Assignment Brief

Submit assignment

Marks and feedback

Submission Deadline	Marks and Feedback
Before 10am on: **17/05/2019**	**20 working days after deadline (L4, 5 and 7)** **15 working days after deadline (L6)** **10 working days after deadline (block delivery)** 31/05/2019

Unit title & code	CIS016-1 Principles of Programming
Assignment number and title	Assignment 2 Group Project
Assessment type	Artefact and presentation
Weighting of assessment	75%
Unit learning outcomes	Demonstrate to following knowledge and understanding • Assess techniques relevant to the challenges faced in today's complex environments, concerning matters including, e.g., enterprise search, cloud and/or crowd-based solutions, risk and virtual organizations. Demonstrate the following skills and abilities • Apply relevant Information Technology solutions using appropriate strategies and paradigms in the context of providing a prototypical IT solution.

Completing Your Assignment

What am I required to do in this assignment?

From the UIF

Group work – develop (75%) and present (25%) a software system based on a system architecture (provided by the teaching team) to demonstrate programming and software development skills and the ability to work in a team. Students are expected to spend 36 hours for preparation and 9 hours for the completion of the assessment. Students will be individually assessed based on the artefact, combined with a group report, peer assessment form and the unit tutors' judgement of their performance during their presentation.

The total effort required to complete both assessments parts is 60 hours.

Assignment description

Work with colleagues in your assigned/formed group to develop and implement your group's **case study/task description** which you find at the end of this document.

Submit a word-processed Group Report detailing the work that you have done as part of the group. This document must include the filename and page numbers in the footer. You will also need to submit an Individual Reflective Report. All members of the group must be clearly identified (providing name and student ID) on the title page.

Each group will be given a mark, but each group member will also be assigned an individual mark. Your individual mark will be influenced by your individual contribution to the development of your program as evidenced in your Group Report, your contribution to the group presentation, and the Individual Reflective Report.

Groups

The size of the groups should not be larger than five (5) people. Assignments by a group of two (2) people are also acceptable in exceptional cases that need to be approved by the unit team. However, you will not gain any advantage by doing so.

You will need to sign-up for one of the groups made available on BREO. You will also need to provide student ID numbers and names for all group members at the front of the submitted Group Report. Only one Group Report should be submitted by each group. If your name does not show on the report, this means you have not attempted this assignment.

If there is a dispute within the group, you should inform the lecturer or unit coordinator, so that a meeting with the group can be held within the first two weeks of the assignment where the matter can be resolved.

Tasks

1. Consider the task description as a group. In your group, agree on the features to be part of your solution to the problem described, and on how you might be able to deliver these features. (This includes determining what data may be required to support the functionality in each case). List and describe these assumptions fully.

2. Develop an appropriate database to store all the required **Global Music Concert/Festival Booking Agency** system information and to retrieve it once needed for operations and dissemination. Develop the requested client/server system with a GUI-driven client and a server that handles appropriate requests (see Task Description).

3. **Code** and **test** the critical features of the **Global Music Concert/Festival Booking Agency system** you have analysed using programming facilities of Java language such as I/O file operations, Networking classes, JDBC etc.

   Hints:

   - Develop GUIs for each of the forms you need to use within the system as user interface. Note that you have 4 different users: Customer, Corporate Organisation, Administrator and Concert/Festival Organiser.

   - Make sure the database is deployed on the server side, and the users access this via a GUI-based client program.

   - Please carefully design your database using your database design and SQL skills.

Deliverables

1. **Group Project Report (10-20 pages with Code Appendices)**

   Illustrate and justify your choices in analysis and implementation. Provide evidence that you actually tested your code. Include and explain your class diagram. Any code may go into the appendix, except core classes that you may explain in the main text. Please do not exceed the page limit as you may be penalised.

   This is a formal report and attention should be given to the format of this work. You should include a title page, table of contents, heading and sub-headings (introduction, main part, summary, appendices), and captions for figures. You are strongly advised to use the report template provided.

