University of the West of England
BRISTOL

MODULAR PROGRAMME

COURSEWORK ASSESSMENT SPECIFICATION

Module Details

Module Code UFCF85-30-3	Run 18SEP/1	Module Title Enterprise Systems Development
Module Leader Chris Simons, Mehmet Aydin	Module Coordinator	Module Tutors Chris Simons, Mehmet Aydin
Component and Element Number B: CW1		Weighting: (% of the Module's assessment) 42%
Element Description GROUP-BASED DEMO (Group-based demonstration of software development)		Total Assignment time

Dates

Date Issued to Students: 25/10/2018	Date to be Returned to Students 25/10/2018
Submission Place: Blackboard (as a complete zipped Netbeans project) Plus: In-class demonstration	Submission Date 06/12/2018
	Submission Time 2.00 pm

Deliverables

1. Zipped Netbeans project
2. In-class demonstration in weeks identified in the text

Module Leader Signature

Dr Mehmet Aydin

Enterprise Software Development (UFCF85-30-3)

Group Coursework Assignment

Description

You (as a group/team) are asked to design and implement a web-based software system using the features and functionality of the examples from the course book, the practical classes, demonstrations in the lecture or other materials referred to in the scope of the course.

The assignment is based on the Alpha-Cab case study, which is provided at the end of this document.

You should design and build your software system using JSP/Servlets using the MVC design pattern on Java Enterprise Edition (EE), and deploy to a Glassfish server with a local Java DB Relational database backend.

As a team, you will be required to demonstrate and discuss your working system using NetBeans IDE. Your team will be treated as a unit and any individual member may be required to demonstrate complete knowledge of the system you are presenting.

You are required to develop your ***web-based software system*** in three sprints as follows:

Sprint 1: Task plan and schedule of tasks. This sprint mainly requires (1) working out with planning and scheduling the tasks, (2) making sure the provided DB is connected and (3) tested with Login function for all user types. Initial deliverables will be assessed on ***Week 8***.

Sprint 2: Completion of backend architecture and partial completion of front-end user interface. This requires, at least half of the tasks are completed. These tasks are expected to be completion of some model classes, controllers and few viewer classes. Deliverables will be assessed on ***Week 10***.

Sprint 3: Final and complete delivery, will be assessed on ***Week 12.***

Assessment System

You are required to submit your final work in the Blackboard VLE as a zipped NetBeans project.

As a team, you will be asked to demonstrate your developed software system in scheduled practical classes as applicable for each sprint. Where appropriate:

- You must download (from Blackboard) and unzip your project.
- You must run the SQL scripts provided to create and populate the required tables.

Your NetBeans project and your DB server must be runnable on the standard FET configured laboratory machines (or the same configuration on your own laptop which you must bring).

- It is your responsibility to attend scheduled classes – failure to demonstrate your system in class will be treated as a non-submission.
- All group members will be awarded the same mark as long as all group members agree that all contributed equally. Otherwise, your individual marks will be generated based on your individual contributions identified based on collected and provided evidence. A guiding scheme is developed and added to the document called “Marking Matrix”

- Any group member failing to take part in the demonstration will be assessed as a non-submission and given zero marks.

The quality of your verbal expression in this demonstration is important – incoherent explanations will not achieve high marks. Please be advised that demonstrations last for a fixed-durations, so be prepared to concisely demonstrate and explain your system.

Important Notes:

Please also be advised that:

1) Teams are expected to use GitHub as the version control software to ease collaboration within the groups and to create evidence of their team work. Other software development communication tools are available and may also be used to generate evidence of team work.

2) ***A Task Delivery Form*** should be filled and signed by all group members and submitted to the tutor prior to each assessment stage. A copy of the Task Delivery Form can be found on BB and be downloaded. *Please ask for guidance if you need further clarifications regarding use of Task Delivery Form.*

3) The developed software system should

   (i) use Java EE components following MVC patterns,
   (ii) interact with a Java DB database, and
   (iii) deploy to a GlassFish container on a server.

4) NetBeans should be used and IDE during the demonstration.

5) You may assume the system will accept cookies. No threading/concurrency considerations are required. All data must be stored and retrieved from the Java DB database, and the system should be accessible on localhost deployed on Glassfish server.

6) You will be supplied with one SQL script to build tables containing some samples of drivers, customers and journeys. (**Alphacab.sql**). You must show that your database contains only the drivers, customers and journey details, as provided in the file.

Please note that each group member should be prepared to perform any of these tasks or explanations._You will be asked to show and explain your code - which must be readable and commented.

