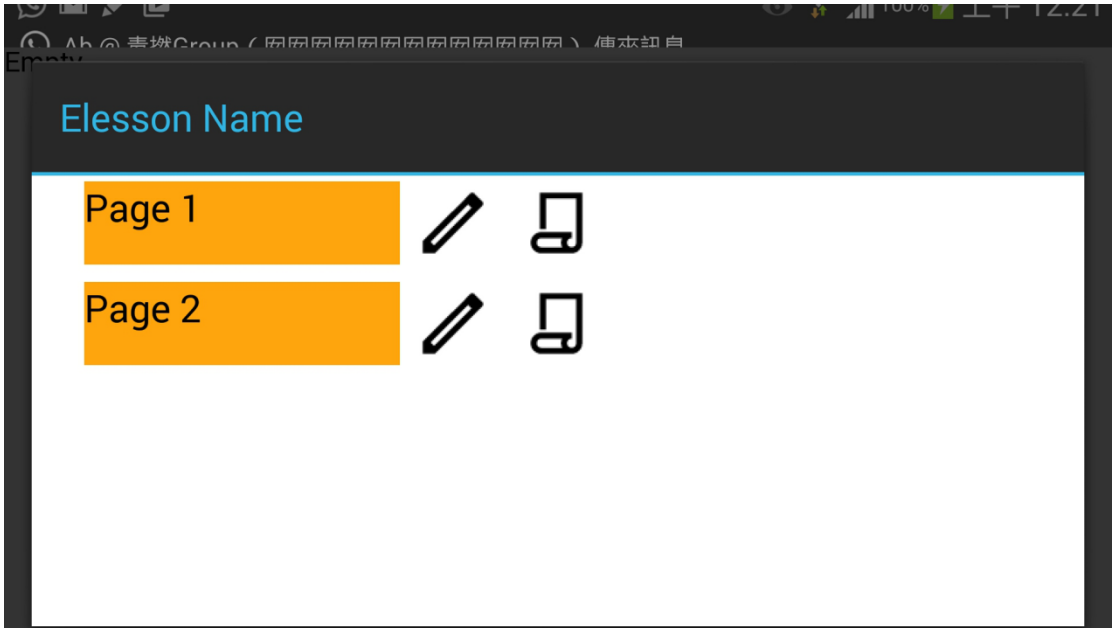


Test Case	Description	Expected Result	Result
12	Non-submitted question backup	Have backup	Have backup
13	Non-submitted question backup after rejoined the elesson	Have backup	Haven't backup

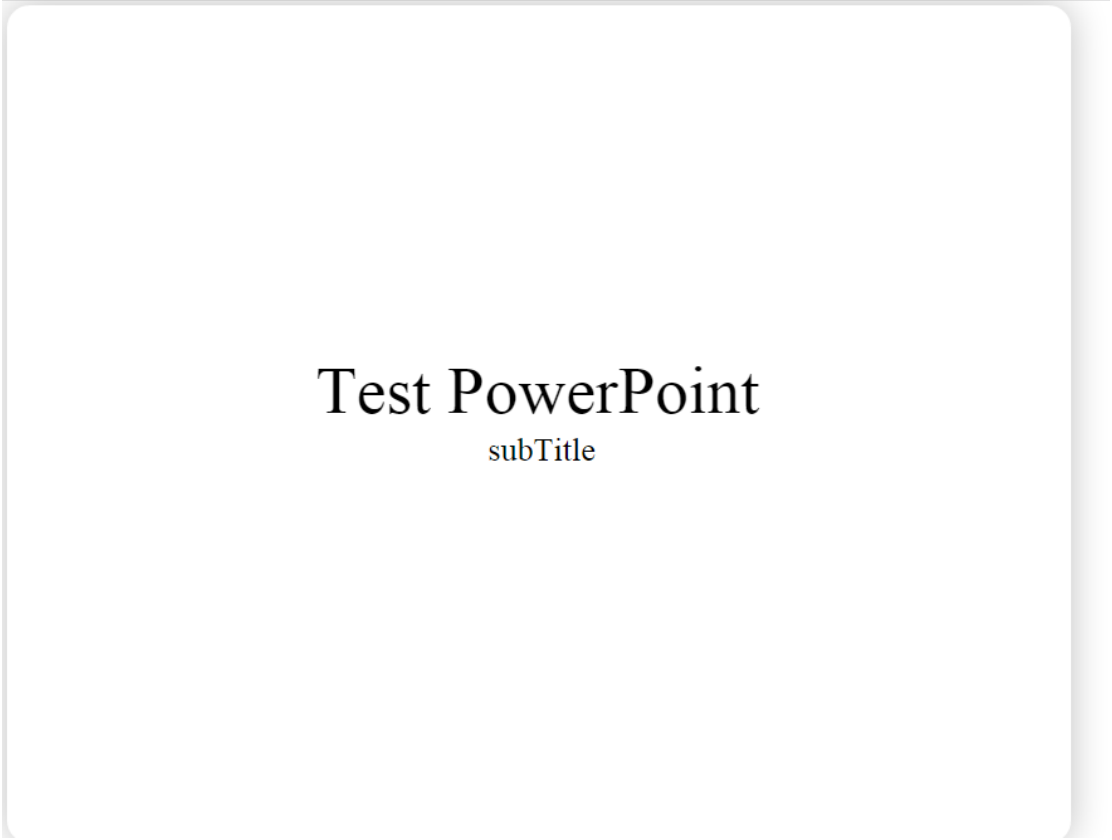




Test Case	Description	Expected Result	Result
14	Avoid resubmit	Disable the submit	success

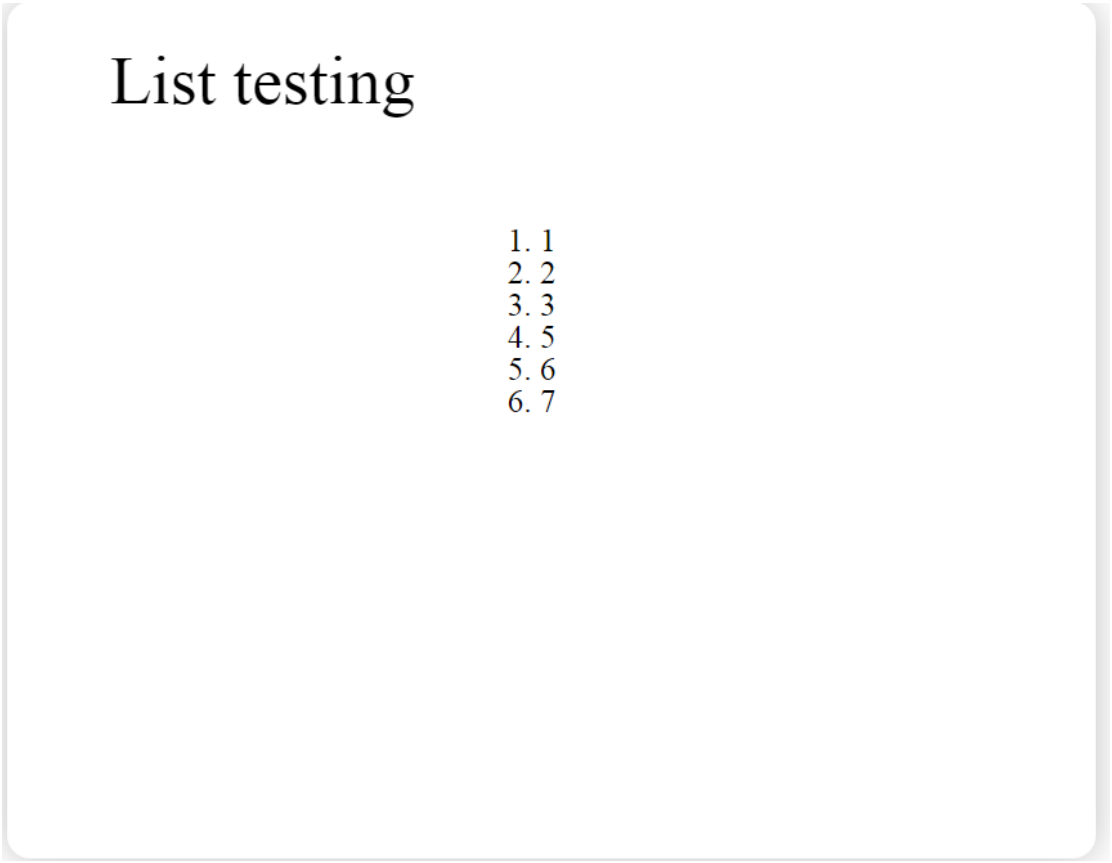


Test Case for PowerPoint HTML Parser



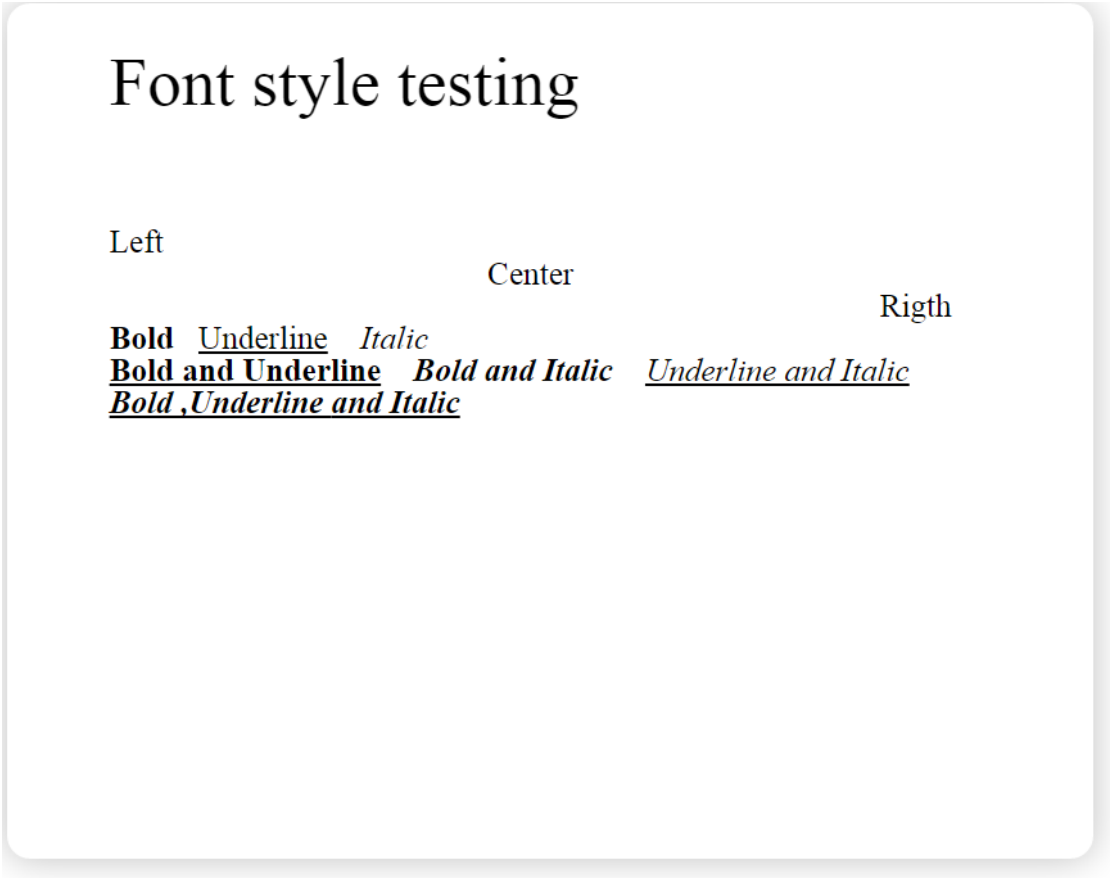
The system can detect the title and subtitle in Power Point. Then set align the title and subtitle to be center and let them can be look like the title and subtitle in Power Point.

Test the order list



The system can detect the order list in PowerPoint. However, the system cannot detect the unordered list if the Power Point file is created from the language of Power Point is Chinese.