   A Group Report template is provided on BREO.

2. **Presentation / Demonstration (5 slides max with demonstration of your program - approx. 10 mins total)**

   Provide an overview of your work as a group and demonstrate your working system.

3. **Individual Student Reflective Report**

   A confidential Student Reflective Report (one from each student) to be submitted using the appropriate link on BREO. This should reflect on the ease or difficulties that you encountered when completing this group assignment. It should include how you worked with the others in the group to produce the final report and program, the difficulties you encountered and how these problems were overcome, your evaluation of your own contribution to the group work as well as the contribution of others in the group, what you have learned from doing this group work, and how would you do things differently in the future.

   An **Individual Reflective Report** template is provided on BREO.

4. **Complete program code** submitted as a zipped file

Note: The tutors will be happy to advise you at any stage during the development of this program.

Is there a size limit?

See above.

What do I need to do to pass? (Threshold Expectations from UIF)

In order to pass Assessment 2 you will need to:

- Demonstrate application of your knowledge and understanding of the development and implementation of an appropriate software solution.
- Present and discuss your own contribution of at least 15% of the total group work with reference to the assumed role in your group.

How do I produce high quality work that merits a good grade?

Contribute equally to the design and implementation of the application under consideration. Contribute equally to the report and presentation. Collaborate and communicate proactively with group peers.

During the presentation, each group members should be able to demonstrate their own contribution and defend the decisions made.

How does assignment relate to what we are doing in scheduled sessions?

The assignment lets you critically reflect and apply the programming solutions and methods presented in the lecture, in order to solve a given problem.

Marks and Feedback

How will my assignment be marked?

Your assignment be marked according to the threshold expectations and the criteria on the following page.

You can use them to evaluate your own work and estimate your grade before you submit.

	Lower 2nd – 50-59%	Upper 2nd – 60-69%	1st Class – 70%+
JDBC (30%)	Good database design. Connectivity is effective.	Very good database design. JDBC connectivity working effectively.	Excellent database design. JDBC connectivity working effectively.
GUIs and Forms (30%)	Forms are complete with an acceptable design.	Forms are working well and complete with a good design.	Forms are working and integrated with excellent design.
Report (30%)	Good structure and acceptable content.	Very good structure and content.	Excellent structure and content.
Presentation (40%)	Good presentation. Demonstrates a good level of subject knowledge.	Very good presentation. Demonstrates a very good level of subject knowledge.	Professional presentation. Demonstrates an excellent level of subject knowledge.

Task Description

Global Music Concert / Festival Booking Agency System

Consider the following scenario:

The number of festivals is increasing by the year, but why are they becoming so popular?

Many people are attracted to the idea of a festival for a day out with good atmosphere and everyone enjoying something they love. Whether it be food or music, if it is something you feel passionate about, a festival dedicated to it is right where you want to be!

The most popular type are music festivals, these tend to extend over a weekend or can just be for a day and are typically held annually in the same place. Over the last 15 years the number of them in the UK has increased significantly. Holding the festival outdoors allows the event hosts to sell large amounts of tickets. With lots of people attending there are a wide range of opportunities for sponsorship deals with big brands where significant amounts of money can be generated for the event hosts. Artists also make a lot of money by performing at festivals and tend to enjoy taking part to perform for large audiences as it can increase their recognition.

Global Music is a startup Concert / Festival Booking agency. They have asked you, initially, to create a prototype desktop application which will eventually be available as a browser-based web application and a mobile app.

Potential customers include the general public and corporate organizations. All customers (whether general public or a corporate organization) initially need to create an account by registering their details and creating a username and password. Details to be recorded could include Title, First Name, Last Name, Address 1, Address 2, Town, Post Code, Contact Name, Organisation Name, Email Address, Phone No, Web Address, Username, Password, Payment Method, Account No, Corporate/General Public.

