University of the West of England
BRISTOL

MODULAR PROGRAMME
COURSEWORK ASSESSMENT SPECIFICATION

Module Details

Module Code UFCFX3-15-3	**Run** **Main**	**Module Title** Advanced Topics in Web Development 1
Module Leader		**Module Tutors** **SM LAU**
Component and Element Number B (CW)		**Weighting:** 50%
Element Description Coursework		**Total Assignment time**

Dates

Date Issued to Students	**Date to be Returned to Students**
Submission Place Soft copy : Moodle	**Draft Submission Date and Time:** 26-Nov-23, Sunday, 23:55
	Submission Date and Time: 17-Dec-23, Sunday, 23:55

Deliverables

Working application and documentation as specified.

Module Leader Signature

Assessed Coursework

Learning objectives

- Develop RESTful Web Services using PHP, XML and XSL
- Develop a client-side application using HTML, CSS, and JavaScript to consume the GET, POST, PUT, and DELETE of the above RESTful Web Services
- Apply a range of web-oriented software architecture and design principles
- Apply the principles of structured, functional & Object-oriented programming
- Apply good practices and testing in Web programming

Overview

You are required to build:

(1) RESTful web services for search and maintenance of "Information of Automated Teller Machines of retail banks" provided by the Hong Kong Monetary Authority.

(2) A web-based application as a client to the RESTful web services in (1). The client offers enquiries, visualization and maintenance of the information of ATM.

Additional requirements that must be fulfilled are:

- The data returned by the web services must be in JSON format. No client-side formatting instructions (such as HTML tags) are allowed to be included in the JSON returns. The RESTful web services must be implemented with PHP. For storing ATM information, you may use local MySQL database or a cloud-based database platform at your own choice.
- The client must employ significant JavaScript for AJAX and DOM manipulations so as to improve user experience. Refreshing and reloading of client pages must be severely limited. In other words, a Single Page Application (SPA) is expected.

This coursework consists of **FOUR** parts, which are specified in subsequent sections.

Part 1: Data Conversion (10%)

To set up your database of ATM information, you need to download the dataset from DATA.GOV.HK with the link:

https://data.gov.hk/en-data/dataset/hk-hkma-bankbranch-banks-atm-locator

The dataset is in JSON format. Meaning of the attributes are self-explanatory with the attribute names.

You are required to develop a server-side command tool to get the dataset from the link above and store the data into a database. Your tool must download, convert and store the data records on the fly without human intervention. In other words, you are **not** allowed to download and save the dataset as a local file, then convert and insert the records into the database by issuing separate commands. Instead of issuing commands, you may include the functionality in this part in your client application (see Part 3).

NOTE:

- The dataset is available in three languages: English, Traditional Chinese and Simplified Chinese. While it is mandatory for your web application to support English, you may opt to support Chinese too (see Section 3.4).
- Design the database schema carefully and make sure that it matches the dataset and supports the server APIs. You are advised to include a unique ID for every record.
- Shown below are the first few records and the last record in the dataset in English.


```
{
   "header":{
      "success":true,
      "err_code":"0000",
      "err_msg":"No error found"
   },
   "result":{
      "datasize":100,
      "records":[
         {
            "district":null,
            "bank_name":"Standard Chartered Bank (Hong Kong) Limited",
            "type_of_machine":"Automatic Teller Machine",
            "address":"Shops 106 - 109, 1/F, Mei Foo Plaza, Mei Foo Sun
Chuen Stage 4, Kowloon",
            "service_hours":"24 Hours",
            "latitude":"22.335983",
            "longitude":"114.14140199999997"
         },
         {
            "district":null,
            "bank_name":"Standard Chartered Bank (Hong Kong) Limited",
            "type_of_machine":"Automatic Teller Machine",
            "address":"Shop G6, G/F & Shop LG1, LG/F Westlands Gardens,
1027 King's Road, Quarry Bay, Hong Kong",
            "service_hours":"24 Hours",
            "latitude":"22.28516",
            "longitude":"114.21308099999999"
         },

……

         {
            "district":"Yuen Long District",
            "bank_name":"Bank of China (Hong Kong) Limited",
            "type_of_machine":"Automatic Teller Machine",
            "address":"8-18 Castle Peak Road, Yuen Long, New Territories",
            "service_hours":"Mon - Fri: 9:00 – 17:00<br>Sat: 9:00 –
13:00<br>Sun and public holiday: Closed",
            "latitude":"22.4443379",
            "longitude":"114.0317002"
         }
      ]
   }
}
```

Part 2: Design & Implementation of RESTful Web Services (Total 30%)

You are required to develop RESTful web services that allow retrieval and maintenance of ATM records in your database constructed in Part 1. The data sent from your server to its clients must be in JSON format.