Test the different font style

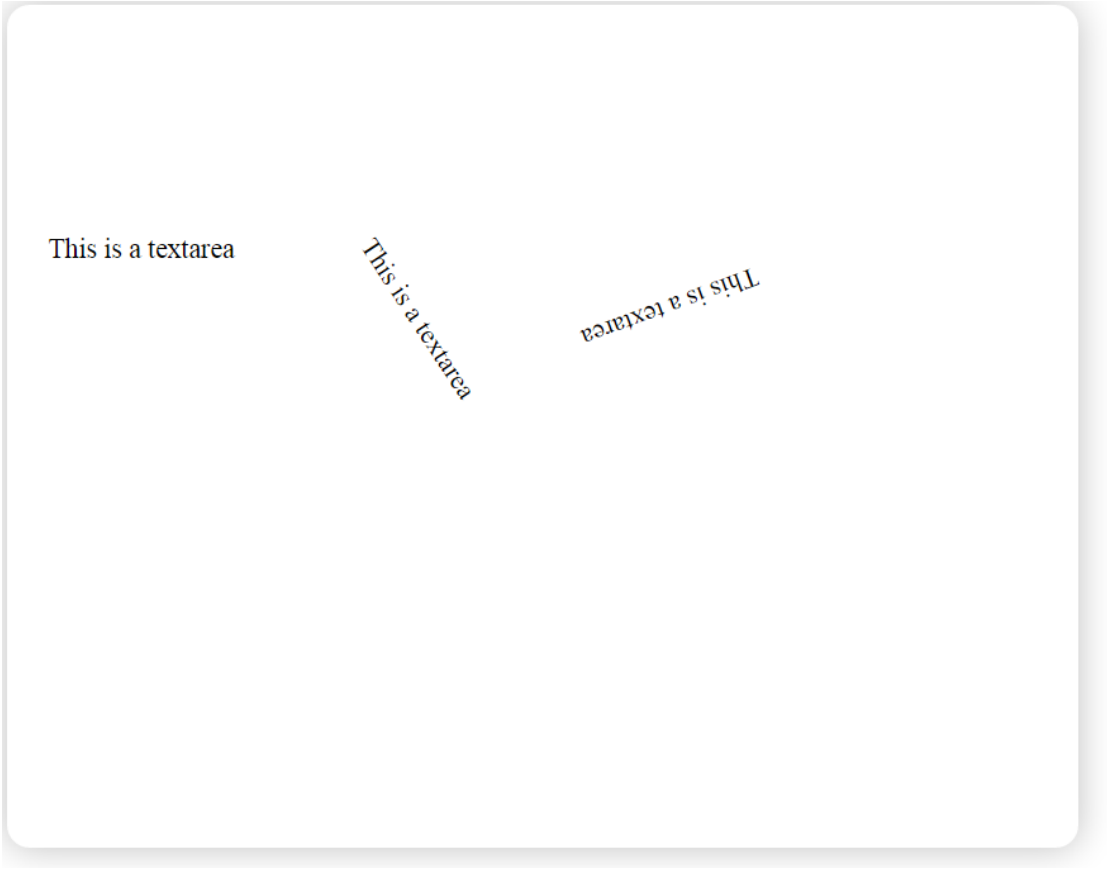


The system can detect the alignment of the text. Moreover, it also can detect the style of the text, like bold, underline and italic. Then three system will set the alignment in the html file.

Test the text with color

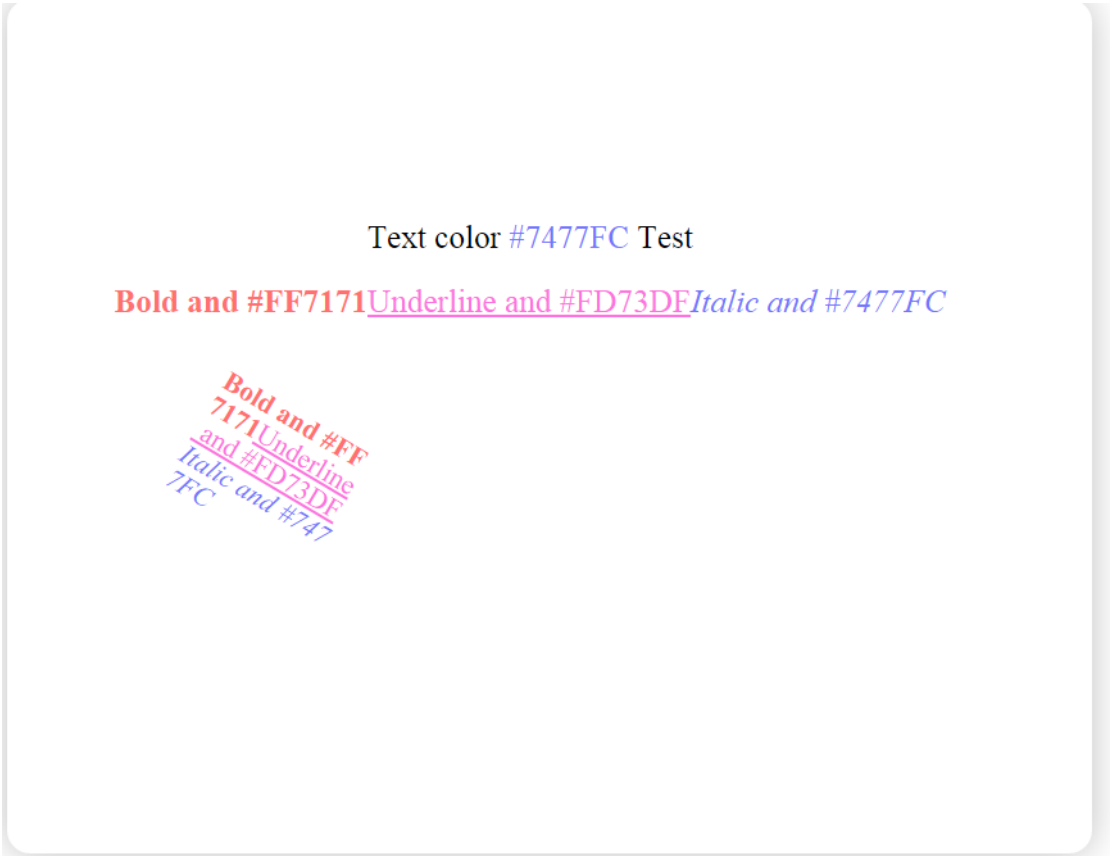
The system can find the color of the text and set the text color in the html file.

Test the rotation text with different angle of rotation



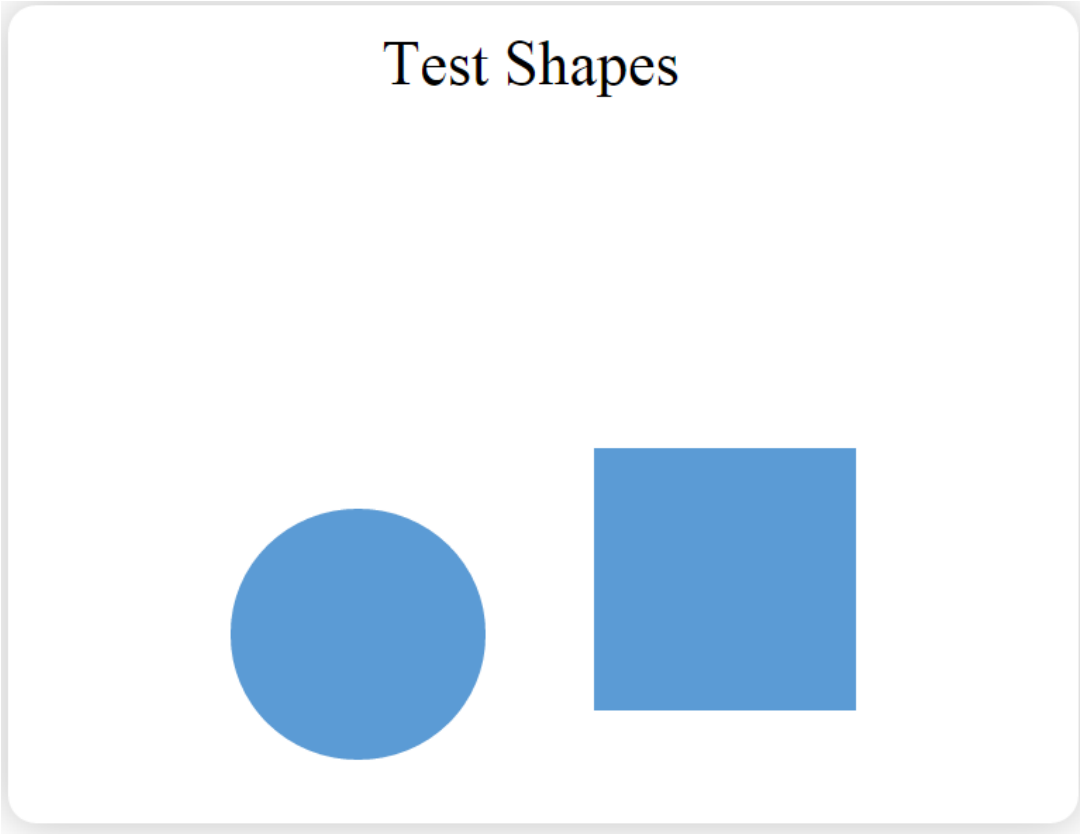
The system can detect the rotation text. The system will find the angle of the rotation text, then set it to the html file.

Test the text with different style and color



The system can detect the different text color with different the text style in the pptx file. Also, the system can detect the rotation text with other text style. Moreover, the system can wrap the text.

Test the Shapes



The system can detect the different the color of shapeand the category of shape in the pptx file.

Future Development

Future development for Mobile Application

As this is a midterm prototype, many functions that we have not developed. In the future development in iOS. This is the list that we are going to take consideration on development.

- To develop the main interactive function of lesson module in iOS.
- To develop the download function of module resources
- Download function also compatible with lesson resources

- Allow users to browse the resource store in local
- Allow users open the resources with other application
- Enabling the caching for some JSON data which make the application run more faster
- Enabling offline browsing details of Moodle cause some data are stored in /Document folder in the app bundle
- To develop another version of iOS Moodle application for teacher for the functions that limited to teacher

Future development For PowerPoint HTML Parser Program

This system is imperfect, because there are some function cannot provide in the html.

First, the html file which created by this system do not have the animation of each item, because there are many default element. These element cannot get in the xml. Therefore, we cannot set the animation before we find enough default element. We will try to find them in the future.

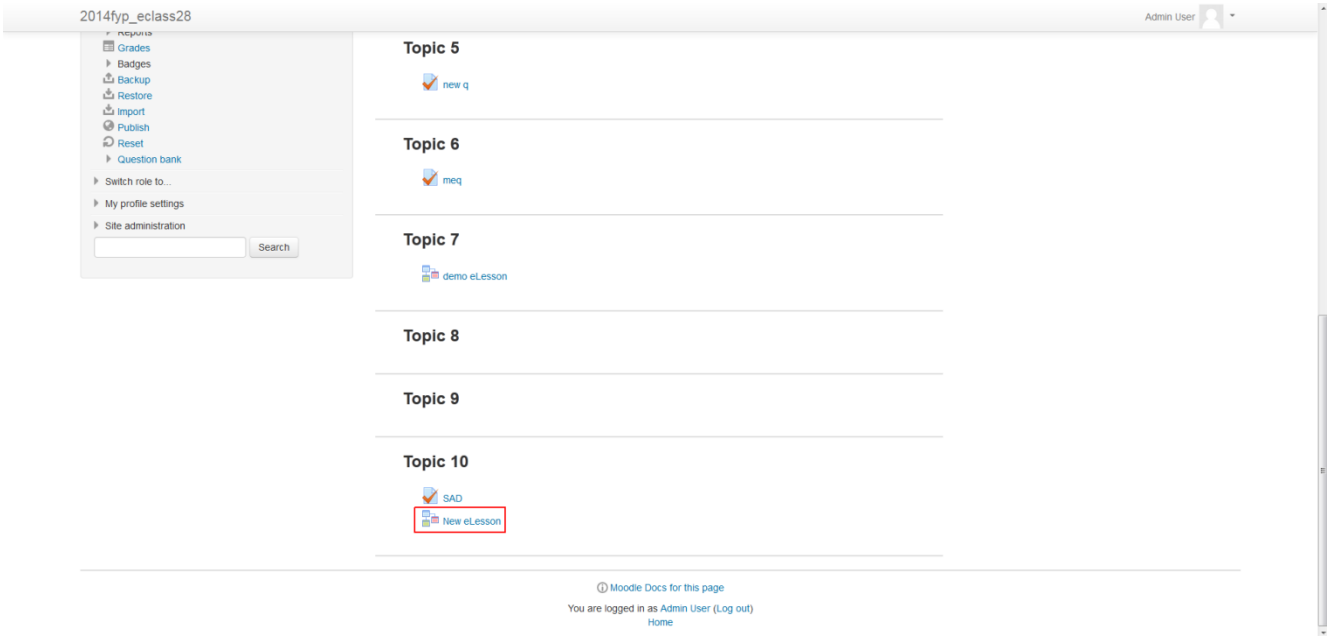
Moreover, there are no any images in the html. Although we have found the file which saved the path of images, we cannot get the path of this images. We will find another way to solve this problem in future.

