Assignment for Advanced Software Development CMS3405

Date Written: September 2017
Handed to Students: 15th February, 2018
Hand In Date: 17 April, 2018

1. Introduction

This assignment is an individual work and has no group work component. It contributes 50% to the overall assessment of the module.

2. Aims

1. To use advanced methods and programming to develop a piece of software
2. To design GUI components
3. To use event-handling, and thread management facilities provided by development tools and environments, such as Eclipse, or Microsoft Visual Studio
4. To develop an interactive system for object manipulation
5. To apply appropriate Design Patterns in the design and creation of that software
6. To explore Distributed Object Technologies as mechanisms for the further development of the system architecture.

3. Outcomes to be assessed

This coursework assesses the following learning outcomes (listed in the module specification):

1.1 Critically analyse a range of techniques for advanced OO programming, XML, and data communication.
1.2 Evaluate methods for the development of real-time, concurrent, and distributed software systems.
2.1 Design and implement complex data structures, using appropriate methods of encapsulation, inheritance, polymorphism, and genericity.
2.2 Model advanced real-time, concurrent, and distributed software systems as sets of objects, and defend the appropriateness and consistency of the model.
2.3 Implement such models using event-based and object-oriented techniques as appropriate.

4. Tasks

You are required to develop a simple application in Client – Server based orientation service for tourist business using portable devices, such as laptops, mobile phones or tablets.

- A general data sharing function is required for the application.
- The main function is to submit data (text, image, documents, video or audio clips) to the server and access the data via the Wi-Fi or 3G mobile wireless network.

- Three points, person 1, person 2 and the server, should be connected into your system, i.e. person 1 uses a portable device to send the request to person 2 about renting, for example, he/she wants to rent a specific car. The request can be a text message.
- Person 2 will send his response back to the system to indicate the availability and the renting process, e.g. daily rate, weekly rate, or monthly payment, and the information for the car, e.g. the make, the number of doors, colour, and the terms and conditions for the customers.
- A response can be a text message, image file or video clip(s).
- Person 1 can view the information from person 2, and then send the response back to person 2 with text, audio or image.
- All the communications are required to be stored in the server.
- You should use an appropriate Design Pattern for the design and creation of your software.
- A written report of 2,000 to 2,500 words (see section 6).

You are not constrained in the choice of the languages, and platform or tools to develop your application as long as you indicate the methods and tools are on advanced approach for your application. For example, you can use LINUX platform, Microsoft Visual Studio, Java, Android, C#, JQuery, AJAX or others. Some code examples are shown in the reference links.

5. Hints and Tips for Approaching the Problem

1. Both people need a login facility, i.e.
   a. Basic login: username, password
   b. Advanced login: create account – username, password, email address, affiliation, etc.
2. Person 1 needs:
   a. A message sending facility to send the request to the server;
   b. A viewing facility to retrieve person2's replies from the server;
   c. Basic retrieve: text message;
   d. Intermediate retrieve: static image;
   e. Advanced retrieve: audio or video clip (s);
3. Person 2 needs a reply facility back to person 1:
   a. Basic sending facility: a text message;
   b. Intermediate sending facility: static image;
   c. Advanced sending facility: audio or video.
4. Server facility needs to
   a. Connect both person 1 and person 2;
   b. Stores person 1's requests/inquiries and person 2's replies.

You can download some sample code from Unilearn but they are not your answers. You can get some resources from the references listed at the end of the document.

6. Evidence to be Submitted and Weightings

You must submit a written (typed) report to contain the following:

- An introduction that references the reviewed literature as a background study for your prototype (10%)
- A discussion of the design of your prototype software system. This should include a discussion of the role Design Patterns have played in the design of the software (i.e. which patterns have been considered, which have been adopted, and an evaluation of how well they have worked). It must also include a Class Diagram for the final system. (15%)
- A discussion of the client-server aspects of the system. This could take the form of a purely theoretical discussion of how client-server architecture could be achieved (e.g. what middle-ware or object broker software would be used, with an updated software design to reflect this). Alternatively you can actually implement (some or all of) this client-server architecture. (25%)
- Statements of the functionality you have implemented and the testing you have carried out. This should include Unit testing (where appropriate) and full system testing. (25%)
- Your programming source code with proper comments will be included in the report as an appendix. (25%) Further guidance will be given in class regarding the scope.