NOTE:

- PHP must be used in implementing the web services and no high-level framework such as CodeIgniter and Laravel can be used.

The web services must employ the HTTP basic methods as below:

HTTP methods	Usage	Marks
GET	**Retrieve** details of ATM records such as bank name, address, and service hours.	10%
PUT	**Update** information of ATM records.	5%
POST	**Add** new ATM records.	5%
DELETE	**Remove** existing ATM records.	5%

2.1 Further note on service invocations

You must define the APIs (format of the URLs) precisely for invoking the web services.

- For retrieval using GET, a possible format may be:

  `http://www.myserver.com/atm?district=Tsuen+Wan`

  The above URL requests the server to return the list of ATMs in the Tsuen Wan district.

- It is your own discretion to decide whether parameter values are attached to the URL and sent through the HTTP body.

NOTE:

- The API above is just an example. You may define your own API formats according to the operations and parameters needed.
- The APIs must be precisely defined and documented in the report (see Part 4) or else developer of client applications will not be able to invoke the services.
- To simplify your work, only English needs to be supported by your web services. However, your application may support Chinese as an advanced feature (see Section 3.4).

2.2 Further note of data format

Any data returned from the server to clients must be in JSON format. Shown below is a possible return from the server when the client has requested a list of ATMs of the Bank of China in the Yuen Long district.

```
{
   "header":{
      "success":true,
      "err_code":"0000",
      "err_msg":"No error found"
   },
   "result":[
      {
         "district":"Yuen Long District",
         "bank_name":"Bank of China (Hong Kong) Limited",
         "type_of_machine":"Automatic Teller Machine",
         "address":"8-18 Castle Peak Road, Yuen Long, New Territories",
         "service_hours":"Mon - Fri: 9:00 – 17:00<br>Sat: 9:00 – 13:00, Sun
and public holiday: Closed",
         "latitude":"22.4443379",
         "longitude":"114.0317002"
      },
      {
         "district":"YuenLong",
         "bank_name":"OCBC Bank (Hong Kong) Limited",
         "type_of_machine":"Automatic Teller Machine",
         "address":"Shop 1-3, G/F, 40-54 Castle Peak Road, Yuen Long",
         "service_hours":"7x24",
         "latitude":"22.444222",
         "longitude":"114.030768"
      },
      {
         "district":"YuenLong",
         "bank_name":"Industrial and Commercial Bank of China Limited",
         "type_of_machine":"Automatic Teller Machine",
         "address":"Shop No. 311, Level 3, Ticketing Concourse, KCR Lok Ma
Chau Station, N.T (Paid Concourse)",
         "service_hours":"Follows the service hours of the mall or MTR
station where it locates",
         "latitude":"22.514434",
         "longitude":"114.065658"
      },
      ……
   ]
}
```

NOTE:

- The above JSON return is just an example. You may define your own JSON format.

2.3 Error Handling (5%)

The web services are required to handle errors in a graceful manner. When an API is invoked incorrectly (for example, missing an essential parameter or having incorrect parameter format), the server must return relevant and meaningful error codes and messages to the client. Again, the error code and message are formatted as JSON strings. An example is shown below where a parameter is invalid for a retrieval service.

```
http://www.myserver.com/atm/district=
```

```
{
   "header":{
      "success":false,
      "err_code":"0120",
      "err_msg":"missing district value "
   }
}
```

You need to define your own error codes and messages according to your own web services. The error codes and messages must be precisely defined and documented in the report (see Part 4).

NOTE:

- The above JSON return is just an example. You may define your own JSON format.

Part 3: Client-side Application (Total 30%)

3.1 Web application to enable retrieval of information on ATM records (10%)

Design and implement a Single Page Application (SPA) that uses the RESTful services implemented in Part 2. Your application should provide an interface for users to enter the retrieval criteria like service hours and location district.

As a SPA, refreshing and reloading of the entire page is prohibited. Instead, JavaScript, AJAX and DOM manipulations should be used to make requests, retrieve server replies, and updates web contents for displaying retrieval results.

NOTE:

- The client-side application must be written in HTML and JavaScript.
- Use of UI libraries such as Semantic UI and Bootstrap are allowed but only limited for providing UI effects and input form validations.
- The use of jQuery is only limited for UI effects. The use of jQuery for making AJAX requests to server APIs is strictly prohibited.
- The use of high-level frameworks for application generation or completing a server request cycle is strictly prohibited.

3.2 Web application to maintain information on ATM records (10%)

Your application should provide web pages to allow maintenance of ATM records including addition of a new ATM, removal of an existing ATM, and update of ATM details. The maintenance functions must employ the RESTful services implemented in Part 2.