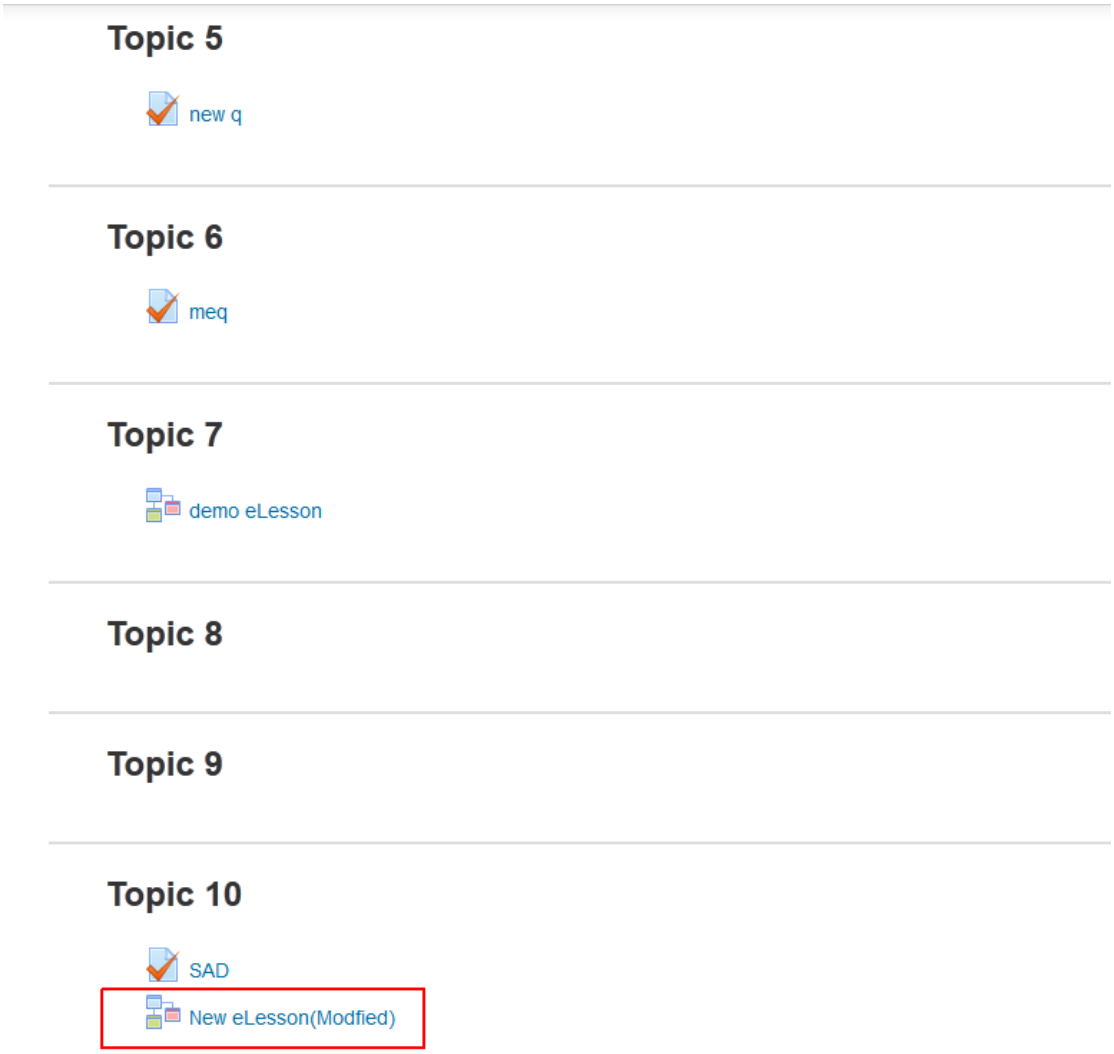
In addition, there are no any background in the html file, because there are some default element to show the background color in Power Point. We only can get the background colors which are not the default color in Power Point. Although we can

get some of the background color, we have not set it in the html file. It is because there are no any background color in some of the slide, which set the default color in Power Point. In order to avoid some of the slide have set the background color but some of them have not set, we do not set the background color for all of the slide in html file. We will find the default color to solve this problem in future.

Test Case for Web Moodle

Test Case	Description	Expected Result	Result
1	Add new eLesson into course	The new eLesson is added successfully	Shown success
2	Modify eLesson	eLesson is modified successfully	Shown success





Test Case	Description	Expected Result	Result
3	Delete eLesson	The eLesson is deleted successfully	Shown success
4	Add a PowerPoint to eLesson	eLesson converted the PowerPoint into online version	Shown success

Topic 5

 new q

Topic 6

 meq

Topic 7

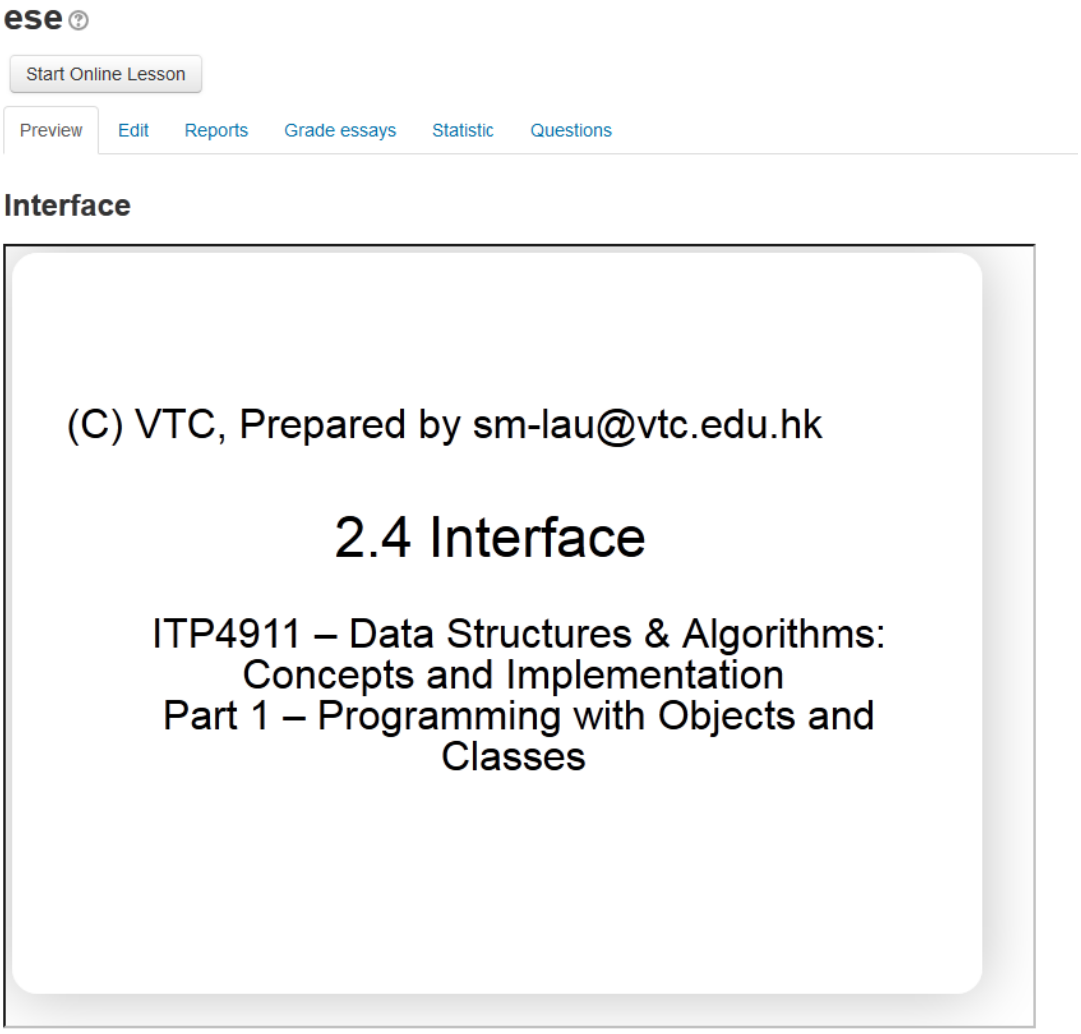
 demo eLesson

Topic 8

Topic 9

Topic 10

 SAD



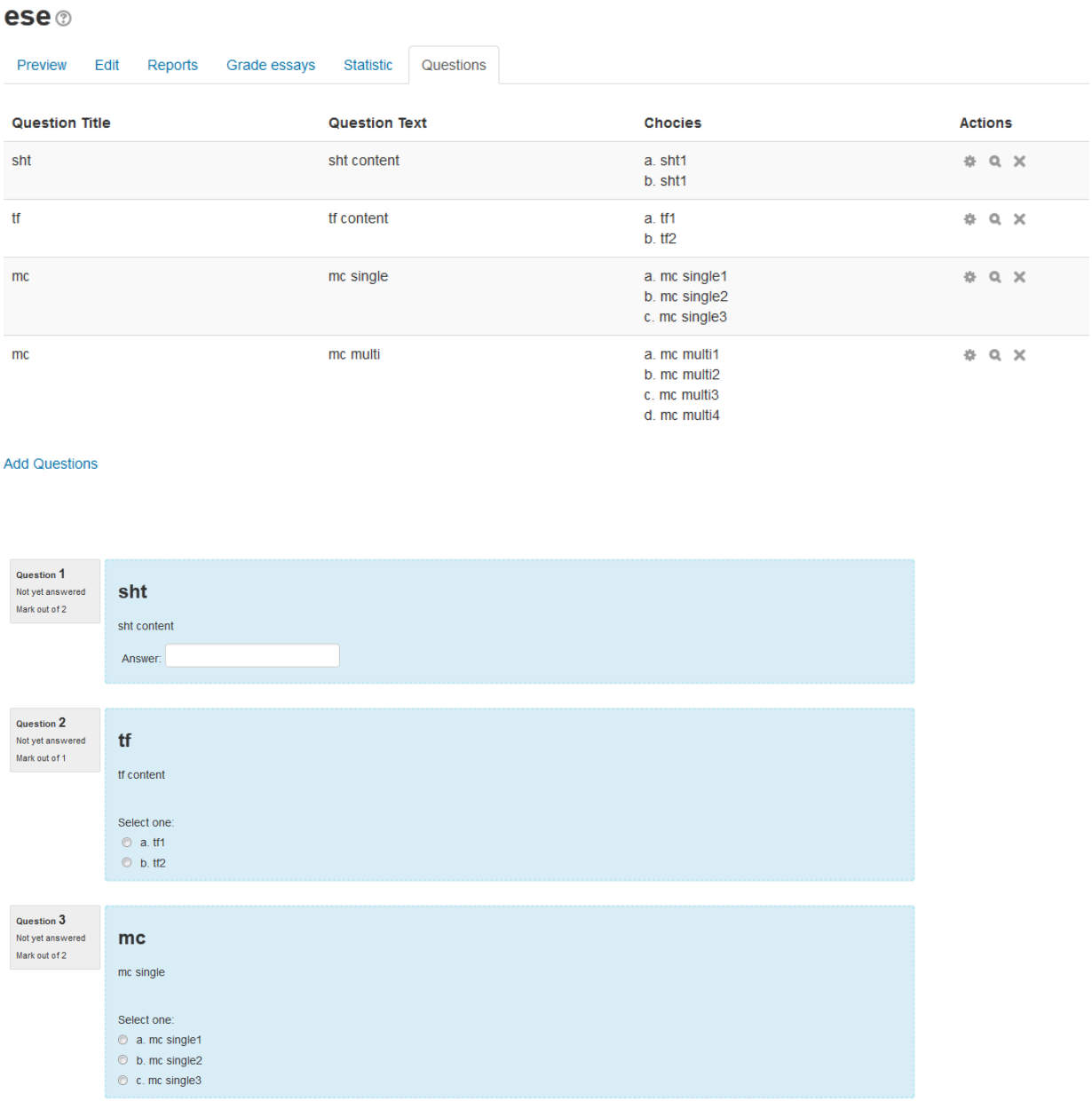
Test Case	Description	Expected Result	Result
5	Upload multiple PowerPoint to eLesson	The eLesson should be able to display more than one PowerPoint in one eLesson	Shown success
6	Delete specific page number of the PowerPoint page	eLesson should be able to delete specific page number but not the whole PowerPoint page.	Shown success