7. Grade Indicators

The list shown below is intended to give a rough idea of the grade to be expected for the differing standards of work which might be submitted.

D - All evidence components attempted.

Some attempt to identify and apply appropriate design patterns

Some discussion of appropriate client-server architecture

Interfaces have been attempted, but there are features missing or errors in their behaviour. At least one response can be used.

Some testing attempted.

Both sending and receiving text message successfully with evidence

C – All evidence components attempted.

A reasonable attempt to apply appropriate design patterns leading to a generally suitable design, but with some errors or omissions

A reasonable discussion of appropriate client-server architecture with some attempt at implementation

Most interface features implemented, with multiple responses.

System may not be robust, or visual interfaces may be muddled and hard to follow. A reasonable attempt at testing, but the test cases may be incomplete or the test strategy may be unclear

Sending and viewing text message and file(s) are successful with discussions and working code as well.

B – A generally good report, containing a good design based upon appropriate consideration of concepts of user oriented approach, and a well thought out discussion of appropriate client server architecture. Some good attempt has been made to apply the client-server extensions to the software.

The software is as for a C, but with some attempt at trapping and handling errors (i.e. a more robust system) and consistent and well-arranged visual interfaces.

Clear separation of person 1 and person 2 interfaces and synchronisation issues recognized.

Sending and retrieving text, file(s), image and video successfully with evidence.

A – As B, but with

- Excellent code clarity, everything working, and superb visual interfaces
- A high quality report throughout covering all required contents to a high standard
- Excellent testing of the software system with evidence
- Data linked to the database system but not necessary follow specific types, such as relational database, object relational database, or object database, XML database system.

Please indicate the total words counted using the word count facility in the word process software. Please follow the guidance in the module specification and module hand book for the assignment report writing. The suggested word limit will be in a range between 2,000 to 2500 words.

8. References:

[1] Fami Rahman, http://fahmirahman.wordpress.com/2011/04/21/connection-between-php-server-and-android-client-using-http-and-json/

[2] Ratan, http://webdesignergeeks.com/mobile/android/android-login-authentication-with-remote-db/

[3] Saranga Rathnayake http://sarangasl.blogspot.co.uk/2011/06/android-login-screen-using-httpclient.html

[4] Lars Vogel, Android ListView and ListActivity – Tutorial, http://www.vogella.com/articles/AndroidListView/article.html

THE UNIVERSITY OF HUDDERSFIELD

School of Computing and Engineering

GRADING FORM

*Module Code/Title.....*CMS3405 – Advanced Software Development

Student Name ....

Course..........MSc.............. Year.......One..............

	GRADE	
LEARNING OUTCOME / ASSESSMENT CRITERION		***COMMENTS ON EACH OUTC CRITERION***
An introduction that references reviewe literature as a background study for yo prototype (10%)		
Discussion of the design of your protot software system. This should include a discussion of the role Design Patterns h played in the design of the software (i.e patterns have been considered, which h been adopted, and an evaluation of how they have worked). It must also include Diagram for the final system. (15%)		
Discussion of the client-server aspects c system. This could take the form of a p theoretical discussion of how client-ser architecture could be achieved (e.g. wh middle-ware or object broker software be used, with an updated software desig reflect this). Alternatively you can actu implement (some or all of) this client-s architecture. (25%)		
Statements of the functionality you have implemented and the testing you have carried out. This should include Unit testing (where appropriate) and full system testing. (25%)		
Your programming source code with proper comments will be included in the report as an appendix. (25%)		

Overall Grade and Comment:

○

GENERAL GRADING CRITERIA

- OUTSTANDING (A)

  Meets a;; reaspman;e reqior,emts. wotj ;ott;e o,[rpve,emt [psson;e om tje to,e a;;pwed
  Shows excellent understanding
  A piece of work worthy as an exemplar
  Excellent approach and use of techniques

- ABOVE AVERAGE (B)

  Exceeds minimum requirements, but some improvement possible
  Shows very good understanding
  Majority of elements covered correctly
  Demonstrates good approach and use of techniques

- AVERAGE (C)

  Meets minimum requirement but needs to be more concise and related to the topic
  Shows reasonable understanding
  Minor elements omitted or incorrect
  Approach and use of techniques could be improved

- SATISFACTORY (D)