NOTE:

- The pages for maintenance of ATM records can be separated from the retrieval page in (3.1) as they are essentially different functionalities. However, excessive refreshing and reload should still be avoided for ensuring better user experience.

3.3 Usability and User Experience (5%)

You have to design the operation flow very carefully. For example, a typical operation is to search for ATMs in a specific district, then select one of the listed ATMs for viewing the details. A very bad design is to request the user to remember the ID or other fields and go to another page for viewing the details by entering the ID or other fields again.

3.4 Challenging features (5%)

You are encouraged to design and implement challenging features besides those stated above, such as locations of ATMs on a Google Map, path going to a selected ATM, support of Chinese, etc.

NOTE:

- When you design the user interface in the client application, make sure that it is user-friendly and is consistent in its appearance in terms of styles, font sizes, and the use of colours. The operations must also be intuitive to novice users.

Part 4: Presentation, Demonstration and Documentation (30%)

Presentation (5%):

- You are required to deliver a 5-minute PPT presentation on what you have done, the design of the server API, tools and techniques employed, critical evaluation, further enhancement, etc.

Demonstration (5%):

- You are required to deliver a 10-minute demonstration on your application, including the server-side data conversion program, the server APIs and the client-side application.
- For the server APIs, you need to use **Postman or similar tools** to make requests to the server and to show the JSON returns from the server, including both normal data and error codes/messages.
- Note that it will be the students to lead the entire demonstration to show the functions and features rather than the marker to guide it.

Report (20%)

- You are required to submit a written report to document the design and technical details of your application.

- While you are the one to decide the structure and contents of the report, the following items MUST BE included.
    - Design & Implementation, including specification of APIs to invoke your web services and the error codes and messages possibly returned by the server.
    - Explain clearly how different software technologies (e.g. XML, XSLT, CSS, JavaScript, Object-Oriented programming and design patterns, etc.) were applied in your system.
    - Evaluation of your application showing the advantages over the traditional non-RESTful systems.
    - Future improvement
    - Conclusion

Draft Submission

In order to ensure that you have proper progress and to your work is on the right track, you are required to submit a draft of your report and a video record of your prototype demonstration by 26-Nov-23, 23:55 to Moodle.

Note that the draft submission is mandatory. A maximum penalty of 20 marks will be imposed on the final score if draft submission is omitted or if the draft is of poor quality without reasonable amount of effort and quality.

Submission Guidelines

You have to submit THREE items listed below.

(1) Presentation PPT

(2) Source Code:

- Submit all source files including the server codes, SQL for creating and manipulating the database, and client-side codes (HTML, js, images, etc.). If you have employed any external libraries, they must also be submitted.
- Organize the files into appropriate folders with self-explaining folder names.
- Compress all source files and folders into one single ZIP file (zip only, rar and 7zip NOT allowed).
- Submit the ZIP file to Moodle before the deadline.
- The submitted source codes **MUST BE THE SAME** as those used in the demonstration.

(3) Report:

- Submit the report to Moodle before the deadline.
- Only MS-Word or PDF formats are allowed.

IMPORTANT: It is the responsibility of students to make sure that the ZIP and other files submitted can be extracted and opened successfully. Students will be awarded **ZERO** marks for the relevant component if the submitted file(s) cannot be opened successfully.

REMINDER

- **The RESTful server you developed in this ATWD1 coursework will be used to support the client application in ATWD2 coursework (next semester).**
- **If you produce poor work in this ATWD1 coursework, very likely you will have to redevelop the server in order to complete the ATWD2 coursework in the next semester and this will increase your workload in semester 2. (Marks will not be awarded for your server redevelopment if needed.)**
- **So make sure that you put adequate effort in developing the RESTful server and to provide enough APIs to support the client application.**

Assessment Criteria

Part 1: Data Conversion (10%)			
0-2	• Failed to download data source and migrate to MySQL database or any cloud-based database platforms.	Comments:	Marks: ____/10
3-5	• Able to download data source and migrate to MySQL database or any cloud-based database platforms. • Some deficiencies in the process, for example, not on the fly but with intervention with separate command steps. • Database schema may not be able to incorporate all fields in the data source.		
6-8	• Successful migration of data to MySQL database or any cloud-based database platforms on the fly using one single command. • Good database design that supports all fields in the data source.		
9-10	• Excellent implementation quality, possibly with some advanced features, for example, report on data items imported or any irregularities.		