Inserted page: test

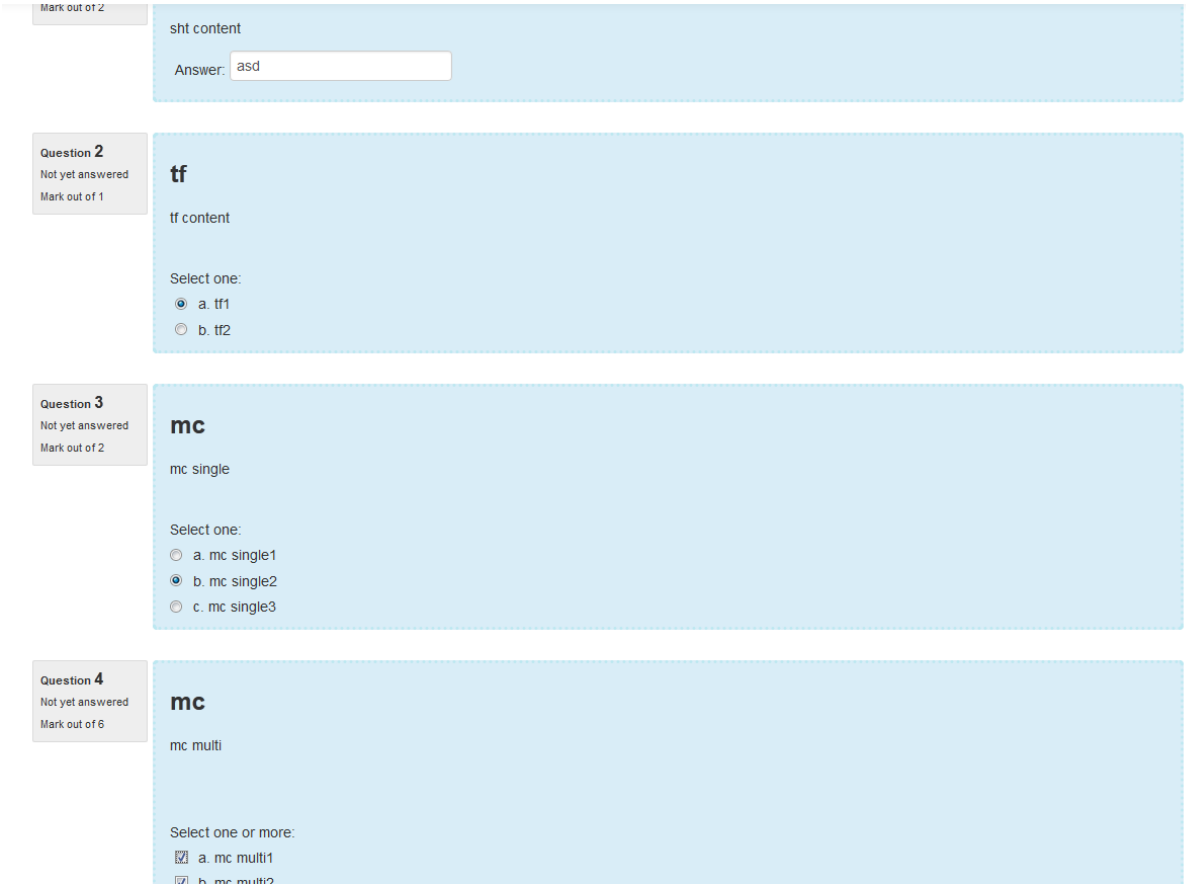
Page title	Page type	Jumps	Actions	Attach Question in Page
Interface	PowerPoint	Next page Previous page	Add a new page...	Attach Question
Interface	PowerPoint	Next page Previous page	Add a new page...	Attach Question
Interface	PowerPoint	Next page Previous page	Add a new page...	Attach Question
Interface	PowerPoint	Next page Previous page	Add a new page...	Attach Question
Interface	PowerPoint	Next page Previous page	Add a new page...	Attach Question
test	PowerPoint	This page	Add a new page...	Attach Question
test	PowerPoint	This page	Add a new page...	Attach Question
test	PowerPoint	This page	Add a new page...	Attach Question
test	PowerPoint	This page	Add a new page...	Attach Question
Interface	PowerPoint	Next page Previous page	Add a new page...	Attach Question
Interface	PowerPoint	Next page Previous page	Add a new page...	Attach Question
Interface	PowerPoint	Next page Previous page	Add a new page...	Attach Question
Interface	PowerPoint	Next page Previous page	Add a new page...	Attach Question

Page title	Page type	Jumps	Actions	Attach Question in Page
Interface	PowerPoint	Next page Previous page	Add a new page...	Attach Question
Interface	PowerPoint	Next page Previous page	Add a new page...	Attach Question
Interface	PowerPoint	Next page Previous page	Add a new page...	Attach Question
Interface	PowerPoint	Next page Previous page	Add a new page...	Attach Question
Interface	PowerPoint	Next page Previous page	Add a new page...	Attach Question
test	PowerPoint	This page	Add a new page...	Attach Question
test	PowerPoint	This page	Add a new page...	Attach Question
test	PowerPoint	This page	Add a new page...	Attach Question
Interface	PowerPoint	Next page Previous page	Add a new page...	Attach Question
Interface	PowerPoint	Next page Previous page	Add a new page...	Attach Question
Interface	PowerPoint	Next page Previous page	Add a new page...	Attach Question
Interface	PowerPoint	Next page Previous page	Add a new page...	Attach Question

Test Case	Description	Expected Result	Result
7	Add question in question bank	The question should be added successfully	Shown success
8	Attach question in PowerPoint Page	The question should be attached successfully	Shown success



Test Case	Description	Expected Result	Result
9	Answer questions	The result should be submitted to moodle server	Shown success
10	Check Answer for the questions	The score should be calculated correctly	Shown success



Mark 0 out of 2

sht content

Answer: asd

Your answer is incorrect.
The correct answer is: sht1

Question 2
Completed
Mark 1 out of 1

tf

tf content

Select one:
- a. tf1
- b. tf2

Your answer is correct.
r1
The correct answer is: tf1

Question 3
Completed
Mark 2 out of 2

mc

mc single

Select one:
- a. mc single1
- b. mc single2
- c. mc single3

Your answer is correct.
r2

Detailed Project Plan

Time schedule:

The project will be finished before 13th of May.

Hardware needed:

Hardware application of mobile or tablet like iPad with the newest version IOS 8 and Android Mobile.

Software tools needed:

Eclipse for the Android application development

Xcode for the IOS application

Wamp for the service operation

Netbeans for php development

etc: like Notepad++ , c#.

Main stage:

* Planning:

* Building skill set of objective c, swift and web service principle.

* Understanding and learning the architecture of Moodle system which allows to get the data from Moodle and develops module or plugin of Moodle.

* Analysis:

* Capture the functional and non-functional requirements of Moodle system.

* Finding solutions which allow a PowerPoint application viewing in the webpage. Turning it into image or retrieving data from PowerPoint and rebuilding the PowerPoint slide in html and css format.

* Design:

* User interface design of the Moodle client application.

* User interface design of the Moodle plugin.

* Implementation:

* Construct the mobile apps and Moodle plugins.

* Test the application with different devices, refine its performance and reliability.

Main Deliverables:

* Fully functional Moodle mobile application in IOS and Android platform.

* Reduce the dependence of using PowerPoint file.

* Increase the usability of Moodle lesson.