All customers then need to login so that they can edit their account details, view their previous bookings, view promoted concerts and festivals, view the main bands involved, and make a booking for one or many people for a concert. Payment needs to be made at the same time as the booking if made by the general public. There is no requirement to implement this in the initial prototype. Corporate organisations can be invoiced by arrangement on a monthly basis. There will be a range of ticket prices (ie. A to C) for each concert – including a corporate price. Only corporate organisations can choose this option. Customers cannot cancel a booking once made.

The system administration will be able to login to the system and view all available concerts/festivals, view all bookings, and confirm a booking on receipt of payment. They will also need to invoice corporate organisations on a monthly basis.

The booking is for a concert/festival at a particular venue at a specific time.

A concert will have a number of bands, and each band will have an agent. The agent arranges the booking for the band at any particular concert.

The festival organizer arranges promotion of the concert with Global Music's booking agency. The festival organizer needs to be able to login to the system and add concert details, edit concert details and cancel concerts when necessary.

Occasionally, a concert needs to be cancelled. All bookings for the concert are then cancelled automatically, and the customers and corporate organisations are notified by the system administrator.

You need to create a GUI desktop application to meet the requirements of this scenario. A database needs to be created to store customer information and booking information, as well as concert/festival information, band details and agent details.

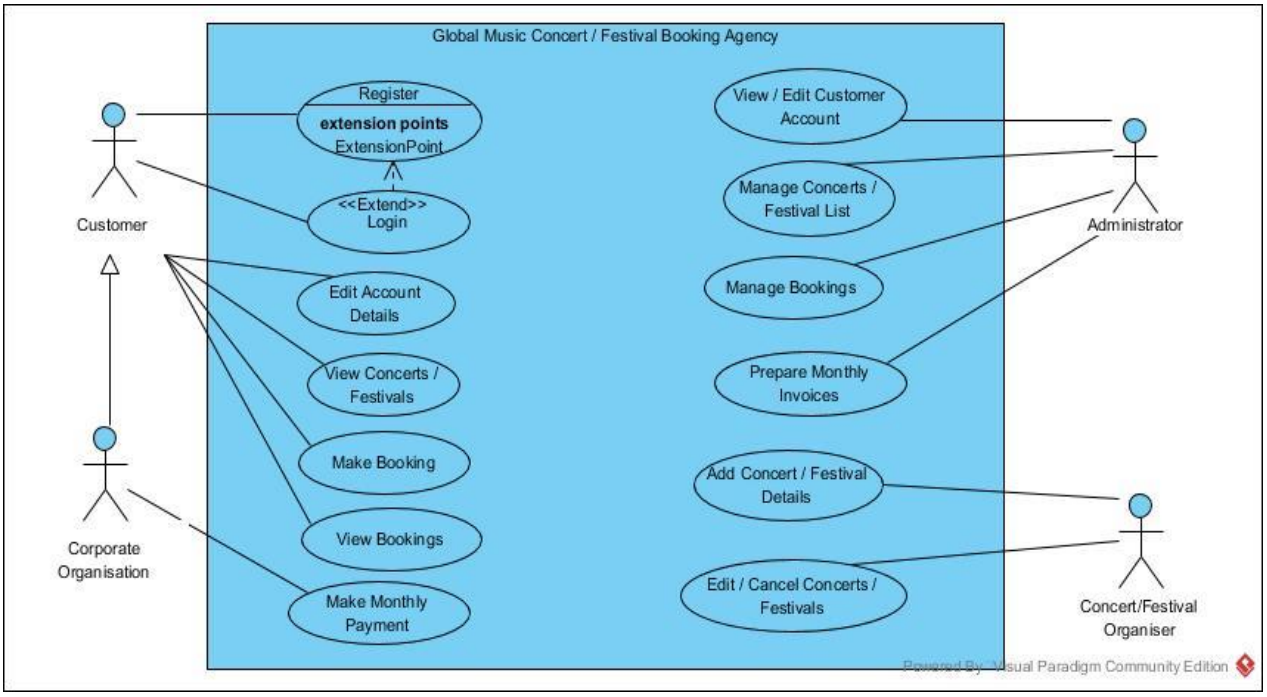

Figure 1: Global Music Concert / Festival Booking Agency System Use Case Diagram

The task of your group is to implement the **Global Music Concert / Festival Booking Agency System** containing the functionality shown in the Use Case Diagram in Figure 1, taking the different user roles (customer, corporate organization, administrator, and concert/festival organiser) into account.

Please note that client-server architecture (network/sockets programming) as outlined below is not essential in order to achieve very high marks for this assignment.

Ideally, the whole system should be based on a client-server architecture, where the database is deployed on the server, and the users are given access to the database via the client graphical user interface (GUI). The server should speak to the database and the client GUI should speak to the server only (not to the database). Figure 2 depicts this situation.

