



**Swinburne University of Technology**  
*Faculty of Information and Communication Technologies*

**HIT3119/8119 Enterprise Java**  
**Assignment**

**Purpose of Assignment:** To familiarise students with the architecture and technologies of Java EE 5.

**Important Dates:**

Initial version Submission      May 3<sup>rd</sup> / 4<sup>th</sup>, 2010 (submitted at the start of the Lab class)

Demo Session:                      May 3<sup>rd</sup> / 4<sup>th</sup>, 2010 (demo to the tutor in your Lab)

Final version Submission      May 24<sup>th</sup> / 25<sup>th</sup> 2010 (submitted at the start of the Lab class)

Demo Session:                      May 24<sup>th</sup> / 25<sup>th</sup> 2010 (demo to the tutor in your Lab)

**Submission of Assignment:** All source and documentation will be submitted.

**Demo Session:** A demo session will be organized on May 3<sup>rd</sup> / 4<sup>th</sup>, 2010. You will be notified about this after the tutor has organized the required information.

**Mandatory Deployment Requirement:** You should be able to deploy your application to a local instance of Sun's GlassFish Enterprise Server version 2.1, hereafter referred to as the GlassFish whenever it is clear from the context. As the development and deployment tools provided by NetBeans may change quite often, it is possible that your application may not be able to be deployed under Sun's GlassFish Enterprise Server version 2.1 even though it can be deployed on newer version such as Sun's GlassFish Enterprise Server version 3.0. ***It is your responsibility to ensure that your application can be deployed on GlassFish Enterprise Server version 2.1. If your application cannot be deployed on to this particular version in the lab machines 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:** 40%

**Requirements:**

**Customers and Bank Tellers**

Customers can access their own accounts in three different ways. First, they can use the Internet Banking services through the Web front end, which is developed using Servlet, JSP or JSF technologies. Second, they can use Automatic Teller Machines (ATMs) where client applications are installed. Third, they can do their everyday banking with the help of Bank Tellers over the counter. The client applications are developed in Java. *For the purposes of this assignment, these client applications run on PCs configured as in our labs and, the passwords for the Web front-end and for the ATM of a customer are the same.*

Bank Tellers can only access customer's account via the computer terminals where client applications are installed.

**Login**

Both Customers and Bank Tellers can login to the system using an id and a password. *Bank tellers should use their own customer id and password, not their staff id and password as bank tellers, to access their customer accounts.*

**Standard Bank Accounts**

A standard account can be either a savings account if the customer is a person or a company account if the customer is a registered company. *For the purpose of the assignment, functionally we do not distinguish savings accounts from company accounts.* A standard account has an account number and a balance. The balance should never be below 0. A customer can have more than one account.

**Cheque Accounts and Credit Card Accounts**

*For the purpose of the assignment, Cheque Accounts and Credit Card Accounts are not considered.*

**Creating and Removing Accounts**

Customers cannot create and remove their own accounts. Bank Tellers should be able to create and remove accounts for Customers.

**Account Balance**

Customers can only view the *latest* account balances of their own accounts whereas Bank Tellers can view the *latest* account balance of any account.

The balance of an account will be calculated and updated whenever there is a transaction on the account. On the twentieth day of every month, the balance will then be archived as a financial record storing the account number, the balance, the date and the time when the financial record is created. The time is the system time when the financial record is created. *For the purpose of this assignment, it is assumed that the time is 0:01am on the date of processing.*

**Interest Calculations**

*For the purpose of the assignment, interest calculations are not considered.*

**Transaction History**

Customers can only view the transaction history of their accounts whereas Bank Tellers can view the transaction history of any accounts. All transactions including money transfers, bill payments, deposits and withdrawals within the requested period should be included in the history.

**Deposit and Withdrawal**

Customers cannot deposit and withdraw money via the Internet. In order to do so, they must be either at the ATMs or over the counters with the help of Bank Tellers.

**Money Transfer**

Money can be transferred between accounts both internally and to other banks. *For the purposes of this assignment, you only need to provide transfer facilities within the Swinburne Bank.*

Customers can transfer money between their own accounts. They can also transfer money from their accounts to other persons' accounts which must be registered with the source account beforehand. *For the purposes of this assignment, you only need to implement fund transfer to Customers' own accounts.*

Bank Tellers can transfer money between any accounts.

### **Bill Payment**

Customers can pay bills through the system via the Internet only. Billers have a code and a name. Customers have a reference number provided by the Biller. The customer should be able to select the account from which to pay the bill, find the appropriate Biller and pay the bill. *For the purposes of this assignment, assume that the Billers have company accounts in Swinburne Bank and the account id is the same as the Billers' code.*

Billers can find out how much a customer has paid within a period of time because the customer can make partial payments through different accounts. *For the purposes of this assignment, you are not required to implement this function. However, your database design should allow this extension.*

### **Architecture and Design**

The application should use Servlets, JSPs, JSFs, Enterprise JavaBeans and Entity Classes where appropriate. The application should be layered into clean tiers.