Program Listing:

```
<?php

require_once(dirname(__FILE__) . '/../../config.php');

class questionlib{

        public $option=array('a. ','b. ','c. ','d. ','e. ','f. ','g. ','h. ','i. ','j. ','k. ','l. ','m. ','n.
','o. ','p. ','q. ','r. ','s. ','t. ','u. ','v. ','w. ','x. ','y. ','z. ');

        public $qdivstyle='padding:0px 0px 0px 5px;';

        public $qboxstyle='border: medium dotted rgb(188, 232, 241);
background-color: rgb(217, 237, 247); padding: 8px 35px 8px 14px; border-radius:
4px; margin: 0px 8em 30px 8.5em;line-height:180%;';

    function initinfobox($qno,$score){

        $infobox='

<div id="info_'.$qno.'" style="background-color: rgb(238, 238, 238); border: 1px solid
rgb(220, 220, 220); padding: 0.5em; width: 7em; border-radius: 2px; float: left;">

<b style="font-size:11px">Question <span
style="font-size:16px">'.$qno.'</span></b><br>

<span name="isanswered_q'.$qno.'" style="font-size:11px">Not yet
answered</span><br>
```

```
<span style="font-size:11px">Mark <span name="score_q'.$qno.'"></span> out of
'.$score.'</span>
</div>';
        return $infobox;
    }

    function getoption(){
        return $this->option;
    }

    function inittruefalse($qno,$score,$qtitle,$qtext,$qchoices,$multians=null){
        $chocies='';
        $i=0;
        foreach($qchoices as $key=>$val){
            $chocies.='<div class="qdiv_'.$qno.'_'.$i.'"
style="'.$this->qdivstyle.'">';
```

```
                $chocies.='<input class="truefalse_q'.$qno.'" type="radio"
name="truefalse_q'.$qno.'" value="'.$i.'"
required="required"><label>'.$this->option[$i++].$qchoices[$key]->answer.'</label>'
;

                $chocies.='</div>';

            }

            $qbox=$this->initinfobox($qno,$score);

            $qbox.='

<div id="qbody_'.$qno.'" style="'.$this->qboxstyle.'">

<b><h3>'.$qtitle.'</h3></b>

                        '.$qtext.'<br>

<span>Select one:</span><br>

                        '.$chocies.'

</div>';

            return $qbox;

        }
```

```
function initmc($qno,$score,$qtitle,$qtext,$qchoices,$multians){

        $chocies='';

        $i=0;

        foreach($qchoices as $key=>$val){

            if($multians){

                $chocies.='<div class="qdiv_'.$qno.'_'.$i.'"
style="'.$this->qdivstyle.'">';

                $chocies.='<input class="mc_q'.$qno.'" type="checkbox"
name="mc_q'.$qno.'[]" value="'.$i.'"
><label>'.$this->option[$i++].$qchoices[$key]->answer.'</label>';

                $chocies.='</div>';

            }else{

                $chocies.='<div class="qdiv_'.$qno.'_'.$i.'">';

                $chocies.='<input class="mc_q'.$qno.'" type="radio"
name="mc_q'.$qno.'[]" value="'.$i.'" required="required"
><label>'.$this->option[$i++].$qchoices[$key]->answer.'</label>';

                $chocies.='</div>';

            }
```

```
            }

            $qbox=$this->initinfobox($qno,$score);

            if($multians){

                $selectstmnt='<span>Select one or more:</span><br>';

            }else{

                $selectstmnt='<span>Select one:</span><br>';

            }

            $qbox.='

<div id="qbody_'.$qno.'" style="'.$this->qboxstyle.'">

<b><h3>'.$qtitle.'</h3></b>

                        '.$qtext.'<br>

                        '.$selectstmnt.'

                        '.$chocies.'

</div>';

            return $qbox;

        }
```

```
        function
initshortasnwer($qno,$score,$qtitle,$qtext,$qchoices=null,$multians=null){

        $qbox=$this->initinfobox($qno,$score);

        $qbox.='

<div id="qbody_'.$qno.'" style="'.$this->qboxstyle.'">

<b><h3>'.$qtitle.'</h3></b>

                        '.$qtext.'

<div class="qdiv_'.$qno.'" style="'.$this->qdivstyle.'">Answer: <input
class="shtans_q'.$qno.'" type="text" name="shtans_q'.$qno.'" required="required"
></div>

</div>';

        return $qbox;

    }


    function getnoofquestions($pageid){

        global $DB;
```

```
$noofquestions=$DB->get_records('elesson_questions',array('pageid'=>$pageid));
    }

    function getquestionname($qno,$qtype){
        $qarray=array("shtans_q","truefalse_q","mc_q");
        $name=$qarray[$qtype].$qno;
        return $name;
    }

    function generateonclick($url){
        $url="onclick=\"javascript:void
window.open('".$url."','1424795719496','width=700,height=500,toolbar=0,menubar
=0,location=0,status=1,scrollbars=1,resizable=1,left=0,top=0');return false;\"";
        return $url;
    }
```

```
    function getcorrectanswers($qid){
        global $DB;
        $qdb=$DB->get_record('elesson_questions',array('id'=>$qid));
$qchs=$DB->get_records('elesson_qchoices',array('qid'=>$qid,'score'=>$qdb->fullsco
re));
        return $qchs;
    }

    function getcorrectanswers_string($qid){
        global $DB;
        $qdb=$DB->get_record('elesson_questions',array('id'=>$qid));
        $anstring='';
        if($qdb->multianswer==0){
            $qchs=$this->getcorrectanswers($qid);
        }else{
```

```
                $qchs=$DB->get_records_select('elesson_qchoices', 'score <> ? and
qid = ?',array('score'=>0,'qid'=>$qid));

            }

            foreach($qchs as $chs){

                $anstring.=$chs->answer.',';

            }

            $anstring=substr($anstring, 0, -1);

            return $anstring;

        }
```

```
public class MainActivity extends ActionBarActivity implements ActionBar.TabListener {
     private ViewPager viewPager;
     private ActionBar actionBar;
     private TabsPagerAdapter pageAdapter;
     private  LayoutInflater inflater;
     private Login loginTask=null;
    private LoadPage loadPageTask=null;
    private Context context;
    private UserBean ub;
    private CoursesBean csb;
    private CourseAlertBuilder coursealertbuilder;
    private ArrayList<View> fragmentGroup;
    private ActionBar.Tab sectiontab;
    @Override
    protected void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);

        context = this;
        inflater =  LayoutInflater.from(context);
         loginTask = new Login();
        loadPageTask= new LoadPage();
        actionBar = getSupportActionBar();
        actionBar.setDisplayShowTitleEnabled(false);
        setContentView(R.layout.login);

}
    private class CallWebService extends AsyncTask<String, Integer, String> {...}
    public void loginBtnClick(View v) {
```

```
    public void loginBtnClick(View v) {

        View loginView = (View)v.getParent();
        EditText username= (EditText)loginView.findViewById(R.id.login_username);
        EditText password= (EditText)loginView.findViewById(R.id.login_password);
        String usernameString = username.getText().toString();
        String passwordString = password.getText().toString();
        if( loginTask.getStatus() == AsyncTask.Status.PENDING  ||  (loginTask.getStatus() == AsyncTask.Sta
            loginTask=new Login();
            loginTask.execute(usernameString,passwordString);
        }
```

```
}
private class Login extends AsyncTask<String, Integer, String> {
    protected String doInBackground(String... args) {
        LoginWebService login = new LoginWebService();
        String token="";
        try {
            token = login.run(args[0], args[1]);
        }
        catch(Exception e){
            Log.d("Login","Login Error");
        }
        if (token != "") {
            try {
                GetUserbyUserName guu = new GetUserbyUserName();
                ub = guu.run(token, args[0]);
                String id = ub.getId();
```

```
                GetCourseByUserIdWebService gcbui = new GetCourseByUserIdWebService();
                csb = gcbui.run(token, id);


                GetCourseContent gcc = new GetCourseContent();
                for (int i = 0; i < csb.getCoursesbean().size(); i++) {
                    SectionsBean ssb = gcc.run(token, csb.getCoursesbean().get(i).getCourseid());
                    csb.getCoursesbean().get(i).setSsb(ssb);
                }
            }
            catch(Exception e){Log.d("GetData",e.getMessage()); }

        }

        return token;
    }
    protected void onProgressUpdate(Integer... progress) {
        // setProgressPercent(progress[0]);
    }

    protected void onPostExecute(String result) {
        Log.d("Login",result);

        if (result != "") {
                setContentView(R.layout.activity_main);
                inflater = LayoutInflater.from(context);
                //Initilization
                viewPager = (ViewPager) findViewById(R.id.pager);
```

```
if (result != "") {
        setContentView(R.layout.activity_main);
        inflater = LayoutInflater.from(context);
        // Initilization
        viewPager = (ViewPager) findViewById(R.id.pager);
       viewPager.setOffscreenPageLimit(4);
        actionBar = getSupportActionBar();
        actionBar.setDisplayShowTitleEnabled(false);
        pageAdapter = new TabsPagerAdapter(getSupportFragmentManager());

        viewPager.setAdapter(pageAdapter);
        actionBar.setHomeButtonEnabled(false);
        actionBar.setNavigationMode(ActionBar.NAVIGATION_MODE_TABS);
        String[] tabs = {"Courses","Material","Profiles", "------"};
        // Adding Tabs
        for (String tab_name : tabs) {
            actionBar.addTab(actionBar.newTab().setText(tab_name)
                    .setTabListener((ActionBar.TabListener) context));

        }
        sectiontab = actionBar.getTabAt(1);
        actionBar.removeTabAt(1);
    //actionBar.addTab(new Tab(),1);
        getSupportActionBar().show();
        setActionBarListener();

  loadPageClick(null);
}
```

```
for(int i=0;i<csb.getCoursesbean().size();i++) {
    final View childview = inflater.inflate(R.layout.courseslayout, null);
    llo.addView(childview, i);
    TextView tv = (TextView) childview.findViewById(R.id.courseText);
    CourseBean cb=csb.getCoursesbean().get(i);
    final int index=i;
    tv.setText(cb.getFullname());
    LinearLayout button= (LinearLayout) childview.findViewById(R.id.linearlayout_coursespress);
    button.setOnClickListener((v) -> {
            //View sessionView = inflater.inflate(R.layout.login, null);
        //  ViewGroupUtils.replaceView(childview,sessionView);x
            coursealertbuilder= new CourseAlertBuilder(context,csb.getCoursesbean().get(index),u
            if(!actionBar.getTabAt(1).getText().equals("Material"))
            actionBar.addTab(sectiontab,1);
            actionBar.setSelectedNavigationItem(1);

    });
    count++;
    ViewGroup.MarginLayoutParams mlParams = (ViewGroup.MarginLayoutParams)childview.getLayoutPa
    mlParams.topMargin = 50;
    childview.setLayoutParams(mlParams);
}
for(int i=count;i<20;i++) {
    try {
        View childview = inflater.inflate(R.layout.courseslayout, null);
        llo.addView(childview, i);
        TextView tv = (TextView) childview.findViewById(R.id.courseText);
        tv.setText(i + 1 + "");
        tv.setTextColor(Color.WHITE);
```

```
    }

    private void loadPageClick(View v){
        if(v == null){
            if( loadPageTask.getStatus() == AsyncTask.Status.PENDING  ||  (loadPageTask.getStatus() == AsyncT
                loadPageTask=new LoadPage();
                loadPageTask.execute();
            }
        }
    }
    private class LoadPage extends AsyncTask<String, Integer, String> {
        protected String doInBackground(String... args) {return "";}
        protected void onPostExecute(String result) {
            try {
                fragmentGroup=new ArrayList<View>();
                View v=viewPager.findViewWithTag("FirstPage");
                View v1=viewPager.findViewWithTag("SecondPage");
                View v2=viewPager.findViewWithTag("ThirdPage");
                View v3=viewPager.findViewWithTag("ForthPage");
                fragmentGroup.add(v);
                fragmentGroup.add(v1);
                fragmentGroup.add(v2);
                fragmentGroup.add(v3);
                LinearLayout llo = (LinearLayout) v.findViewById(R.id.linearlayout_courses);
                int count=0;
                for(int i=0;i<csb.getCoursesbean().size();i++) {
                    final View childview = inflater.inflate(R.layout.courseslayout, null);
                    llo.addView(childview, i);
                    TextView tv = (TextView) childview.findViewById(R.id.courseText);
```

```
<?php

class JamesDOMApi {

   static function getFirstEleByTagName(DOMNode $domEle, $name = NULL) {
      $nodeList = $domEle->getElementsByTagName($name);
      return ($nodeList->length > 0) ? $nodeList->item(0) : NULL;
   }

   /**
    *
    * @param DOMElement $domEle
    * @param string $name
    * @return DOMElement
    */
   static function getEleFirstEleByTagName(DOMElement $domEle, $name = NULL) {
      $nodeList = $domEle->getElementsByTagName($name);
      return ($nodeList->length > 0) ? $nodeList->item(0) : NULL;
   }

   /**
    *
    * @param DOMDocument $domDoc
    * @param string $name
    * @return DOMElement
    */
   static function getDocFirstEleByTagName(DOMDocument $domDoc, $name =
NULL) {
      $nodeList = $domDoc->getElementsByTagName($name);
      return ($nodeList->length > 0) ? $nodeList->item(0) : NULL;
   }

   static function getFirstEleByTagNameArray(DOMElement $domEle, array
$nameArray = NULL) {
      foreach ($nameArray as $name) {
         $domEle = JamesDOMApi::getEleFirstEleByTagName($domEle, $name);
         if ($domEle === NULL) {
```

```
                return NULL;
            }
        }
        return $domEle;
    }

    static function getEleAtt(DOMElement $domEle, $attribute) {
        return $domEle->getAttribute($attribute);
    }

}
?>
<?php

require_once('locallib/Remove.php');
class Unzip{
    function doUnzip($path){
        $zip = new ZipArchive;
        $res = $zip->open($path);
        if ($res == TRUE) {
            $remove=new Remove;
            $remove->rrmdir('temp/pptx');
            $check = $zip->extractTo('temp/pptx/');
            $zip->close();
        } else {
            echo 'file not find!<br/>';
        }
    }
}
?>
<?php
class textSet{

    private $text="";

    function textSet($textAr){
        for($i=0;$i<count($textAr);$i++){
```

```
        $textArray=$textAr[$i];

