**CSB1**

**Distributed teaching tool for COMP2611**

by

Rajanala Vivek, To Man Yin, Wong Siu Fai, Yu Pui Yin

Advised by

Prof. Bensaou Brahim

Submitted in partial fulfillment

of the requirements for COMP4988/COMP4981

in the

Department of Computer Science and Engineering

The Hong Kong University of Science and Technology

2015-2016

Name of student: \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ Signature: \_\_\_\_\_\_\_\_\_\_\_\_ Date: \_\_\_\_\_\_\_\_\_\_

Name of student: \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ Signature: \_\_\_\_\_\_\_\_\_\_\_\_ Date: \_\_\_\_\_\_\_\_\_\_

Name of student: \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ Signature: \_\_\_\_\_\_\_\_\_\_\_\_ Date: \_\_\_\_\_\_\_\_\_\_

Name of student: \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ Signature: \_\_\_\_\_\_\_\_\_\_\_\_ Date: \_\_\_\_\_\_\_\_\_\_

# Content

[Content](#h.neefz5euhcjl)

[Abstract](#h.oa1396lop21r)

[Introduction](#h.5w9e2nputyzu)

[Overview](#h.5cmmk2k3r2hh)

[Objective](#h.8ocage5ne81v)

[Data management](#h.du2n10q9gc9x)

[Design](#h.48w4j1y75kak)

[Support multiple user](#h.keqg9nbl6b9w)

[Functions](#h.o6qxxm2g6gib)

[User interface](#h.nlp1mts5z3b)

[Q&A](#h.od8n8us5w0hw)

[Knowledge Graph](#h.m1rmqj7v3mkp)

[Self Test](#h.qktggx5o7da)

[Challenges](#h.y8x8mh3u5ydk)

[Literature survey](#h.z1fq2xixfq5f)

[Visualizations of teaching materials](#h.7430ukzbr8cp)

[List view](#h.7m8qk07bfdtm)

[Categories](#h.6ad52sliliva)

[Tree structure](#h.wkmil5i2agb4)

[Design](#h.xj92pmdbeq6d)

[Database](#h.eaapberszi2s)

[Construct object tables](#h.nmq4pqrd4o02)

[Construct relationship tables](#h.x0n4nlfg28qm)

[Construct the constraints of the tables](#h.9nvb9u9g14st)

[Server](#h.e7tjco1xkn0t)

[Establish connection through WAN](#h.lmpo7p6o74i1)

[Multiple user support](#h.k3i78ekdqfye)

[Separate two database](#h.vjpd7ider0v2)

[Communication method](#h.9vp1ek96ukxj)

[Frequent access](#h.bvg3dawunf3j)

[Fast data access](#h.j27ph8o74br2)

[Streaming data access](#h.cm76x67yec1n)

[Construct data service interfaces](#h.tpdky5psapnk)

[Knowledge point manipulation](#h.ef8f8ayeooqs)

[Message box manipulation](#h.fgp1olkbuy86)

[Self-test manipulation](#h.yuquxej8t9fw)

[Course material management manipulation](#h.n9avrel210z1)

[User identity manipulation](#h.9jpkfeqoepsf)

[User interface](#h.7qt18cwzmar5)

[Pages](#h.l6gulplfoe9p)

[Knowledge point graph](#h.v4na2alxghgy)

[Self-test](#h.316bsfsgawtb)

[Q & A module](#h.n28c0z2oqnq7)

[Implementation](#h.g3k1jcwwca6o)

[Tasks & tools](#h.4bijnzx4r1bo)

[Libraries used](#h.av6rmphnn4mh)

[Testing](#h.6f35lw9k7xnp)

[Evaluation](#h.vvdae8u8ptqm)

[Server-client communication](#h.gy2po1gie9bi)

[User comments](#h.5e95s26gbb9r)

[Discussion](#h.653t727d8d5y)

[Conclusions](#h.9llf724kyvyw)

[References](#h.9ykjry67to4o)

# Abstract

This report describes a new visualization of lecture materials allows students can have a sufficient access to the related learning materials. The resulting web-based application of this project can be embedded to Canvas (student portal system in HKUST).

The application has 3 main sessions(knowledge point graph, online Q&A and self-test) which frequently interact with database.

The admin side of this application allows instructors to construct a knowledge point graph which can organize the lecture materials and map a session of notes (i.e. asset) to a specific topic. Students can have a shortcut to access the lecture materials according to the knowledge point and review them efficiently. Apart from the basic lecture materials (lecture notes), the application also allows to store the in-class Q&A and self-test as other materials for students.

With this application, students can study efficiently according to the well-organized materials and self-assessments. They follow the parent-child relations of the topics and learn from the stretch to the bottom.

Development was done using C#.NET, SQL, Windows Communication Fundamental, JavaScript and HTML.

# Introduction

## Overview

The learning management system, Canvas, using in HKUST has some limitations on content formats and data representation. A new visualization teaching tool called Knowledge Point Graph, which build relations between different parts of materials to a particular topic and state the relationship of each topics.

