**COSC 2450 - Assignment 2**

**Web Development Technologies**

**COSC2450 ASSIGNMENT 2 SPECIFICATIONS**

**Submission Guideline**

 Assignment 2 is due **Saturday Week 11, 11:59PM** and **worth 35%** of the assessment. You must attempt this assignment individually.

 Each submission package must include

o Visual Studio.NET project files and all source files

o All SQL script necessary to recreate the database schema, tables schema, stored procedures and triggers used in the system and populate default data

**Test your script properly before submission** – it must be executed in the lecturer’s machine without any manual change. Marks will be deduced for scripts containing errors or requiring manual changes.

o A README.txt file documenting the followings

Your full name, your Student ID

Features that you have implemented including *known limitations* of those features

Instructions to deploy and execute the whole application.

o *While the readme.txt file won’t give you any mark, missing it or any of the required information may incur a mark deduction to up to 2 marks.*

 Refer to Submission Standard document (in Black Board) for more details about assignment submission rules for this course.

**Important Points**

 You are supposed to *demo your application* during the lab of **Week 12** or **Week 13**. Each student must book demo time at one of these 2 labs. Your presence to the lab demonstration is necessary for your assignment to be marked.

**Specification**

**1. Overview**

For this assignment, you will deal with the following topics: ASP.NET, ASP.NET AJAX, AJAX Control Toolkit, ADO.NET, SQL Server **2005** *(note the version, do not submit scripts for SQL Server 2008)*, stored procedures/LINQ-2-SQL, and triggers.

**2. Program Description**

You are to build a sub module in a bigger Enterprise Resource Planning System (ERP), a large enterprise system that links all departments in a company via a unique system, share and exchange data are done easily and quickly. Big and famous ERP like SAP or Oracle ERP can have more than 100 modules and be specialized for many different fields: banking, finance, production, commerce, pharmacy, and logistics. In Vietnam, local vendors also offer wide range of choices on ERP like LacViet, MISA though most of them are far to be compared with foreign products in terms of functionality, architecture, and scalability. In this assignment you are invited to develop one module of a whole ERP system. The assignment aims to sharpen your enterprise web application development and prepare you to be working for any ERP projects in your later career life as a software architect or technical consultant.

The ERP system you are making is a part of the whole ERP system (say RMIT ERP) which will be broken down to modules.

This non-functional requirement is for all the modules

**Non-functional Requirements (7 marks)**

**a) User-friendly UI (5 mark)**

The UI must be easy to use. Therefore, spend time thinking about the design of your application (e.g. workflow, screen navigation, web layout etc.) before actually implementing it. Note that this component is only concerned with overall application’s usability. Marks for individual function’s usability are already incorporated in the marking guideline in Functional Requirement section.

**b) Good OO design/code (2 marks)**

Adherence to the coding standards described in this document (

http://www.idesign.net/idesign/download/IDesign%20CSharp%20Coding%20Standard.zip), good OO design and exception handlings mechanism, use of generics collections.

**Technical Requirements**

**These technical requirements must be strictly adhered to**. Not adhering to these requirements may make you lose many marks for the relevant components.

a) You are recommended, but not required, to use ASP.NET Membership API, Role API and/or Profile API for the user management functionality. If you do this, you must use aspnet\_regsql to generate the schema and merge that with your database.

b) For database operations, you can choose between these 2 strategies: use *ADO.NET with stored procedures* or *LINQ-2-SQL*. You must **not** use dynamic or hardcoded SQL statements in your application.

c) You must use UpdatePanel and UpdateProgress controls for most modification operations in the site except for the popup. For example saving timesheet, approving timesheet etc. should be done via asynchronous postback enabled by UpdatePanel.

d) You must use validation controls to make sure that only the valid data is entered into the system. For example email address of a user must be valid.

e) You must use Calendar control of the AJAX Control Toolkit for any date time selection in your application.

f) You must use ASP.NET Master Page to organize reusable site elements and common styling.

**3. Module description:**

|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| **Build the Warehouse management module**  The warehouse management module is one of the crucial module any production and sales companies. The companies can make their own products or buy products from other partners and redistribute them. This module basically maintains the balance of stocks (quantity of products in a warehouse). It has a master table of product. Information of a **product** includes  - id (either entered by user or generated by a system following a certain  rules)  - name  - description  - unit (box, bottle)  - price  - current balance  A list of warehouses should also be managed. Information of a **warehouse** includes id, name, and address.  Importing of products to a warehouse is recorded in a **Warehouse entry slip – WES** (Import card). Information of a **WES** includes date, warehouse id, from what provider, total value, and user who creates the card.  A list of **providers** should be maintained. Information of a provider includes id, name, address, phone, email, contact person, phone and email of the contact person.  Each **WES** will have a list of details says **WESDetail**. Each **WESDetail** will has product id, quantity, unit, and price.  Similarly, there is **Delivery order slip - DOS** (ExportCard) to record the exporting of products. Information of a **DOS** is similar to **WES** such as date, time, from what warehouse to what warehouse, details of the DOS (product, quantity, price etc).   1. Build the GUI to support add/update/delete products, warehouse, WES and its details, DOS and its details 2. Based on the import and export transactions, the module shall allow the two main reports:   - List of transactions of a product in a warehouse; input parameters: from date, to date  **Product: A. From: 1/1/2012 To: 1/1/2013**   |  |  |  |  |  | | --- | --- | --- | --- | --- | | Transaction | Date | Quantity | Entry | Delivery | | 001 | 1/1/2012 | 5 | X |  | | 002 | 2/1/2012 | 20 |  | X | | 003 | 4/1/2013 | 10 |  | x |   - Balance report of all products in a warehouse; input parameters: from date, to date  **From: 1/1/2012 To: 1/1/2013**   |  |  |  |  |  | | --- | --- | --- | --- | --- | | Product | Begin | Entry | Delivery | End | | A | 1 | 2 | 3 | 0 | | B | 4 | 3 | 2 | 5 | | C | 3 | 2 | 4 | 1 |  1. Build a web services to get all products and their current balance in a particular warehouse. 2. Build a web services to get all warehouse transactions of a particular project in a particular warehouse within a period |