if((!empty($textArray['color']))||(!empty($textArray['size']))||(!empty($textArray['bo
ld']))||(!empty($textArray['italics']))||(!empty($textArray['underline']))){
            $this->text.="<span style=¥" ";
            if((!empty($textArray['color']))){
               $this->text.="color:#".$textArray['color']."; ";
            }
            if((!empty($textArray['size']))){
               $this->text.="font-size:".(((int)$textArray['size'])/80)/16*100 . "; ";
            }
            if((!empty($textArray['bold']))){
               $this->text.="font-weight:".'bold'."; ";
            }
            if((!empty($textArray['italics']))){
               $this->text.="font-style: italic; ";
            }
            if((!empty($textArray['underline']))){
               $this->text.="text-decoration: underline; ";
            }
            $this->text.=" ¥" >";
            if(empty($textArray['text'])){
               $this->text.="</span><br/>";
            }else{
               $this->text.=$textArray['text'];
               $this->text.="</span >";
            }
         }else{
            $this->text.=$textArray['text'];
         }
      }
   }

   function getText(){
      return $this->text;
   }
```

```
}
?>
<?php
require_once('locallib/createDiv.php');
require_once('locallib/OOXML_bg.php');
class slideToHtml {

    private $create;
    private $document;

    function toHtml($index){
        $filename="temp/json/slide".$index.".json";
        $this->create="<div class=¥"slide¥"";
        $bg=new OOXML_bg;
        $bgVal=$bg->getBg($index);
        if(!empty($bgVal)){
            $this->create.="style=¥"";
            $this->create.=$bgVal;
            $this->create.="¥" ";
        }
        $this->create.="/>";
        try{
            $file=fopen($filename, 'r');
        }
        catch(Exception $e){
            echo "File not fond!";
            return;
        }
        $line_of_text = fgets($file);
        $data = json_decode($line_of_text, true);
        for($i=0;$i<count($data);$i++){
            $temp= new createDiv($data[$i], $index);
            $this->create.=$temp->returnHtml();
        }
        $this->create.="</div>";
        return $this->create;
    }
```

```
}
?>
<?php
require_once('locallib/textSet.php');
class setArea{
   private $returnD=" ";
   private $algn='';
   function setArea($textTetail, $type){
      $this->algn="left";
      if($type=='ctrTitle' || $type=='subTitle' || $type=='title'){
         $this->algn='center';
      }
      if((!empty($textTetail['algn']))){
         $this->algn=$textTetail['algn'];
         switch ($this->algn) {
            case 'r' :
               $this->algn="right";
               break;
            case 'l' :
               $this->algn="left";
               break;
            case 'ctr' :
               $this->algn="center";
               break;
         }
      }

      $textData=$textTetail['textContent'];
      $text="";
      if((!empty($textData))){
         for($i=0;$i<count($textData);$i++){
            $in=$textData[$i];
            $set=new textSet($in);
            $text=$set->getText();
         }
      }
```

```
        if(empty($text)){
            $text='';
        }
        $this->returnD.=$text;
    }

    function getArea(){
        return $this->returnD;
    }

    function getAlgn(){
        return $this->algn;
    }
}
?>
<?php
class Remove{
    function rrmdir($dir) {
        if (is_dir($dir)) {
            $objects = scandir($dir);
            foreach ($objects as $object) {
                if ($object != "." && $object != "..") {
                    if (filetype($dir."/".$object) == "dir") {
                        $remove=new Remove;
                        $remove->rrmdir($dir."/".$object);
                    }else{
                        unlink($dir."/".$object);
                    }
                }
            }
            reset($objects);
            $remove=rmdir($dir);
        }
    }
}
?>
<?php
```

```
require_once('locallib/lib/JamesDOMApi.php');
class Relation{
   function printRel($rels){
      $res = $rels->getElementsByTagName("Relationship");
      $rel = array();
      foreach ($res as $domNode) {
         $tagName = $domNode->tagName;
         $target = $domNode->getAttribute("Target");
         $Id = $domNode->getAttribute("Id");
         $array=array('id'=>$Id,'target'=>$target);
         array_push($rel, $array);
      }
      return $rel;
   }
}
?>
<?php
require_once('lib/Displayable.php');
require_once('lib/JamesDOMApi.php');

class PPTXSlideAnimation implements Displayable {

   /**
    *
    * @var array of DOMElement
    */
   public $animaArray;

   function __construct(DOMNodeList $cTnList) {
      $this->animaArray = static::getAnimaArray($cTnList);
   }

   /**
    *
    * @param DOMNodeList $cTnList
    * @return array
    */
```

```
static function getAnimaArray(DOMNodeList $cTnList) {
    $cTnArray = [];
    foreach ($cTnList as $cTn) {
        $presetID = $cTn->getAttribute('presetID');
        if ($presetID) {
            $cTnArray[] = $cTn;
        }
    } // foreach
    return $cTnArray;
}

public function displayData() {
    ?>
    <div>
    <?php
        foreach ($this->animaArray as $anima) {
            $id = $anima->getAttribute('id');
            $presetID = $anima->getAttribute('presetID');
            $presetClass = $anima->getAttribute('presetClass');
            $presetSubtype = $anima->getAttribute('presetSubtype');
            $repeatCount = $anima->getAttribute('repeatCount');
            $noteType = $anima->getAttribute('nodeType');

            $a = ['stCondLst', 'cond'];
            $cond = JamesDOMApi::getFirstEleByTagNameArray($anima, $a);
            $delay = ($cond === NULL) ? NULL : $cond->getAttribute('delay');

            $a = ['cBhvr', 'tgtEl', 'spTgt'];
            $spTgt = JamesDOMApi::getFirstEleByTagNameArray($anima, $a);
            $spid = ($spTgt === NULL) ? NULL : $spTgt->getAttribute('spid');
            ?>
    <table border="1">
            <?php $this->drawHeader(); ?>
            <tbody>
            <tr>
                <td><?php echo $id; ?></td>
                <td><?php echo $spid; ?></td>
```

```
                <td><?php echo $presetID; ?></td>
                <td><?php echo $presetClass; ?></td>
                <td><?php echo $presetSubtype; ?></td>
                <td><?php echo $repeatCount; ?></td>
                <td><?php echo $noteType; ?></td>
                <td><?php echo $delay; ?></td>
            </tr>
            </tbody>
    </table>
            <?php
        }
        ?>
    </div>
    <?php
}

public function drawHeader() {
    ?>
    <thead>
    <tr>
            <th>id</th>
            <th>spid</th>
            <th>presetID</th>
            <th>presetClass</th>
            <th>presetSubtype</th>
            <th>repeatCount</th>
            <th>nodeType</th>
            <th>delay</th>
    </tr>
    </thead>
    <?php
}

static function echo_cTn(DOMElement $node) {
    $id = $node->getAttribute('id');
    $presetID = $node->getAttribute('presetID');
    $nodeType = $node->getAttribute('$nodeType');
```

```
        ?>
        cTn
        <table border="1">
        <tr>
                <th>id</th>
                <th>presetID</th>
                <th>nodeType</th>
        </tr>
        <tr>
                <td><?php echo $id; ?></td>
                <td><?php echo $presetID; ?></td>
                <td><?php echo $nodeType; ?></td>
        </tr>

        </table>
        <?php
    }

}
?>
<?php

require_once('lib/JamesDOMApi.php');

require_once('OOXML_sp.php');
require_once('OOXML_pic.php');
require_once('OOXML_cxnSp.php');
require_once('PPTXSlideAnimation.php');
require_once('Color.php');

class PPTXSlide {

    private $document;
    private $animation;
    private $presentation;
    public static $array = [
            'p:nvGrpSpPr',
```

```
            'p:grpSpPr',
            'p:sp',
            'p:pic',
            'p:cxnSp',
        ];

        function __construct(DOMDocument $doc) {
            //parent::__construct();
            $this->document = $doc;
        }

        static function slide_api($slideXMLFileName,$file) {
            $doc = new DOMDocument();
            $doc->load($slideXMLFileName);

            $node = JamesDOMApi::getDocFirstEleByTagName($doc, 'cSld');
            if ($node === NULL) {
                return NULL;
            }
            $temp=array();
            $bg = JamesDOMApi::getEleFirstEleByTagName($node, 'bg');
            $bgPr = JamesDOMApi::getEleFirstEleByTagName($node, 'bgPr');
            if($bgPr!=null){
            $bgA=array();
                $domNodeList = $bgPr->childNodes;
                foreach ($domNodeList as $domNode) {
                    $tagName = $domNode->tagName;
                    switch ($tagName) {
                        case'a:solidFill':
                            $solidFill = JamesDOMApi::getEleFirstEleByTagName($node,
'solidFill');
                            $srgbClr=JamesDOMApi::getEleFirstEleByTagName($solidFill,
'srgbClr');
                            $schemeClr=JamesDOMApi::getEleFirstEleByTagName($solidFill,
'schemeClr');
                            if($srgbClr!=null){
                                $color=$srgbClr->getAttribute('val');
```

```
                $bgA=array('type'=>'color', 'val'=>$color);
            }else{
                $scheme = $schemeClr->getAttribute('val');
                $lum = $schemeClr->getElementsByTagName('lumMod');
                $lumMod='';
                foreach($lum as $n){
                    $lumMod=$n->getAttribute('val');
                }
                if($lumMod==null){
                    $lumMod=0;
                }
                $col=new Color;
                $color = $col->getColor($scheme, $lumMod);
                $bgA=array('type'=>'color', 'val'=>$color);
            }
            break;

        case'a:blipFill':
            $blipFill = JamesDOMApi::getEleFirstEleByTagName($node, 'blipFill');
            $blip=JamesDOMApi::getEleFirstEleByTagName($blipFill, 'blip');
            $rId=$blip->getAttribute('r:embed');
            $bgA=array('type'=>'img', 'val'=>$rId);
            break;
      }
   }
   $bgJ=fopen('temp/json/'.$file.'_bg.json', 'w');
   fwrite($bgJ, json_encode($bgA));
   fclose($bgJ);
}
$node = JamesDOMApi::getEleFirstEleByTagName($node, 'spTree');
if ($node === NULL) {
   return NULL;
}

$domNodeList = $node->childNodes;
foreach ($domNodeList as $domNode) {
   $tagName = $domNode->tagName;
```

```
            switch ($tagName) {
                case'p:sp':
                    $ooxml_sp = new OOXML_sp($domNode);
                    $dataIn=$ooxml_sp->displayData();
                    array_push($temp,$dataIn);
                    break;
                case'p:pic':
                    $ooxml_pic = new OOXML_pic($domNode);
                    $dataIn=$ooxml_pic->displayData();
                    array_push($temp,$dataIn);
                    break;
                case'p:cxnSp':
                    $ooxml_cxnSp = new OOXML_cxnSp($domNode);
                    $ooxml_cxnSp->displayData();
                    break;
            }
        }// foreach
        $allFp=fopen('temp/json/'.$file.'.json', 'w');
        fwrite($allFp, json_encode($temp));
        fclose($allFp);
    }

}
?>
<?php

require_once('lib/JamesDOMApi.php');
require_once('OOXML.php');

class PPTXPresentation {

    private $document;
    private $cx, $cy, $type;
    private $notesSz_cx, $notesSz_cy;
    /*
     * <p:sldSz cx="9144000" cy="6858000" type="screen4x3"/>
     * <p:notesSz cx="6858000" cy="9144000"/>
```

```
     */

    function __construct() {
        //parent::__construct();
    }

    static function presentation_api($presentationXMLFileName) {
        $doc = new DOMDocument();
        $doc->load($presentationXMLFileName);

        //echo '<br>';
        $node = JamesDOMApi::getDocFirstEleByTagName($doc, 'sldIdLst');

        $domNodeList = $node->childNodes;
    }

}
?>
<?php
require_once('locallib/Unzip.php');
require_once('locallib/createSlide.php');
require_once('locallib/Remove.php');
class pptx2html{
    private $slide;
    private $name;

    function toHtml($name,$pPath,$fp){
        //error_reporting(E_ERROR | E_PARSE);
        $file_name = $name;
        $dp=$fp."/".substr((string)$file_name,0,-5);
        $remove=new Remove;
        if(scandir('temp')==false){
            mkdir('temp');
        }
        $new_file_name="temp/temp.zip";
        $path="temp/temp.zip";
        $remove->rrmdir("temp/json");
```

```
        mkdir("temp/json");
        if($pPath !=null){
            if(copy($pPath, $path)){
                $unzip=new Unzip();
                $unzip->doUnzip($path);
                $pptx=new createSlide;
                $pptx->doToHtml($dp);
                $this->name=$pptx->getHtmlName();
                $this->slide=$pptx->getHtmlContent();
                return $this->slide;
            }else{
                return "Error";
            }
        }
        return "Error";
    }
    function getSlideName(){
        return $this->name;
    }
    function getSlideCon(){
        return $this->slide;
    }
}
?>
<?php
class getPo{
    private ps='';

    function setPo($po){
        $count=0;
        $checkS=false;
        $xy='';
        $xy2='';
        for($i=0;$i<count($po);$i++){
            $char=substr($po,$i,($i+1));
            if(($char==' ' || $char=='' ) && $count<2){
                $checkS=true;
```

```
            $count++;
        }elseif(($char==' ' || $char=='' ) && $count<=2){
            $checkS=false;
        }elseif($count==1){
            $xy.=$char;
        }elseif($count==2){
            $xy2.=$char;
        }
      }
      $this->ps=((int)$xy/(int)$xy2)*100;
   }

   function getPo(){
      return $this->ps;
   }
}
?>
<?php

require_once('lib/JamesDOMApi.php');
require_once("locallib/Color.php");

class OOXML_Text {

   /**
    * <a:t> nodeValue
    * @var string
    */
   public $text;      //
   /**
    * <a:rPr sz="2000">
    * @var int
    */
   public $size;
   /**
    * <a:rPr>
    * <a:solidFill>
```

```
     * <a:srgbClr val="FF0000">