### **Instructions**

**Please provide clear and concise instructions for all functions of the system.** The instructions should include how to run different client applications (e.g., the Web application, the Client application on the ATM and the Client application over the counter). Penalties may be incurred if the system does not work or instructions are not clear.

### **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 50% of your marks. All submitted code will be analysed for similarities with other submitted work to detect plagiarism.

### **Database Design**

Here is a guideline on the database tables for you to consider. Depending on how complicated or how flexible you want your applications to be, **you may need to add extra data fields and, even, extra data tables** for your applications to suit your needs. Make sure that you document the Database Design section in your document.

**Table 1: Login Table**

Field Name	Description	Type	Memo
CustId	Customer Id	6 characters	Primary Key. All are digits
Password	Password	6 characters	All are digits.

**Table 2: Account Table**

Field Name	Description	Type	Memo
AccountId	The identifier of the Account	14 characters	Primary Key, all are digits
CustId	The identifier of the customer who owns the account	6 characters	See Table 1
AccountType	The type of the account	Character	Can be either B or P where B and P mean Business and Personal, respectively
Balance	Current Balance of the Account	Numeric	

**Table 3: Transaction Table**

Field Name	Description	Type	Memo
TransId	The identifier of the transaction	10 characters	Primary Key. All are digits.
TransType	The type of the transaction	Character	Can be either B, D, T, or W where B, D, T, and W mean Bill payment, Deposit, Money Transfer, and Withdrawal, respectively
Date	The date when the transaction is processed	Date	
Time	The time when the transaction is processed	Time	
FrmAccountId	The identifier of the source account	14 characters	All are digits (See AccountId in Table 2).
Amount	The amount of the transaction	Numeric	
RecAccountId	The identifier of the receiving account	14 characters	All are digits (See AccountId in Table 2).
Description	A description of the transaction	128 characters	

**Table 4: Financial Statement Table**

Field Name	Description	Type	Memo
StatementId	The identifier of the financial statement	6 characters	Primary Key. All are digits.
Date	The date when the financial statement is created	Date	
Time	The time when the financial statement is created	Time	
AccountId	The identifier of the account of this financial statement	14 characters	All are digits (See AccountId in Table 2).
Balance	The balance of the account when this financial statement is created	Numeric	

**Table 5: Biller Table**

<b>Field Name</b>	<b>Description</b>	<b>Length</b>	<b>Memo</b>
BillerId	The identifier of the Billing Company	6 characters	Primary Key. All are digits.
Name	The name of the Billing Company	40 characters	
AccountId	The identifier of the account owned by the Billing Company	14 characters	All are digits (See AccountId in Table 2).

**Initial Version (worth 10%, Total mark is 20 and will be scaled down to 10)**

This is the initial version. The purpose of submitting this version is for you to start the project early and for the tutors to give you feedback. It includes the design of the system (which may be incomplete) and some prototypes to demonstrate some simple functions of the system. There is no need for the entire team to be in the demo session. However, it is expected that the person doing the demo can answer questions asked by the marker.

For Items 1 and 2 below, you need to submit the following both in hard and soft copies. You only need to submit soft copies of Item 3 below. The soft copies should be in either CD-ROM or DVD-ROM. Your submission should demonstrate the functionalities of F1 and F4 as described in the marksheet.

1. A design document related to F1 and F4 as minimal (In this document, you need to identify the roles and responsibilities of each component in your system, remember to include both 3-Tier and 4-Tier architectures in your system and put the right “module” in the right place. It is strongly recommended that you use diagram to illustrate your design.)
2. A simple instruction on running the related application clients with F1 and F4 (This is a set of simple steps for people to execute your software. Simple descriptions like those in the lab exercises will be sufficient. There is no need to capture screen dumps.)
3. The source code demonstrating that F1 and F4 could work (At this stage, you should be able to code using Servlet, JSP and/or JSF. However, you may still be unfamiliar with EJB coding. Hence, you are allowed to use some mock-up data to demonstrate that your system could work.)

