# HIT3119/8119 Assignment 2

## *Purpose of Assignment:*

To familiarise students with good practices in enterprise system architecture and J2EE technologies

## *Important Dates:*

| | |
|---|---|
| ***Submission of Team formation sheet*** | 5:30pm Thursday 14th of September (hand in at start of Lecture) (Week 8) |
| ***Notification of Team Id*** | 5:30pm Monday 8th of October (Week 10) |
| ***Submission of Assignment*** | 9:30am Monday 23rd October (Week 12) |
| ***Notification of Demo Session*** | 5:30pm Tuesday 24th October (Week 12) |

## *Team Id and Submission of Team Formation Sheet*

This is a group assignment. A group of 3-4 students forms a team. You will be assigned a Team Id after you have submitted the Team Formation Sheet by the due date (See Team Formation Sheet at the end of this document).

You can submit a Team Formation Sheet electronically as an email attachment in to guy@conchus.com or hand it in during the lecture on week 8. Here are the instructions for the electronic submission:

- Scan the properly signed (by all team members) Team formation sheet into a pdf or graphics file
- Send the pdf file or graphics file to guy@conchus.com with the subject "HIT3119/8119 Assignment 2 Team Formation" and cc a copy to all team members involved in the team.

## *Submission of Assignment*

All source and documentation will be submitted. Submission is via ESP (https://esp.it.swin.edu.au/). Zip all your files into 1 and name it as "teamXXA2.zip" where "XX" is your Team Id and "A2" means Assignment 2. The size of the zip file will be set to 20MB. Please let me know if your file is larger than 20MB.

## *Demo Session*

A demo session will be organized during the lab hours in Week 12. You will be notified about this after your assignment team has been formed (See Important Dates above).

## *Mandatory Deployment Requirement*

You should be able to deploy your application to a local instance of Sun Java System Application Server Platform Edition 9, hereafter referred to as the Application Server or simply AppServer whenever it is clear from the context. It is your responsibility to ensure that your application can be deployed on Application Server 9. If your application cannot be deployed on to Sun's Application Server 9 in the FICT's lab machines or on your laptop during the demo session, you will not get the marks allocated in the "Function" category of the Mark distribution (See Mark Distribution below).

**Contribution to Final Assessment: 25%**

## *Introduction*

This system is going to allow customers to book vehicles, drivers to review scheduled work, managers to assign vehicles and drivers to jobs and generate a series of simple reports view the operations of the company. The system can get a little tricky because there are different types of clients and cars and drivers with different kinds of licences. Please read through the project requirements carefully and then if in doubt please post queries to the discussion board on blackboard.

## *Software Requirements*

This assignment involves the design and development of a server-based chauffeured vehicle booking system for Swinburne Swish Sedans and Limos. The system should use JSF, JSP, Servlets, and Enterprise Java Beans. It should be deployed to Sun Application Server 9. The target database is Derby, which is bundled with the AppServer. There are three parts to this project. They are

- a written design document
- the source code and the packaged deployment files – zipped as one file, and
- written instructions on how to run your client applications.

## *Software Requirements*

Please note that all requirements are subject to change.

## Users

There are four kinds of users in the system:

- Casual Clients
  - These users don't have to log onto the system but from the public website can request a booking for a vehicle.
- Corporate Clients
  - These user have to log in and once logged in can:
    - View upcoming jobs
    - Book a new job
    - View a statements of activity (i.e. History of jobs done)
  - They can update their details
- Drivers
  - These users have to log in and once logged in can see what jobs they are assigned and what vehicles they are using for each job.
  - They can update their details except type of license they have (must be done by manager who has viewed the license)
- Managers

- These users have to log in and once logged in can:
  - View new bookings (jobs) and assign drivers and vehicles to jobs
  - Add/Edit/Delete bookings
  - Add/Edit/Delete vehicles to system (what if a vehicle is already assigned to a job)
  - Add/Edit/Delete drivers to system
  - Add/Edit/Delete corporate customers to system
  - Add/Edit/Delete other managers to the system
  - View reports
    - Jobs done by drivers
    - Jobs done by vehicle
    - Jobs done by corporate customers

## Login

Users can log into the system using an id and a password. For the purpose of this assignment, you are not required to use JAAS (Java Authentication and Authorization Services) to support the user login and the user credentials.

