FIT5170 Programming for Distributed, Parallel and Mobile Systems

Assignment One
Semester 1, 2014

Introduction

This assignment is due by 11:55 PM, Tuesday 15th April, 2014. It is worth 25% of the marks for your final assessment in this unit. **A penalty of 10% per day, including each day of a weekend, will apply for late submission.** Refer to the FIT5170 Unit Guide for the policy on extensions.

This is an individual assignment and must be your own work. Please note the section on plagiarism, cheating and collusion in this document.

Specification

Assignment 1 of this semester involves the design and implementation of a Hotel Booking Broker. A Hotel Booking Broker is an online system that allows users to search for, compare rates, check availability and make bookings at the hotels connected to the service in various cities throughout Australia. Note that, for the purpose of this assignment, the various hotels with the same name, but in different cities, are unrelated. So, for example, the Hilton Hotel in Melbourne is considered to be totally unrelated to the Hilton Hotel in Perth – and so on.

The system will require a complete separation of presentation from logical aspects, so you should develop a traditional client/server system that can later be modified to use other middleware technologies. The system should be structured so that changes to clients, servers or middleware technology can be done as easily as possible.

Details

You will need to implement a client and one or more servers using a socket-based protocol.

The client will need to be able to make the following requests to the central Hotel Booking Broker server:

- What cities are currently serviced by the system? The response should be a list such as: Melbourne, Sydney, Perth, etc.
- For a given city, what hotels are currently serviced by the system? The response should be a list such as: Hilton, Chevron, Regent, Windsor, etc.
- For a particular hotel, what is the room rate? (For simplicity, you can assume a hotel has one room rate, but the different hotels may have different rates).
- For a particular hotel, does it have a vacancy between given check-in and check-out dates? (For simplicity, you can assume all vacancies and bookings should be for July, 2014).
- Book a room at a hotel. Completing the booking request should include the user submitting

the following information: check-in date, check-out date, guest's name, contact (phone or e-mail address), and credit card number. You should decide on appropriate valid values.

Assignment levels

This assignment has graded levels. The simplest version will be given a *maximum* of a Credit grade if it is implemented satisfactorily, provides the required functionality, is robust, exhibits the application of sound design principles and coding conventions, all deliverables are submitted according to requirements and the student performs satisfactorily at the compulsory interview. To gain higher grades more features will have to be added and, likewise, assessment of functionality, design and coding, deliverables and student understanding will be applied to those added features.

General comments:

- Your program must be able to be launched from a command line and therefore must be able to run independently of any IDE you choose to develop in.
- *Your system must compile and run in the university's student labs in which the tutorials for FIT5170 are held this semester.*
- Your clients can use a text-based interface, reading from `System.in` and writing to `System.out`, or can use a GUI with AWT or Swing objects – if you know how to implement such an interface. It does not matter which you choose -the mark will not depend on the user-interface style. *But please note that you can be expected to explain the code and working of any GUI you submit to demonstrate that it is your own work.*
- The quality of design and code will be assessed. Poor quality will lose marks. The principles of good design should be employed. Code should comply with Java coding conventions, be appropriately indented, self-documenting in terms of variable, method and class names and comments in the body of methods or blocks should be kept to a minimum and used only to explain complex sections of code.
- Consistency between specifications, diagrams and code will be checked.
- Invalid user input, errors and exceptions, service faults and unavailable services should be handled appropriately.

Credit level

Clients talking to a single server. Your system should be capable of handling at least two clients concurrently. The server handles all of the Hotel Booking Broker requests. The server maintains information on the cities, hotels, their rates and bookings for each hotel. All of this can be done using data structures, Java Collections and local text or data files kept in memory. For the purpose of satisfying these criteria you should have hotels "located" in at least two cities, with at least two hotels in one city and one in the other.

All clients and servers will be written in Java. However, you should not assume that the other end of a communication channel will be in Java -it could be in some other language such as C# or Perl. So do not use any Java-specific techniques such as serialising Java objects.

This level demonstrates that you can specify a protocol for communication between a client and a single server and implement this protocol in a simple way.

Extra levels

The specification given above will gain *up to a maximum* of a Credit grade if done satisfactorily. Higher grades may be achievable on completion of additional functionality. However, implementing or attempting the functional requirements for a Distinction or High Distinction alone does not guarantee that grade – refer to the earlier part of this document. The functional criteria for the higher grades are as follows:

Distinction level

You must have successfully implemented the Credit level specification before you attempt the Distinction level.

Implementing multiple servers so that each hotel runs its own server will receive *up to* a *maximum* of a Distinction grade. Your system should be capable of handling at least three hotel servers, two in one city and one in another, concurrently. The Hotel Booking Broker will forward certain requests (such as rates, vacancies etc.) directly to the hotel's server rather than maintaining information on its own server.

This demonstrates that you can deal with a multi-tier system with many servers at the backend.