Part 2: Design & implementation of RESTful Web Services (Total 30%)				
Retrieval (10%)	0-2	• Partially functional with errors when retrieving information on ATM records.	Comments:	Marks: _____ /10
	3-5	• Functional but only very limited or no searching criterion is supported.		
	6-8	• Support different search criteria • Reasonable API format • Support some error checking • Reply to client application in JSON format		
	9-10	• Sophisticated search criteria • Good API format		
Update (5%)	0-2	• Partially functional with errors in updating ATM records. • NOT PUT – Award max 2 marks if not using the PUT method, regardless of other assessment criteria in this component	Comments:	Marks: _____ /5
	3-5	• Using PUT method • Reasonable API format • Support some error checking		
Add	0-2	• Partially functional with errors in adding new	Comments:	Marks:

(5%)		ATM records. • NOT POST – Award max 2 marks if not using the POST method, regardless of other assessment criteria in this component.		_____ /5
	3-5	• Using POST method • Reasonable API format • Support some error checking		
Remove (5%)	0-2	• Partially functional with errors in deleting ATM records. • NOT DELETE – Award max 2 marks if not using the DELETE method, regardless of other assessment criteria in this component.	Comments:	Marks: _____ /5
	3-5	• Using DELETE method • Reasonable API format • Support some error checking		
Error Handling (5%)	0-2	• No or only very limited error codes and messages	Comments:	Marks: _____ /5
	3-5	• Adequate to sophisticated error codes and messages • Well design of error codes (e.g. hierarchical numbering according to error types) • Clear and meaningful error messages that describe the nature and cause of errors clearly		

Part 3: Client-side application (Total 30%)				
Retrieval of information on ATM records (10%)	0-2	• Partially functional with errors when retrieving ATM records.	Comments:	Marks: _____ /10
	3-5	• Functional and have reasonable UI for specifying searching criteria • NOT SPA – Award max 5 marks if not being SPA, regardless of other assessment criteria in this component.		
	6-8	• SPA • Good UI that allows users to specify searching criteria easily and precisely • Provides reasonable feedbacks to users where errors are returned from server.		
	9-10	• SPA • Sophisticated search criteria		

		• Clear and intuitive formatting of ATM details possibly with advanced features like using JQuery for enhancing visual effects.		
Maintenance of ATM records (10%)	0-3	• Partially functional with errors	Comments:	Marks: _____ /10
	4-7	• Functional and have reasonable UI • Some error checking and feedbacks to users when errors occur		
	8-10	• Provide adequate responses to users after successful completion of maintenance operations • Sophisticated error checking and feedbacks to users when error occur		
Usability & User Experience (5%)	0-2	• Poor workflow design • Not intuitive to users for performing the operations.	Comments:	Marks: _____ /5
	3-5	• Good UI and workflow design • Straight forward and intuitive for users to perform required operations.		
Extra and challenging features (5%)	• Award marks according to usefulness and implementation difficulty • Each feature should normally be awarded with 3 marks and max 5 marks		Comments:	Marks: _____ /5

Part 4(a): Presentation and Demonstration (Total 10%)				
Presentation (5%)	0-2	• Insufficient contents • Lack of preparation	Comments:	Marks: _____ /5
	3-5	• Adequate contents with essential items covered • Well prepared and smooth presentation		
Demonstration (5%)	0-2	• Poor organization and difficult to follow • Demo steps not planned in advance • Functions are not illustrated clearly	Comments:	Marks: _____ /5
	3-5	• Good organization with carefully planned steps to		

		illustrate the functions		
Part 4(b): Documentation and Critical Evaluation (Total 20%)				
Design and specification of web service APIs (5%)	0-2	• Insufficient details provided for API formats. • Poor API design that may not support necessary parameters	Comments:	Marks: _____ /5
	3-5	• Good API design with clear and precise specification on formats • Sufficient details provided for API users		
Specification of error codes and messages (5%)	• Award marks according to comprehensiveness of error types and descriptiveness of error messages.		Comments:	Marks: _____ /5
Explanation of technologies employed (3%)	• Students are expected to provide some discussions on the technologies employed in the implementation and WHY those technologies are good to use in the system		Comments:	Marks: _____ /3
Evaluation (3%)	• Award marks according to whether the student has critical thinking about the design, functionality and use of technologies		Comments:	Marks: _____ /3
Discussion of future improvement (2%)	• Students are expected to provide discussion on enhancement of functions, and/or improvement of design		Comments:	Marks: _____ /2
Formatting and overall report quality (2%)	• The reports are expected to be formatted with a high standard and look professional. • Consistency in formatting like font faces, font sizes, paragraph spacing are expected. • All sections are expected to be numbered and reflected in the Table of Contents. • References are to be provided.		Comments:	Marks: _____ /2

END.