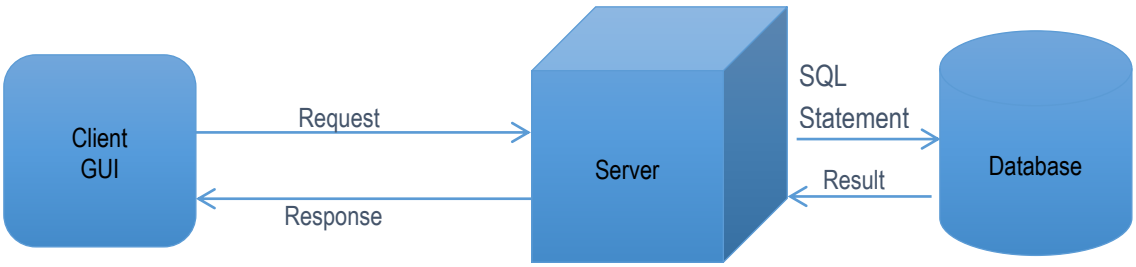

Figure 2: System Architecture

Clients should send simple requests with parameters (for instance "book_concert, customer123, 08/03/2019, c244, 9" to send a new booking to the database or "view_bookings customer123" to view the bookings for a specific customer) to the server. The server should process these requests and create suitable SQL statements. Results from the database should be set up accordingly and sent to the client as a response. For instance, if the request was "view_bookings customer123" the server should return the bookings assigned to customer123 fetched from the database. One task of the server will be to translate the requests to SQL statements and turn the results from the database into a proper response to the client.

To get the maximum full marks (100/100), you do not need to implement a client/server solution as indicated above. Groups who do not follow this architecture (where for instance their GUI directly accesses the database) will still be able to pass with the maximum achievable marks. Groups who implement the required functionality but without a database/JDBC and a server will attract 60/100 marks maximum. You need 40 marks to pass. This is a group assignment but an individual assessment, but single group members may attract different marks based on their contribution to the group (determined for instance by peer assessment and the presentation).

Background: Why a client/server application?

You may ask yourself, "Why do we have to develop a server? Isn't it sufficient when the client GUI speaks to the database directly?". The reason is that many of today's applications are like this. (If you create a Facebook client, do you want it to talk to the Facebook database directly or wouldn't it better to make use of Facebook's API, which would be more convenient?).

One of the advantages of the client/server architecture is that it provides a simple API for clients so they don't have to bother with storage and database issues.

Consider the case where you no longer want to store the data in a database, but in some other data structure - for instance files, a Cloud storage or a NoSQL database. (Services like Google and Facebook have been changing their backend storage several times in their history). If each client speaks to the database directly (using embedded SQL commands), you would have to re-implement each single client (and there can be several client implementations around for a server). In a client/server scenario, you would only need to update the server implementation to cope with your new storage. No client needs to be touched. Therefore, there are good reasons to provide an API and a further abstraction layer such as we propose in this assignment.