     * @var string
     */
    public $color;
    /**
     *
     * @var int
     */
    public $bold;
    public $italics;
    public $underline;

    public function __construct(DOMElement $domNode_r) {
        $this->text = JamesDOMApi::getEleFirstEleByTagName($domNode_r,
't')->nodeValue;
        $color_tag = array('rPr', 'solidFill', 'srgbClr');
        $color_tag2 = array('rPr', 'solidFill', 'schemeClr');


        if ($srgbClr = JamesDOMApi::getFirstEleByTagNameArray($domNode_r,
$color_tag)) {
            $this->color = $srgbClr->getAttribute('val');
        }

        if($schemeClr = JamesDOMApi::getFirstEleByTagNameArray($domNode_r,
$color_tag2)){
            $scheme = $schemeClr->getAttribute('val');
            $lum = $schemeClr->getElementsByTagName('lumMod');
            $lumMod='';
            foreach($lum as $n){
                $lumMod=$n->getAttribute('val');
            }
            if($lumMod==null){
                $lumMod=0;
            }
            $col=new Color;
            $this->color = $col->getColor($scheme, $lumMod);
```

```
        }

        if ($rPr = JamesDOMApi::getEleFirstEleByTagName($domNode_r, 'rPr')) {
            $this->size = $rPr->getAttribute('sz');
            $this->bold = $rPr->getAttribute('b');
            $this->italics = $rPr->getAttribute('i');
            $this->underline = $rPr->getAttribute('u');
        }
    }

    public function __toString() {
        return $this->text;
    }

}
?>
<?php
require_once('lib/JamesDOMApi.php');
require_once('OOXML.php');
require_once('OOXML_p.php');
require_once('OOXML_Text.php');
require_once("locallib/Color.php");

class OOXML_sp extends RotatableOOXML {
    public $id;
    public $name;
    public $prst;
    public $color;
    public $type; // p:sp, p:nvSpPr, p:nvPr, p:ph
    /**
     *
     * @var OOXML_p
     */
    public $array_p = [];

    function __construct(DOMElement $domNode_sp) {
        parent::__construct($domNode_sp);
```

```
        // ===== id, name
        $cNvPr = JamesDOMApi::getEleFirstEleByTagName($domNode_sp, 'cNvPr');
        if ($cNvPr === NULL) {
            //throw new Exception('Missing "cNvPr" Tag.');
            $this->id = NULL;
            $this->name = NULL;
        } else {
            $this->id = $cNvPr->getAttribute('id');
            $this->name = $cNvPr->getAttribute('name');
        }

        // ===== prst

        $array = ['spPr', 'prstGeom'];
        $prstGeom = JamesDOMApi::getFirstEleByTagNameArray($domNode_sp,
$array);
        $this->prst = ($prstGeom) ? $prstGeom->getAttribute('prst') : NULL;

        // ===== color
        $color_tag = array('style', 'lnRef', 'srgbClr');
        $color_tag2 = array('style', 'lnRef', 'schemeClr');
        $srgbClr=JamesDOMApi::getFirstEleByTagNameArray($domNode_sp,
$color_tag);
        $schemeClr=JamesDOMApi::getFirstEleByTagNameArray($domNode_sp,
$color_tag2);
        if ($srgbClr != null) {
            $this->color = $srgbClr->getAttribute('val');
        }elseif($schemeClr != null){
            if(is_object($schemeClr)){
                $scheme = $schemeClr->getAttribute('val');
            }
            //$lum = $schemeClr->getElementsByTagName('shade');
            $lumMod='';
            foreach($lum as $n){
                $lumMod=$n->getAttribute('val');
            }
            if($lumMod==null){
```

```
            $lumMod=0;
        }
        $col=new Color;
        $this->color = $col->getColor($scheme, $lumMod);
        if($this->color==null){
            $this->color=$scheme;
        }
    }

    // ===== type
    $nvPr = JamesDOMApi::getEleFirstEleByTagName($domNode_sp, 'nvPr');
    if ($nvPr) {
        $ph = JamesDOMApi::getEleFirstEleByTagName($nvPr, 'ph');
        $this->type = ($ph) ? $ph->getAttribute('type') : NULL;
    }
    // ===== p
    $this->setup_array_p($domNode_sp);
}

private function setup_array_p(DOMElement $domNode_sp) {
    $txBody = JamesDOMApi::getEleFirstEleByTagName($domNode_sp, 'txBody');
    if ($txBody === NULL) {
        return NULL;
    }
    $pList = $txBody->getElementsByTagName('p');
    $this->array_p = [];
    foreach ($pList as $p) {
        $this->array_p[] = new OOXML_p($p);
    }// end foreach
}

public function displayData() {
    $textData=array('id'=>$this->id,'eType'=>'sp', 'name'=>$this->name,
'type'=>$this->type,'color'=>$this->color, 'prst'=>$this->prst,
'array_p'=>$this->array_p, 'x'=>$this->x, 'y'=>$this->y , 'cx'=>$this->cx,
'cy'=>$this->cy, 'rot'=>$this->rot);
    return $textData;
```

```
    }

    protected function drawHeader() {
        ?>
        <th>id</th>
        <th>name</th>
        <th>prst</th>
        <th>type</th>
        <th>array p</th>
        <th>x</th>
        <th>y</th>
        <th>cx</th>
        <th>cy</th>
        <th>rot</th>
        <th>rot(degree)</th>
        <?php
    }

}
?>
<?php
require_once('lib/JamesDOMApi.php');
require_once('OOXML.php');

class OOXML_pic extends RotatableOOXML {

    // p:pic, p:blipFill, a:blip => r:embed=
    public $rId;

    function __construct(DOMElement $domNode_pic) {
        parent::__construct($domNode_pic);
        $this->setupRId($domNode_pic);
    }

    /*private function setupSrc() {

    }*/
```

```
    private function setupRId(DOMElement $domNode_pic) {
        $rId = NULL;
        if (($blipFill = JamesDOMApi::getEleFirstEleByTagName($domNode_pic,
'blipFill')) !== NULL) {
            if (($blip = JamesDOMApi::getEleFirstEleByTagName($blipFill, 'blip')) !==
NULL) {
                $rId = $blip->getAttribute('r:embed');
            }
        }
        $this->rId = $rId;
    }

    function displayData() {
        $textData=array('id'=>$this->rId, 'eType'=>'pic','x'=>$this->x, 'y'=>$this->y ,
'cx'=>$this->cx, 'cy'=>$this->cy, 'rot'=>$this->rot);
        return $textData;
    }

    protected function drawHeader() {
        ?>
        <th>id</th>
        <th>x</th>
        <th>y</th>
        <th>cx</th>
        <th>cy</th>
        <th>rot</th>
        <th>rot(degree)</th>
        <?php
    }

}
?>
<?php
require_once('lib/JamesDOMApi.php');
require_once('OOXML_Text.php');
```

```
class OOXML_p {

    /**
     * @var OOXML_Text
     */
    public $textContent;
    public $algn;
    public $marL;
    public $indent;
    public $char;
    public $lvl;

    /**
     * case: arabicPeriod - 1.
     * case: arabicParenR - 1)
     * case: alphaLcParenR - a)
     * case: alphaLcPeriod - a.
     * case: alphaUcPeriod - A.
     * @var type
     */
    public $type;

    public function __construct(DOMElement $domNode_p) {
        if ($pPr = JamesDOMApi::getEleFirstEleByTagName($domNode_p, 'pPr')) {
            $this->algn = $pPr->getAttribute('algn');
            $this->marL = $pPr->getAttribute('marL');
            $this->indent = $pPr->getAttribute('indent');
            $this->lvl = $pPr->getAttribute('lvl');

            if ($buChar = JamesDOMApi::getEleFirstEleByTagName($pPr, 'buChar')) {
                $this->char = $buChar->getAttribute('char');
            }

            if ($buAutoNum = JamesDOMApi::getEleFirstEleByTagName($pPr,
'buAutoNum')) {
                $this->type = $buAutoNum->getAttribute('type');
            }
```

```
    }
    // pPr

    $pChildNodes = $domNode_p->childNodes;
    $this->setupTextContent($pChildNodes);
}

public function setupTextContent(DOMNodeList $pChildNodes) {
    $this->textContent = [[]];
    $row = 0;
    foreach ($pChildNodes as $node) {
        $tagName = $node->tagName;
        switch ($tagName) {
            case 'a:r':
                $this->textContent[$row][] = $this->getText($node);
                break;
            case 'a:br':
                $row++;
                $this->textContent[] = [];
                break;
        }
    }// end foreach
}

/**
 *
 * @param DOMElement $r
 * @return OOXML_Text
 */
private function getText(DOMElement $r) {
    $t = JamesDOMApi::getEleFirstEleByTagName($r, 't');
    return ($t === NULL) ? NULL : new OOXML_Text($r);
}

public function displayData() {
    ?>
    <div>
```

```
        OOXML_p displayData()
    <table border = "1">
        <?php $this->displayHeader(); ?>
        <tr>
        <td><?php echo $this->algn; ?></td>
        <td><?php echo $this->marL; ?></td>
        <td><?php echo $this->indent; ?></td>
        <td><?php echo $this->type; ?></td>
        <td><?php echo $this->char; ?></td>
        </tr>
        <tr>
        <td colspan="5"><?php $this->displayTextContent(); ?></td>
        </tr>
    </table>
    </div>
    <?php
}

private function displayHeader() {
    ?>
    <tr>
    <th>algn</th>
    <th>marL</th>
    <th>indent</th>
    <th>type</th>
    <th>char</th>
    </tr>
    <?php
}

private function displayTextContent() {
    ?>
    <table border = "1">
    <tr><?php $this->displayTextContentHeader(); ?></tr>
    <?php
        $row_num = 0;
        foreach ($this->textContent as $row) {
```

```
            $this->displayTextContentRow($row, $row_num);
            $row_num++;
        }
        ?>
    </table>
    <?php
  }

  private function displayTextContentHeader() {
    ?>
    <th>Row</th>
    <th>Text</th>
    <th>Color</th>
    <th>Size</th>
    <th>Bold</th>
    </tr>
    <?php
  }

  private function displayTextContentRow($row, $row_num) {
    foreach ($row as $text) {
      ?>
    <tr>
    <td><b><?php echo $row_num; ?></b></td>
    <td><?php echo $text->text; ?></td>
    <td><?php echo $text->color; ?></td>
    <td><?php echo $text->size; ?></td>
    <td><?php echo $text->bold; ?></td>
    </tr>
    <?php
    } // foreach
  }

}
?>
<?php
class OOXML_Img{
```

```
    public function getImg($rId,$index){
        $jsonP='temp/json/_rels'.$index.'.json';
        $json=fopen($jsonP,'r');
        $line_of_text = fgets($json);
        $data = json_decode($line_of_text, true);
        $val=$this->getEleByRId($rId, $data);
        return $val;
    }

    private function getEleByRId($rId,$array){
        $imgP='';
        foreach($array as $rel){
            if($rel['id']==$rId){
                $imgP=substr($rel['target'],3);
                break;
            }
        }
        return $imgP;
    }
}
?>
<?php
require_once('lib/JamesDOMApi.php');
require_once('OOXML.php');
require_once('OOXML_p.php');

class OOXML_cxnSp extends LocatableOOXML {

    public $id = NULL;
    public $name = NULL;
    public $flipH = NULL;
    public $flipV = NULL;

    /**
     * <p:spPr>
     * <a:prstGeom prst="line">
     * @var type
```

```
     */
    public $prst = NULL;
    public $fmla = NULL;

    /**
     * <a:avLst>
     * <a:gd name="adj1" fmla="val 16326"/>
     * @var type
     */
    public $headEndType = NULL;
    public $tailEndType = NULL;

    function __construct(DOMElement $domNode_cxnSp) {
        parent::__construct($domNode_cxnSp);
        // ===== id, name
        $this->setupIdName($domNode_cxnSp);

        // ===== flipH, $flipV
        $this->setupFlip($domNode_cxnSp);

        // ===== prst, fmla
        $this->setupPrstFmla($domNode_cxnSp);

        // ===== headEndType, tailEndType
        $this->setupEndType($domNode_cxnSp);
    }

    private function setupIdName(DOMElement $domNode_cxnSp) {
        $cNvPr = JamesDOMApi::getEleFirstEleByTagName($domNode_cxnSp, 'cNvPr');
        if ($cNvPr === NULL) {
            //throw new Exception('Missing "cNvPr" Tag.');
            $this->id = NULL;
            $this->name = NULL;
        } else {
            $this->id = $cNvPr->getAttribute('id');
            $this->name = $cNvPr->getAttribute('name');
        }
```

```
    }

    private function setupPrstFmla(DOMElement $domNode_cxnSp) {
        $array = ['spPr', 'prstGeom'];
        $prstGeom = JamesDOMApi::getFirstEleByTagNameArray($domNode_cxnSp,
$array);
        $this->prst = ($prstGeom) ? $prstGeom->getAttribute('prst') : NULL;

        if ($prstGeom) {
            $array = ['avLst', 'gd'];
            $gd = JamesDOMApi::getFirstEleByTagNameArray($prstGeom, $array);
            $this->fmla = ($gd) ? $gd->getAttribute('fmla') : NULL;
        }
    }

    private function setupFlip(DOMElement $domNode_cxnSp) {
        $array = ['spPr', 'xfrm'];
        $xfrm = JamesDOMApi::getFirstEleByTagNameArray($domNode_cxnSp, $array);
        if ($xfrm) {
            $this->flipH = $xfrm->getAttribute('flipH');
            $this->flipV = $xfrm->getAttribute('flipV');
        }
    }

    private function setupEndType(DOMElement $domNode_cxnSp) {
        $array = ['spPr', 'ln', 'headEnd'];
        $headEnd = JamesDOMApi::getFirstEleByTagNameArray($domNode_cxnSp,
$array);
        $this->headEndType = ($headEnd) ? $headEnd->getAttribute('type') : NULL;

        $array = ['spPr', 'ln', 'tailEnd'];
        $tailEnd = JamesDOMApi::getFirstEleByTagNameArray($domNode_cxnSp,
$array);
        $this->tailEndType = ($tailEnd) ? $tailEnd->getAttribute('type') : NULL;
    }

    public function displayData() {
```

```
    ?>
    <div>
    <table border = "1">
        <tr><?php $this->drawHeader(); ?></tr>
        <tr>
        <td><?php echo($this->id); ?></td>
        <td><?php echo($this->name); ?></td>
        <td><?php echo($this->flipV); ?></td>
        <td><?php echo($this->flipH); ?></td>
        <td><?php echo($this->prst); ?></td>
        <td><?php echo($this->fmla); ?></td>
        <td><?php echo($this->headEndType); ?></td>
        <td><?php echo($this->tailEndType); ?></td>
        <td><?php echo($this->x); ?></td>
        <td><?php echo($this->y); ?></td>
        <td><?php echo($this->cx); ?></td>
        <td><?php echo($this->cy); ?></td>
        </tr>
    </table>
    </div>
    <?php
}

protected function drawHeader() {
    ?>
    <th>id</th>
    <th>name</th>
    <th>flipV</th>
    <th>flipH</th>
    <th>prst</th>
    <th>fmla</th>
    <th>headEndType</th>
    <th>tailEndType</th>
    <th>x</th>
    <th>y</th>
    <th>cx</th>
    <th>cy</th>
```

```
        <?php
    }

}
?>
<?php
require_once('locallib/OOXML_Img.php');
class OOXML_bg{
    function getBg($index){
        $jP='temp/json/slide'.$index.'_bg.json';
        $json=fopen($jP,'r');
        if(empty($json)){
            return 'background-color: #ffffff';
        }else{
            $json=fopen($jP,'r');