## Bookings

A booking (job) will need to have information like:

- Contact Name (person arranging/paying for booking)
- Contact Phone and/or Email
- Customer's Name (person being picked up)
- Pick Up Date
- Pick Up Time
- Pick Up Address (where to pick up customers)
- Drop Off Address (where to end the journey)
- Booking Type
  - Transfer (Just take customer from pick up to drop off)
  - Hourly (Driver and car will be on the job for X number of hours)
- Event Type
  - Wedding / Party / Special event (can only be done with “Hourly” booking type) [SV License]
  - Other (airport drop off/pick up, hotel transfer etc.) [VH License]
- Vehicle Type (you might want to allow customer to select from models and colours in system)
  - Luxury Sedan
  - Limo

- Special requirements
  - baby seat
  - wedding ribbons
  - other
- Free text field for notes

Once a booking has been entered then if it is:

- An anonymous customer: Ask for credit card details and then assume that transaction went fine and enter booking request in the system (no need to hook up to credit card gateway for assignment)
- A corporate customer: Just accept the booking and enter it in the system. The client should them be able to view the booking in the upcoming jobs view.

What if a client books a job in the past? When a manager is assigning cars and driver are they double booked? Is the driver and vehicle licensed correctly for the job? What other things should you check for before accepting a booking?

## Licences

Please note that I am making up these license names and conditions.

Vehicles have **one** of two licences:

- SV License (Special Vehicle License): Is fairly cheap to purchase but means the vehicle can only be used for special events like weddings, parties and débutantes etc.
- VH License (Vehicle Hire License): Is quite expensive but means the vehicle can do SV jobs and also do things like pick up and drop off customers at the airport or hotels etc.

Drivers have **one** of two licences:

- HV Licence (Heavy Vehicle License): Is fairly cheap and easy to get and allows the drivers to drive limousines (as well as sedans). They can only drive on special events (SV License) and not do any pick ups or drop offs. They can drive a VH vehicle but only use it like a SV vehicle.
- Taxi Licence: This is more difficult and expensive to get but allows a driver to do pick ups and drop offs as well as special events. A driver with a Taxi license can also drive a limousine but can only use the limousines for pick ups and drop offs if it has a VH license.

## System Views

Because each user has access to different functionality and data then it makes sense to have several GUI views for each type of user or maybe even several web apps. How you solve this problem is up to you but consider things like:

- Only Mangers can add vehicles and drivers
- Divers should be able to see only their
- Is it a good idea to separate the public and administrative views into separate applications for security reasons (administrative application can be run on Intranet only)
- etc

## Reports

As mentioned the Manger and Corporate client want to be able to view historical and future operations. All these need to be are web pages that list the desired data and allow the user to do some filtering i.e. select what driver or what customer to filter results on.

Please remember that a Manager should be able to see a fairly detailed report listing things like exact car and driver assigned to each job but a Corporate client should not have access to this level of detail (Manager should be able to change the vehicle to another 6 seater silver limousine without concerning the Corporate client).

With the reports being quite details how you display the data is up to you... feel free to be creative and use thing like tool tips and icons. But bear in mind that a public website has to be accessible to lots of different browsers and configurations... though you can stipulate that a Manager must use a certain browser at a minimum resolution because there will only be a few managers and they are demanding a rich interface.

## *Architecture and Design*

The application should use JSF, JSP, Servlets and Enterprise Java Beans. The application should

be layered into clean tiers. Session and Entity Beans should be used where appropriate. Give careful thought to design and try and produce some Use Cases, ER diagrams and Acceptance Test to clarify the system before you start coding.

This is a very BRIEF technical requirements document so please feel free to make some assumptions but document them so you can show us that you had considered the issues. Please post your queries on the "Discussion Board" rather than emailing your tutor so other students can see if their questions have already been answered.