**Marking Scheme of Initial Draft**

Category	Items	Marks
<b>Functionality</b>		
F1	Login	5
F4	Account Balance	5
	<b>Sub-Total</b>	<b>10</b>
<b>Design and Coding</b>	<b>(Must be completed related to F1 and F4) – If students include other functionalities in the design, please give them feedback and directions to proceed as well.</b>	
DC1	Overall architecture and use of object orientation	3
DC2	Database Design	1
DC3	Appropriate use of Servlets, JSPs and/or JSFs	1
DC4	Appropriate use of Enterprise Java Beans and Entity Classes <b>(Need to demonstrate some understanding in the design doc. No need to do the coding at this stage)</b>	1
DC5	Clear instructions	1
DC6	Code Quality – Must be simple and elegant <b>(Students may not be able to finish EJB related coding at this stage, some mock-up using “hard-coded” data will be fine. However, in the final version, EJB coding should be completed.)</b>	2
	<b>Sub-Total</b>	<b>9</b>
<b>Deployment</b>	<b>Sub-Total</b>	<b>1</b>
<b>Total</b>		<b>20</b>

## Final Submission (worth 30%, Total mark is 100 and will be scaled down to 30)

It is the complete system developed and submitted to the tutor for their marking. There is no need for the entire team to be in the demo session. However, it is expected that the person doing the demo can answer questions asked by the marker.

For Items 1 and 2 below, you need to submit the following both in hard and soft copies. You only need to submit soft copies of Item 3 below. The soft copies should be in either CD-ROM or DVD-ROM.

1. A design document (In this document, you need to identify the roles and responsibilities of each component in your system, remember to include both 3-Tier and 4-Tier architectures in your system and put the right “module” in the right place. It is strongly recommended that you use diagram to illustrate your design.)
2. A simple instruction on running the functions from the application clients (This is a set of simple steps for people to execute your software. Simple descriptions like those in the lab exercises will be sufficient. There is no need to capture screen dumps.)
3. The source code

## Marking Scheme of final submission

Category	Items	Marks
<b>Functionality</b>		
F1	Login	5
F2	Creating a Bank Account	5
F3	Removing a Bank Account	5
F4	Account Balance	5
F5	Transaction History	5
F6	Deposit	5
F7	Withdrawal	5
F8	Money Transfer	10
F9	Bill Payment	5
	<b>Sub-Total</b>	<b>50</b>
<b>Design and Coding</b>		
DC1	Overall architecture and use of object orientation	15
DC2	Database Design	5
DC3	Appropriate use of Servlets, JSPs and/or JSFs	5
DC4	Appropriate use of Enterprise Java Beans and Entity Classes	5
DC5	Clear instructions	5
DC6	Code Quality	10
	<b>Sub-Total</b>	<b>45</b>
<b>Deployment</b>	<b>Sub-Total</b>	<b>5</b>
<b>Total</b>		<b>100</b>

## Functionality

For the first part “Functionality”, you will be assessed based on your submitted applications. The user interface should be clear and unambiguous. Simple command line interface should be sufficient for the non-Web applications. **There is no extra point in building fancy GUIs using Swing or alike.**

**Design and Coding**

For DC1 – DC4 in the second part “Design and Coding”, you may 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. This will explain your design for the application in sufficient detail so that it can be used for implementation by another team.

1. Overall architecture
  - What the components of the application are
  - To what tier the components belong
  - How these components interact with each other
  - Java EE technologies utilised
  - Server-side interaction & integration
2. Database Design
  - Description of records and their structures (including the data types).
  - A database schema (probably using Network Diagrams)

**Note:** This document is very important for your understanding on the theory behind Java EE. It helps you prepare your examination.

For DC5 in the second part, you need to write instructions explaining how to run your client applications. For each function from F1 to F9, it is expected that the instructions should be at most 1 page. **NO screen dump is required.**

For DC6 in the second part, marks will be awarded on quality of implementation, including the user interface. Simple command line or text interface should be sufficient. **There is no point in building fancy GUI using Swing or similar technologies.**

**Deployment**

You need to demonstrate that your application functions properly in one machine. That is, the application clients (ATM, Teller, and the Web Browser) and the AppServer GlassFish are all in the same local machine.



**Swinburne University of Technology**  
**Faculty of Information and Communication Technologies**

**ASSIGNMENT COVER SHEET**

---

Subject code: HIT3119/8119                      Subject title: Enterprise Java  
Assignment number and title: \_\_\_\_\_ Due date: \_\_\_\_\_  
Assignment group number: \_\_\_\_\_ Lecturer: Dr. Edmonds Lau

---

Family name: \_\_\_\_\_ Identity no: \_\_\_\_\_

Other names: \_\_\_\_\_

***To be completed because this is a group assignment (at most 4 students in a group)***

We declare that this is a group assignment and that no part of this submission has been copied from any other student's work or from any other source except where due acknowledgment is made explicitly in the text, nor has any part been written for us by another person.

ID Number	Name	Signature
_____	_____	_____
_____	_____	_____
_____	_____	_____
_____	_____	_____

---

Marker's comments:

Total Mark: \_\_\_\_\_

---

**Extension certification:**

This assignment has been given an extension and is now due on \_\_\_\_\_

Signature of Subject Convener: \_\_\_\_\_