This project aims to develop a part of module for educational website, in order to provide a better content formats support and data management.

## Objective

## Data management

### Design

* + - Support the structure of knowledge point graph
    - Support the authentication of different users
    - Support the storage of data

### Support multiple user

* + - Support 200 users
    - Support frequencial access with 3~5 seconds delay time

### Functions

* + - Provide URL templates for user end to request
    - Access the data with a well-constructed format

## User interface

### Q&A

To implement a module called the “Q&A message box” for posting questions and handling replies.  
Students can ask questions about the course content  
Instructors can choose to give audio and/or text replies immediately or later  
The Q&A message box can be used both in-class and after-class

### Knowledge Graph

To implement a module that allows instructors to create a knowledge graph (containing knowledge points, also known as, concepts) for a particular course and also allows them to link each knowledge point to the associated resources (lecture notes, videos, questions and self-tests)

### Self Test

To create a module for instructors to create self-tests for knowledge points  
Students can use the self-tests for self-evaluation  
Instructors can use the self-tests for evaluating students’ progress and finding out common misconceptions/problems so as to improve the course

## Challenges

## 

## 

## Literature survey

### Visualizations of teaching materials

#### List view

List view is a common and simple visualization for a set of objects with same characteristic[[1]](#footnote-1). Many learning portals allow instructors to organize the learning materials in list structure, but a file list presentation may not help students to figure out the small topic inside a PDF/PPT file since some files contain a lot of contents.

In HKUST, one lecture takes 80 minutes in average with several of key knowledge included. Instructors often organize the notes in unit of lectures therefore students should clearly understand which lecture the knowledge key belongs to before they can reach the knowledge key.



#### Categories

Categorization is an organization methodology to separate the objects into different groups[[2]](#footnote-2). It has been widely used in data representation in web page and has a clear direction for user to reach the target. However, categorization only allows objects stay in one particular route for searching, it will be much complex if it support multi-parent-child relationship. This relationship is applicable in the visualization of learning materials since knowledge is correlated to each other.



#### Tree structure

In tradition, course materials present in portal with simple list view according to lecture time. It is clear but not sufficient for student to search and understand the relationship of the content. In the past 150-200 years, since the education system begins, humans adopt a “start from simple” learning strategy. Therefore, “knowledge” can be presented in a tree structure[[3]](#footnote-3).



To visualize the knowledge points (KPs), tree structure can provide a clear relationship and good organization diagram.

# Design

## Database

### Construct object tables

To implement this project, there are 5 basic objects are needed.

* Knowledge point
* File
* Q & A pair
* Self-test
* Result

The “File”, “Q & A pair” and “Self-test” are implemented to demonstrate the existing learning materials can be interacted with the “Knowledge Point Graph”.

To integrate the interaction of the objects above, 2 external objects are needed. “Asset” is an object belongs to “Knowledge point”. It consist of a pointer to a file object and has “start” and “end” indications to show which of the slides/pages related to the knowledge point. Another object “Result” stores the performance of each student for further analysis.

For a simple database design, each object will correlate to one table.

### Construct relationship tables

Under the database design concept[[4]](#footnote-4), the relationships between objects are also form a table. In this project, the relationships are the following:

* Knowledge point parent-child relationship
* Assets-Knowledge Point relationship

### Construct the constraints of the tables

Figure

## Server

### Establish connection through WAN

Internet Information Service (IIS)[[5]](#footnote-5) can help developers establish a server use a simple Microsoft Windows PC.

### Multiple user support

#### Separate two database

To reduce the traffic in one single database, this system separated 2 different databases which is authentication database and data model database. This may help if other systems have different authentication like HKUST CAS.

#### Communication method

According to the storyboard of user experience, the data server supports three types of communication method.

##### Frequent access

This is the most frequent access method. It allows users to keep communicating with the server for a short period (5 seconds expected) by refreshing.

Frequent access is used in in-class messaging (i.e. Q & A message box).

##### Fast data access

This method allows small amount of data transfer. It is the fastest communication method in this system.

Fast data access is mainly used for inserting self-test records, inserting new messages and logging into the system.

##### Streaming data access

This method takes a period of time (depends on the data size) for the communication to complete.

Streaming data access is used in file upload and knowledge point update.

### Construct data service interfaces

#### Knowledge point manipulation

Update knowledge points (Insert new or update existing knowledge point)

Parameters: updated knowledge points

Return knowledge points

Parameters: course ID

#### Message box manipulation

Insert new question

Parameters: course ID, user ID, question string

Get question list

Parameters: course ID

Insert answer to a question

Parameters: answer type, answer string, question ID

#### Self-test manipulation

Insert new self-test question

Parameters: knowledge point ID, question string, default answer key

Get self-test question set

Parameters: knowledge point ID

Answer a question

Parameters: self-test question ID, answer string, user ID

#### Course material management manipulation

Upload files

Parameters: file stream, course ID, path

#### User identity manipulation

Login

Parameters: username, password

Registration

Parameters: username, password, type of user

## User interface

The implementation of the interface elements will mainly be done using HTML and CSS and partly using jQuery. The functionality will be implemented using JavaScript and jQuery. The sending and getting of necessary data from the server will be implemented using Ajax.  
  