Marking

Sprint 1

A 10-minute presentation + Q/A session is needed from each group while all group members must attend. A set of slides can be used for presentation and **(1)** detailed plan, ideally in the form of Gantt Chart, and **(2)** a Task Delivery Form is needed to be filled, signed and submitted to the tutor, and **(3)** a quick demonstration of Login function.

Sprint 1 carries a weighting of 10% of the overall coursework marks.

Sprint 2

Completion of backend architecture and partial completion of system functionality including some servelets and front-end user interface. A 10-minute presentation + Q/A session is needed from each group while all group members must attend. A set of slides can be used for presentation and **(1)** detailed plan for the remaining tasks, ideally in the form of Gantt Chart, and **(2)** a Task Delivery Form is needed to be filled, signed and submitted to the tutor, **(3)** a quick demo of few completed functionalities.

Sprint 2 carries a weighting of 30% of the overall coursework marks.

Sprint 3

Sprint 3 carries a weighting of 60% of the overall coursework marks.

For sprint 3, additionally –

1- A clear and understandable use of "**Web Services**" is requested that be incorporated into the system - merely for demonstration purposes. This can be for calculation of the charges per journey.

2- The distance between two addresses can be calculated via **Google map services**, which can be used in a Web Service that calculates the charge of taxi services.

3- It can also be useful for the system to use "**Filtering**" for admin user authentication purposes- a clear example of this would also be favourably considered (“**Filtering**” is discussed in Chapter 13 of reference book, HF Servlet & JSP).

Marks for sprint 3 are allocated as follows:

Step 1: The Test Cases.

1. Login to the system as a typical user (e.g. user = "admin", password= "admin") and create a Session lasting for 20 minutes.
2. List all drivers and all customers registered for the head office
3. Show the list of jobs for each driver upon request
4. Create a daily report (including the turnover, and the number of customers served)
5. List all customers served per day including the destinations and the charge incurred.
6. Let a customer book a cab for a particular date and time - (if it is the 1st time, the office should add a new customer record, the office should assign the customer to an available driver).
7. Add more drivers to the list
8. Create a customer invoice - showing incurred cost and VAT for a total cost.
9. Remove a driver
10. Change the price of a destination

Each successful test case is worth 4 marks (total 40).

Each test must be demonstrated sequentially and as described. Marks could be lost – e.g. for a lack of clarity or failure to demonstrate tests sequentially. Do not combine the Test Cases - each must be a distinct operation. Normally a test will either pass or fail.

If you think there may be some ambiguity in the specification or Test Cases you must check for clarification with the tutor in your normal practical class. False assumptions may cost you marks.

Step 2: Design Walk Through.

You should briefly and concisely explain the design and implementation of the following features of your system (50 marks)

- The MVC pattern and project file structure (15 marks)
- The relational database (10 marks)
- Sessions (5 marks)
- Web Services (develop and use) (10 marks)
- Using Filters (5 marks)

Step 3: Team Work Evidence.

Your team work will be assessed based on the evidence you provide via GitHub version control software and other team-based software development tools. (15 marks)

- Evidence of each group member's contribution (5 marks)
- Task management and allocation (5 marks)
- Clear history of collaboration (5 marks)

The Case Study

Alpha-Cab Ltd is a minicab company, which is planning to develop a WEB-based booking system to organise their services giving access to all stakeholders including customers in order to boost the quality of the services. The company has one head office to plan and orchestrate the operation, a number of drivers to deliver cab service, and a number of registered customers.

The head office keeps all historical data and business rules, and operate the whole system day-by-day. The operations include registering new drivers (cars) and new customers, and keeping drivers' and customers' records up-to-date. The head office also conducts booking operations assigning each customer request to a driver based on driver's availability.

A customer is expected to book a car for a particular transportation with date and time using a client interface, where the following customer information is needed for booking: Name, Address, Destination address, date and time. Once a request is booked, the corresponding fee should be calculated, an invoice to be issued and the system to be updated. A driver can update the system once the transport is completed and the fee is paid.

The system should be operated via a database (DB), where the connectivity should be maintained via a server and DB operations should be carried out using SQL. Other relevant processing will be done by Java EE components orchestrated by Servlet(s). The company expects three different client types: Head office (admin), Customer, Driver. **Web-based** access is to be granted to all customers.

The drivers should be able to get the list of the jobs assigned to them each day, while the head office should be able to display all the bookings completed per day by each driver, and be able to calculate their daily turnover.