High Distinction level

The criteria for a High Distinction must not be attempted, and implementations will not be marked, unless all of the criteria for the Distinction level are satisfied.

Achieving *up to* a High Distinction grade for this assignment requires you to demonstrate that you can manage a typical 3-tier system with presentation, logic and database handled separately. All databases could be managed by a MySQL database system / Java DB (Derby) database in NetBeans IDE and the server will communicate with the database by JDBC calls.

NOTE: The requirements stated herein for the High Distinction level require an amount of independent research on your part that you must investigate and implement – it is NOT taught as part of this unit.

Submission Requirements

There is NO hard copy submission for this assignment. You are required to submit your assignment as a .zip file named with your student number. For example, if your student number were **12345678** then you would submit a file named **12345678_A1.zip** - marks will be deducted if this requirement is not strictly complied with.

Your .zip file is to be submitted via the Assignments link in the FIT5170 Moodle site by the deadline specified in this document.

Deliverables

Your submitted .zip file should contain the following:

- A completed Assignment Cover Sheet. This document is available for download from the Assignments section of the FIT5170 Moodle site.
- An MS Word document, stating:

  The level of assessment you have attempted, and the parts you have completed.

  Clear and complete instructions on how to install and run each client and server, including any database and database files. (Note*: Your system must compile and run in the university's student labs in which the tutorials for FIT5170 are held this semester. Any submission that does not will receive 0 marks.*)

  Specification of all messages between client and server(s). This specification should include enough details about message formats that someone else could implement a client or server (in any language, not just Java) to handle the messages.
- A document, readable in the university's student labs, containing UML diagrams showing the classes in your system and their linkages.
- Electronic copies of ALL your files needed to run your system.

Marks will be deducted for any of these requirements that are not complied with.

Interview

You will be asked to demonstrate your system at an interview in the week following the submission date. At the interview you can also expect to be asked to explain your system, your code, your design, discuss design decisions and alternatives and modify your code/system as required. Marks will not be awarded for any section of code or functionality that a student cannot explain satisfactorily. (The marker may delete excessive comments in code before a student is asked to explain that code).

Interview times will be arranged in the tutorial labs immediately preceding the submission deadline. It is your responsibility to attend the lab and obtain an interview time. Interviews will not be arranged by email or by students on behalf of other absent students. *Students who do not attend an interview will receive 0 marks for the assignment.*

Wiki

To help you learn and progress with this assignment, each student has their own Assignment 1 Wiki, accessible from the Assignments section of the unit's Moodle site. Each student's Wiki is visible to, and editable by, the student and the lecturer only.

The Wiki is to be used as a log or diary in which you will record your progress with the assignment – what you have attempted, completed, problems you encountered, how you solved those problems and, most importantly, reflections on what you have learned in attempting and completing the various stages of the assignment. As such, you should also record suitable questions and tasks that could be used to demonstrate and verify such learning in the compulsory interview following the assignment's submission.

Please note that the questions and tasks nominated by any student may or may not be used at interview. Their usage will depend on their suitability, as determined by the lecturer, and questions and tasks used at the interview may not be restricted to those recorded in any student's Wiki.

The Wiki component is worth 5 of the 25 marks this assignment contributes to your final assessment in the unit. The marks available for an individual student's Wiki will be dependent on the graded level the student has attempted and completed. For example, a student who attempts and completes a Credit level of the assignment implementation can only achieve *up to* a Credit level for the Wiki component, subject to the quality of the Wiki.

The Assignment 1 Wiki is not a question and answer forum for students to get help on their assignment. You must therefore not post questions on your Wiki expecting a response. You are to restrict your Wiki entries to your progress with, and reflection on, Assignment 1 only.

While you are encouraged to be creative and express your personality with your Wiki writing, you must not write anything that can be perceived as abusive or offensive to anyone in any way.

Finally, it is hoped that the Assignment 1 Wiki will enhance your learning experience in developing your assignment by encouraging you to reflect on the practical programming skills and theoretical knowledge you gain as you develop your solution. It should also allay your anxiety about the compulsory interview following submission by empowering you with the potential opportunity to determine your own interview agenda.

Plagiarism, cheating and collusion

Students should consult University materials on this matter at:

http://www.infotech.monash.edu.au/units/appendix.html#plagiarism

The following excerpt is from the aforementioned URL:

Plagiarism and cheating are regarded as very serious offences. In cases where cheating has been confirmed, students have been severely penalized, from losing all marks for an assignment, to facing disciplinary action at the Faculty level. While we would wish that all our students adhere to sound ethical conduct and honesty, I will ask you to acquaint yourself with the University's Plagiarism policy and procedure (http://www.policy.monash.edu/policy-bank/academic/education/conduct/plagiarism-procedures.html) which applies to students detected plagiarizing.
It is your responsibility to make yourself familiar with the contents of these documents.