## *Written Instructions*

Please provide clear and concise instructions for all functions of the system including installation instructions. Penalties may be incurred if the system does not work as described or instructions are not clear.

## *Mark Distribution*

A details marking sheet will be distributed several weeks before assignment submission.

| *Category* | *Item* | *Marks* |
|---|---|---|
| Functionality | Login | 5 |
| | Create Bookings | 5 |
| | Manage Bookings | 10 |
| | Manage Customers | 5 |
| | Manage Drivers | 5 |
| | Manage Vehicles | 5 |
| | Reporting | 5 |
| | X-factor | 10 |
| | **Sub-Total** | **50** |

| *Category* | *Item* | *Marks* |
|---|---|---|
| Design and Coding | Overall architecture and use of object orientation | 10 |
| | Database Design | 5 |
| | Appropriate use of JSF, JSP and Servlets | 5 |
| | Appropriate use of Enterprise Java Beans | 10 |
| | Clear instructions | 10 |
| | Code Quality | 10 |
| | **Sub-Total** | **50** |
| | **Total** | **100** |

## Functionality

You will be assessed based on your submitted applications. It is expected that your application will run during a demo session to be arranged soon after the submission. If your application cannot produce correct results of a particular function on a lab PC or laptop during the demo session, you will not get the marks allocated for that particular function.

## Design and Coding

### *Design Documentation*

You will need to write a small and brief design document. This document should outline your system architecture and design. It should contain the overall architecture of your proposed application, some Use Cases, ER diagram and Acceptance Tests. It will explain your design for the application in sufficient detail so that it can be used for implementation by another team. Consider the following in your design:

- Overall architecture
  - What the components of the application are
  - To what tier the components belong
  - How these components interact with each other
  - J2EE technologies utilised
  - Server-side interaction & integration
- UI Design
  - How is using the system and how
  - Are your screens capturing the necessary information
  - Where and how is validation being performed

- Database Design
  - Is the schema normalised
  - Can the system be extended easily

*Note: This document is very important for your understanding of the theory behind J2EE. It will help you prepare for your examination*.

### *Instructions*

You need to write instructions explaining how to run your client application. For each function it is expected that the instructions should be at most 1 page. NO screen dump is required.

### *Code Quality*

Marks will be awarded on quality of implementation, including the user interface.

## *Code Libraries & Plagiarism*

The use of third-party code libraries, particularly Open Source, is encouraged. Please restrict library usage to utility and infrastructure resources. Core business logic and functionality MUST be implemented. Plagiarism WILL NOT be tolerated. All third-party libraries and code must be acknowledged in the Design Documentation. Failure to properly acknowledge code will be marked down significantly by 30% of your marks.

All submitted code will be analysed for similarities with other submitted work to detect plagiarism.

**Swinburne University Of Technology**

**Faculty of Information and Communication Technologies**

# TEAM FORMATION SHEET

**Unit Code:** HIT3119/HIT8119 **Unit Title:** Enterprise Java

**Convenor:** Guy Gershoni **Due date:** 8:30am Monday Week 6

Team Id will be assigned and released on Blackboard in Week 10.

| ***Student Id*** | ***Student Name*** | ***Tutorial*** | ***Tutor*** | ***Signiture*** |
|---|---|---|---|---|
| | | | | |
| | | | | |
| | | | | |
| | | | | |

## Preferred Demo Session*

| ***Preference*** | ***Lab Session and Location*** (e.g. "Tue 3:30pm – 5:30pm EN310" or "any time is fine" | ***First Hour / Second Hour*** (e.g. Second Hour) |
|---|---|---|
| 1 (most preferred time) | | |
| 2 | | |
| 3 | | |
| 4 (least preferred time) | | |

* Normally, 2 – 4 teams will be scheduled in an hour depending on the total number of teams. Preference will be given to those who submitted the form the earliest (based on date and time). In case, your choice of a particular session is full you will be scheduled to an empty time slot based on any team member's registered lab hours by the lecturer.

If you do not choose a time then a time will be assigned to you by the lecturer.