The backend and frontend setup are separated, here are the ideas and work that have been done.

### Pages

### Knowledge point graph

Basically there are 3 pages for the part of Knowledge Point Graph.  
 The view of Knowledge Point Graph  
 - View the existing Knowledge Point Graph on canvas, clicking on one Knowledge Point pops up a box to display all materials that are assigned to it  
 - Edit mode - adds association of materials from the right column to Knowledge Points  
 - click ‘Save’ to send the Knowledge Point Graph with assigned materials to the server  
 - clicking the column at the left brings up the Knowledge Point Graph Construction page  
  
Knowledge Point Graph Construction on canvas  
- Add new or delete existing Knowledge Points by selecting tools from the menu on the left column  
- Modify relationships between Knowledge Points  
- Click ‘Save’ to send the new Knowledge Point Graph to the server  
  
Tagging of materials  
- Open related viewer of the materials, i.e. PDF viewer, video viewer or audio player  
- A slide bar at the bottom to select the range of materials to tag  
- click ‘Save’ to send the new tagging of materials to the server

### Self-test

1. A household messenger is introduced.   
  
2. It is a one-to-many setup. Professor and students are allowed to use the platform to communicate by raising and solving questions.   
  
3. The questions involved will be automatically saved to the database and displayed at the message panel. The mechanism mainly consists of the data server using Visual Studio Integrated Development Environment as a tool to make use of WCF (Windows Communication Foundation) to enable a web service. The data server uses a dataset to interact with SQL database.   
  
4. Some methods or functions have been used to implement these features. For example, Insert new question, Get question list, Insert answer to a question. All of these manipulations requires parameters and makes use of Ajax requests provided by JavaScript for its purpose. Currently, the text feature for communication has been implemented in the messenger and the audio recording part is in progress. Audio recording allows a native and friendly way for course stakeholders to share views and information.

### Q & A module

The self-test module contains 5 interfaces for different purposes. The purposes of each interface are as follows:   
1. For instructors to add test questions   
2. For instructors to view and mark students’ answers   
3. For students to answer questions   
4. For students to evaluate their answers in comparison to model answers   
5. For students to check their test scores  
   
Below are the main details about the implementation of each of the interfaces:  
  
1. Interface for instructors to add test questions  
- A left toggle menu displaying the questions (with options to edit and delete)  
- Add Test Questions (in textbox)  
- Edit Test Questions  
- Delete Test Questions  
- Select answer type  
- Enter corresponding answers (depending on type of answer)  
- 4 answer types: 1. Multiple Choice 2. Text 3. Drawing 4. Text & Drawing  
  
2. Interface for instructors to view and mark students’ answers  
- A left toggle menu to click and open a student’s answer  
- An area to display the student’s answer  
- A slide bar to assign marks  
  
3. Interface for students to answer questions  
- Questions will appear one by one from the 1st to the last question in Question Area  
- Answer Area (Display varies based on answer type):  
 Canvas for drawing, Textbox for words, Radio Buttons for Multiple Choice  
- Students can answer each question and go to the next one  
- A progress bar showing percentage of test done  
  
4. For students to evaluate their answers in comparison to model answers  
- Two parts of interface: left part showing student’s answer and right part showing model Answer  
- Answers to each question will appear one by one from the 1st to the last question  
- A slide bar to assign marks  
  
5. For students to check their test scores  
- A list view with each question and the corresponding mark  
  
At the moment, the first 3 interfaces mentioned above have been partly implemented.

# 

# 

# Implementation

## Tasks & tools

## Libraries used

## 

# Testing

Test cases for objectives

# Evaluation

## Server-client communication

User comments

## 

# Discussion

Future of this project

In the future, the system may have automatic marking for some questions based on keywords entered by the instructor.

# Conclusions

# References

1. "Structure and Interpretation of Computer Programs - MIT Press." 2011. 12 Apr. 2016 <<https://mitpress.mit.edu/sicp/full-text/book/book.html>> [↑](#footnote-ref-1)
2. Cohen, Henri, and Claire Lefebvre. *Handbook of categorization in cognitive science*. Henri Cohen & Claire Lefebvre. Elsevier, 2005. [↑](#footnote-ref-2)
3. Shneiderman, Ben. "Tree visualization with tree-maps: 2-d space-filling approach." *ACM Transactions on graphics (TOG)* 11.1 (1992): 92-99. [↑](#footnote-ref-3)
4. "Introduction to database design - Datanamic." 2008. 12 Apr. 2016 <<http://www.datanamic.com/support/lt-dez005-introduction-db-modeling.html>> [↑](#footnote-ref-4)
5. "Home : The Official Microsoft IIS Site." 2002. 12 Apr. 2016 <<http://www.iis.net/>> [↑](#footnote-ref-5)