            $line_of_text = fgets($json);
            $data = json_decode($line_of_text, true);
            if($data['type']=='img'){
                $rId=$data['val'];
                $img = new OOXML_Img;
                $val='background-image: url('.$img->getImg($rId,$index).');';
                return $val;
            }elseif($data['type']=='color'){
                $color=$data['val'];
                return 'background-color: #'.$color.';';
            }
        }
    }
}
?>
<?php
require_once('lib/JamesDOMApi.php');

abstract class LocatableOOXML {

    public $x;
    public $y;
```

```
    public $cx;
    public $cy;

    function __construct(DOMElement $domNode) {
        $this->setupLocation($domNode);
    }

    private function setupLocation(DOMElement $domNode) {
        $x = $y = $cx = $cy = NULL;
        if (($xfrm = JamesDOMApi::getEleFirstEleByTagName($domNode, 'xfrm')) !==
NULL) {
            if (($off = JamesDOMApi::getEleFirstEleByTagName($xfrm, 'off')) !== NULL) {
                $x = $off->getAttribute('x');
                $y = $off->getAttribute('y');
            }
            if (($ext = JamesDOMApi::getEleFirstEleByTagName($xfrm, 'ext')) !== NULL) {
                $cx = $ext->getAttribute('cx');
                $cy = $ext->getAttribute('cy');
            }
        }
        $this->x = $x;
        $this->y = $y;
        $this->cx = $cx;
        $this->cy = $cy;
    }

    abstract public function displayData();

    abstract protected function drawHeader();
}

abstract class RotatableOOXML extends LocatableOOXML {

    public $rot;

    function __construct(DOMElement $domNode) {
        parent::__construct($domNode);
```

```
        $this->setup_rot($domNode);
    }

    private function setup_rot(DOMElement $domNode_pic) {
        $rot = NULL;
        if (($spPr = JamesDOMApi::getEleFirstEleByTagName($domNode_pic,
'spPr')) !== NULL) {
            if (($xfrm = JamesDOMApi::getEleFirstEleByTagName($spPr, 'xfrm')) !== NULL)
{
                $rot = $xfrm->getAttribute('rot');
            }
        }
        $this->rot = $rot;
    }

}
?>
<?php
require_once('locallib/PPTXSlide.php');
require_once('locallib/slideToHtml.php');
require_once('locallib/Remove.php');
require_once('locallib/Relation.php');

class createSlide{

    private $htmlN=Array();

    private $htmlC=Array();

    private $htmlCode="PPTXTOHTMLFYP";

    function doToHtml($dp){
        error_reporting(E_ERROR | E_PARSE);
        $remove=new Remove;
        $div=scandir($dp);
        if(count($div)==0){
            mkdir($dp);
```

```
        }else{
            $remove->rrmdir($dp);
            mkdir($dp);
        }
        $fMdir="temp/pptx/ppt/media";
        $tMdir=$dp."/media";
        $div=scandir($fMdir);
        if(count(scandir($fMdir))!=0){
            rename($fMdir, $tMdir);
        }
        $hp='';
        $head="<!doctype html><html><head><meta name=¥"viewport¥"
content=¥"width=1200¥" />";
        $head.="<link href=¥"../present/fathom.sample.css¥"";
        $head.="rel=¥"stylesheet¥" type=¥"text/css¥" /><link
href=¥"../present/pptx.css¥" rel=¥"stylesheet¥">";
        $path="temp/pptx/ppt/slides";
        $files = scandir($path);
        $index=1;
        $loop=true;
        $head.="</head>";
        $head.="<body><div id=¥"presentation¥" style=¥"center¥">";
        $end="</div></body></html>";
        while($loop==true){
            $check=false;
            for($i=0;$i<count($files);$i++){
                if($files[$i]=="slide".$index.".xml"){
                    $check=true;
                }
            }
            if($check!=true){
                $loop=false;
            }else{
                $relsP=$path."/_rels/slide".$index.".xml.rels";
                $xml=new DOMDocument();
                $xml->load($relsP);
                $rel=new Relation;
```

```
            $temp=$rel->printRel($xml);
            $rels=fopen('temp/json/_rels'.$index.'.json','w');
            fwrite($rels, json_encode($temp));
            fclose($rels);
            $html1= new slideToHtml;
            PPTXSlide::slide_api($path."/slide".$index.".xml","slide".$index);
            $slide='<!--'.$this->htmlCode.'-->';
            $slide.=$html1->toHtml($index);
            $hp=$dp.'/slide'.$index.'.html';
            $slide.='<!--'.$this->htmlCode.'-->';
            $body=$head.$slide.$end;
            $html=fopen($hp, 'a');
            fwrite($html,$body);
            fclose($html);
            $arrayN='slide'.($index);
            $this->htmlN[$index]=$hp;
            $this->htmlC[$arrayN]=$body;
            $index++;
          }
        }
    }

    function getHtmlName(){
        return $this->htmlN;
    }

    function getHtmlContent(){
        return $this->htmlC;
    }

}
?>
<?php
require_once('locallib/setArea.php');
require_once('locallib/Create.php');
class createDiv{
```

```
        private $in="";
        private $textIndex=1;

        function createDiv($div, $index){
            $eTypr='';
            $eType=$div['eType'];
            $create = new Create;
            switch ($eType) {
                case'sp':
                    $this->textIndex=$create->sp($div,$this->textIndex);
                    $this->in.=$create->getIn();
                    $this->textIndex++;
                    break;
                case'pic':
                    $create->pic($div, $index);
                    $this->in.=$create->getIn();
                    break;
            }
        }

        function returnHtml(){
            return $this->in;
        }

}
?>
<?php
require_once('locallib/OOXML_Img.php');
require_once('locallib/setArea.php');
class Create{

    private $in='';

    function sp($div,$index){
        $x=0;
        $y=0;
        $w=0;
```

```
$h=0;
$rot=0;
$prst='';
$color='';
if((!empty($div['x']))){
   $x=(int)((int)$div['x'])/9144000*100;
}
if((!empty($div['y']))){
   $y=(int)((int)$div['y'])/6858000*100;
}
$pw="100%";
if((!empty($div['cx']))){
   $w=((int)$div['cx'])/9144000*100+8;
   $pw=((int)$w-10)."%";
//720
}
if((!empty($div['cy']))){
   $h=(int)$div['cy']/6858000*100;
//540
}
if($div['cy']==$div['cy']){
   $h=(int)$w*720/540;
}
if((!empty($div['rot']))){
   $rot=((int)((int)$div['rot'])/60000);
}
if((!empty($div['prst']))){
   $prst=$div['prst'];
}
if((!empty($div['color']))){
   $color=$div['color'];
}
$this->in='';
$textTetail=$div['array_p'];
$this->in.='<div';
if((!empty($div['prst']))){
   $this->in.=" id='".$prst."'";
```

```
}
if((!empty($div['type']))){
   $this->in.=' class="'.$div['type'].'"';
}
$this->in.=' style=" position: absolute;';
if((!empty($div['color']))){
   $this->in.=' background-color: #'.$color.';';
}
if((!empty($div['rot']))){
   $this->in.='    ';
   $this->in.=' transform: rotate('.$rot.'deg);';
}
if((!empty($div['y']))||(!empty($div['x']))){
   $this->in.="height: ".$h."%; width: ".$w."%;";
}
if((!empty($div['y']))&&(!empty($div['x']))){
   $this->in.=" left :".$x.'%; top: '.$y."%;";
}else{
   if($div['type']=='ctrTitle'){
      if((!empty($div['y']))){
         $this->in.=' top: '.$y.'%; left: 7%;';
      }else if((!empty($div['x']))){
         $this->in.=' left: '.$x.'%; top: 27%;';
      }else{
         $this->in.=" top: 27%; left: 7%;";
      }
   }else if($div['type']=='subTitle'){
      if((!empty($div['y']))){
         $this->in.=' top: '.$y.'%; left: 13%;';
      }else if((!empty($div['x']))){
         $this->in.=' left: '.$x.'%; top: 50%;';
      }else{
         $this->in.=' top: 50%; left: 13%;';
      }
   }else if($div['type']=='title'){
       if((!empty($div['y']))){
         $this->in.=' top: '.$y.'%; left: 4.4%;';
```

```
            }else if((!empty($div['x']))){
                $this->in.=' left: '.$x.'%; top: 3.5%;';
            }else{
                $this->in.=' top: 3.5%; left: 4.4%;';
            }
        }else{
            $size=4/3*100;
            $this->in.="height: 58.2%; width: 87.4%;";
            $this->in.="font-size: 200%;";
            $this->in.=" top: 20.6%; left: 4.4%;";
        }
    }
    $this->in.=' ">';
    for($i=0;$i<count($textTetail);$i++){
        $tType=$textTetail[$i]['type'];
        $char=$textTetail[$i]['char'];
        $lvl=$textTetail[$i]['lvl'];
        $area=new setArea($textTetail[$i], $div['type']);
        $con=$area->getArea();
        $marL=$textTetail[$i]['marL'];
        if($char=='n'){
            $char="□";
        }elseif($con==' '||$con==''){
            $char='';
        }$this->in.="<!--name:".$div['name'].';prst='.$prst.'; type:'.$div['type'].";
char:".$char."; content:".$con.";-->";
        if($con==' '||$con==''){
            $this->in.='<pre></pre>';
        }elseif(substr($div['name'],0,5)=='Title'){
            $this->in.="<pre style=¥" text-align:
".$area->getAlgn().";¥">".$area->getArea()." </pre>";
        }elseif(!empty($char)){
            if($con==' '||$con==''){
                $this->in.='<pre></pre>';
            }else{
                if((int)$lvl==1){
```

```
                $this->in.="<table style=¥" text-align: ".$area->getAlgn().";¥"><tr><td
style=¥" width: 5%¥"></td><td>".$char."</td><td style=¥"width:
1%¥"></td><td><pre >".$area->getArea()." </pre></td></tr></table>";
            }else{
                $this->in.="<table style=¥" text-align:
".$area->getAlgn().";¥"><tr><td>".$char."</td><td style=¥"width:
1%¥"></td><td><pre >".$area->getArea()." </pre></td></tr></table>";
            }
        }
    }elseif(!(empty($tType))){
        if($tType=="arabicPeriod"){
            $this->in.="<table style=¥" text-align: ".$area->getAlgn().";
¥"><tr><td>".$index.". </td><td style=¥"width: 1%¥"></td><td><pre
>".$area->getArea()." </pre></td></tr></table>";
            $index++;
        }else{
            $chrInd=chr($index+64);
            $content=$area->getArea();
            $this->in.="<table style=¥" text-align: ".$area->getAlgn().";
¥"><tr><td>".$chrInd.". </td><td style=¥"width: 1%¥"></td><td><pre
>".$area->getArea()." </pre></td></tr></table>";
            $index++;
        }
    }else{
        if(substr($div['name'],0,19)=='Content Placeholder' && ($div['type']=='' ||
$div['type']==' ')){
            $content=$area->getArea();
            if($marL!=null){
                $this->in.="<pre style=¥" text-align: ".$area->getAlgn().";
¥">".$area->getArea()." </pre>";
            }elseif(!empty($content)){
                if((int)$lvl==1){
                    $this->in.="<table style=¥" text-align: ".$area->getAlgn().";
¥"><tr><td style=¥" width: 5%¥"><pre></pre></td><td><pre style=¥" margin:
0px¥">-    </pre></td><td style=¥" width: 1%¥"></td><td><pre >".$area->getArea()."
</pre></td></tr></table>";
                }else{
```

```
                    $this->in.="<table style=¥" text-align: ".$area->getAlgn().";
¥"><tr><td>"• </td><td style=¥" width:1%¥"></td><td><pre >".$area->getArea()."
</pre></td></tr></table>";
                }
            }else{
                $this->in.='<pre></pre>';
            }
        }else{
            $this->in.="<pre style=¥" text-align:
".$area->getAlgn().";¥">".$area->getArea()." </pre>";
        }
      }
    }
    $this->in.='</div>';
    return $index;
  }

  function pic($div, $index){
    $rId=$div['id'];
    $x=0;
    if((!empty($div['x']))){
      $x=(int)((int)$div['x'])/9144000*100;
        //x=720
    }
    $y=0;
    if((!empty($div['y']))){
      $y=(int)((int)$div['y'])/6858000*100;
      //y=540
    }
    $w=((int)$div['cx'])/9144000*100;
    $h=((int)$div['cy'])/6858000*100;
    $rot=0;
    if((!empty($div['rot']))){
      $rot=((int)((int)$div['rot'])/60000);
    }
    $image = new OOXML_Img;
    $src=$image->getImg($rId,$index);
```

```
        $this->in="<img class=¥"pic¥" src=¥"".$src."¥" style=¥"position: absolute;";
        $this->in.=" transform: rotate(".$rot."deg); left :".$x.'%; top: '.$y."%; width:
".$w."%; height:".$h."%;" ;
        $this->in.="¥" />";
        $this->in.='<!--cx='.$div['cx'].'; cy='.$div['cy'].'; w= '.$w.'; h= '.$h.'; -->';
    }

    function getIn(){
        return $this->in;
    }

}
?>
<?php
class Color{
    public function getColor($name,$lum) {
        if($lum!=0){
            $lumMod = $lum/100000;
        }
        $c1='';
        $c2='';
        $c3='';
        switch ($name) {
            case'bg1':
                $c1=255;
                $c2=255;
                $c3=255;
                break;
            case'tx1':
                if($lumMod==0){
                    $c1=0;
                    $c2=0;
                    $c3=0;
                }else{
                    $c1=255;
                    $c2=255;
                    $c3=255;
```

```
    }
    break;
case'bg2':
    $c1=231;
    $c2=230;
    $c3=230;
    break;
case'tx2':
    if($lum!=0){
        $lumMod= 1 - $lumMod;
    }
    $c1=68;
    $c2=84;
    $c3=106;
    break;
case'accent1':
    $c1=91;
    $c2=155;
    $c3=213;
    break;
case'accent2':
    $c1=237;
    $c2=125;
    $c3=49;
    break;
case'accent3':
    $c1=165;
    $c2=165;
    $c3=165;
    break;
case'accent4':
    $c1=255;
    $c2=192;
    $c3=0;
    break;
case'accent5':
    $c1=68;
```

```
            $c2=144;
            $c3=196;
            break;
        case'accent6':
            $c1=112;
            $c2=173;
            $c3=71;
            break;
    }
    if($lum!=0){
        $c1*=$lumMod;
        $c2*=$lumMod;
        $c3*=$lumMod;
        if($name!='bg'){
            $c1=255-(int)$c1;
            $c2=255-(int)$c2;
            $c3=255-(int)$c3;
        }
    }
    $color1=str_pad(dechex($c1), 2, "0", STR_PAD_LEFT);
    $color2=str_pad(dechex($c2), 2, "0", STR_PAD_LEFT);
    $color3=str_pad(dechex($c3), 2, "0", STR_PAD_LEFT);
    return $color1.$color2.$color3;
  }
}
?>
```

References

Datypic, I. (n.d.). OOXML pml-slide.xsd - Complete documentation and samples.

Retrieved from datypic: http://www.datypic.com/sc/ooxml/s-pml-slide.xsd.html

Office Open XML (OOXML) - Presentations - Presentation.(n.d.). Retrieved from Office Open XML (OOXML): http://officeopenxml.com/prPresentation.php

Socket.IO. (n.d.). Retrieved from Socket.IO: http://socket.io/

wikipedia. (2014, 9 30).Office_Open_XML. Retrieved from wikipedia: http://en.wikipedia.org/wiki/Office_Open_XML