COMP4711 Assignment 1 (Winter 2013)
Due Date: Feb 3, 2013 23:59

Business Information System I

Background

We will be building modular simple business information systems this term, with each team of students building a module which will be a component of the "business" they are working on, with two businesses per set. The completed web applications will be built in parts, as assignments in this course.

This first assignment is to get the webapp structure and presentation in place, with really simple models and controllers. It will have a master file per module, with appropriate CRUD.

Student Teams

Each set will have two business sites to develop.

Each business will have a system administrator, responsible for managing the github code repository, ensuring consistency and creating unit tests.

Each business will have six modules, each built by a team of one or two students:

- general ledger (GL)
- accounts payable (AP) <suits one>
- accounts receivable (AR) <suits one>
- purchasing (PO)
- order entry (OE)
- inventory control (IC)

The purpose of the webapp is to model and process XML data suitably, not to create a serious competitor to any commercial accounting software.

Deployment

I have arranged for professional hosting of your work this term, using the domain bcitxml.com, hosted with hostpapa.com in Toronto. I am setting up subdomains for each business, and will give each team syssadmin an FTP account they can use to upload/deploy their webapps. Hostpapa does not support shell access.

I will be downloading a copy of the websites when the assignment is due, and marking from that.

Each subdomain will be based on the business name. For instance, team abc's site will be at abc.bcitxml.com.

Hostpapa Details

Hostpapa's PHP configuration is accessible at http://hp28.hostpapa.com/phpinfo.php

You can access your site using FTP...

Domain: bcitxml.com
Account name: abc@bcitxml.com
Password: XXXXXXXX (per the email I send you)

Development

I have arranged for github hosting of shared code for each “business”. I am setting up separate repositories for each business, and will give each team sysadmin the acocunt management details.

I may post updated common/shared components from time-to-time, for the repositories.

Business Context

Each of the six businesses will be a “retail” business, buying stuff from vendors, storing it in inventory, and selling it to customers.

Each business will have a three-letter acronym (abc, def, and so on) used to identify the business, its database and its data files folder.

The business model:

Module	Master file	Transactions	Data From	Data To
GL	Accounts	Ledger; Journal	All modules	N/A
AP	Vendors	Details	PO	GL
AR	Customers	Details	OE	GL
PO	Orders	Order details	IC	AP, GL
OE	Invoices	Invoice details	N/A	AR, GL
IC	Products	History	PO, OE	OE, GL

The master files will be stored in a MYSQL relational database.
The transactions will be stored in structured XML documents, one per master file record.

Each team is responsible for their master and transaction files – metadata, model, sample data, and business logic. The modules must work together, so the G/L module dictates the form and structure of transactions to post, for example.

Assignment 1 deals only with the master files, and with related CRUD. The transaction file models and CRUD will come in assignment 2, and the interoperability will come in assignment 3.

Data Requirements

Each of the master files have some shared data:

- identifying number (3 digit)
- name (string)
- status (char)

For now, the status code will be A for active or D for disabled.

The G/L account number ranges are 100-199 for assets, 200-298 for liabilities, 299 for equity, 300-399 for income and 400-499 for expenses.

The O/E and P/O master files also need a date field (string, formatted yyyy.mm.dd).

The products table needs some additional fields:

- quantity on hand
- reorder point
- reorder quantity
- preferred supplier account #

This is a rather simplistic view of the accounting/bookkeeping world, but it suits our purposes.

Assignment Tasks

1. Make a PHP webapp in NetBeans, structured similarly to what we have done in labs 2+. I will expect to see the `nbproject` folder inside, so I can conveniently open up each project using NetBeans.

2. Business and team assignments will be chosen during lab. Your project name (which reduces to the acronym) is up to you, as is coordination of the team members work on and updating of content. The sysadmins are the only ones posting updates to the hosted site.

3. The website should have a common layout throughout (again, similar in principle to labs 2+). The homepage should provide a way to reach any of the modules.

4. You are to use CodeIgniter models, views and controllers, with a layout and templating structure similar to what we have used in lab.

5. The home page should clearly identify the business acronym. It can list the student teams as well, or they can be on an "about" page.

6. Apart from the basic site structure (pages required), you may be innovative and have a bit of fun with the content.

7. Each module team should create SQL scripts to create their master file and add a dozen records to it.

8. Pages required:

- **welcome** home page, per the above

- **gl/welcome** homepage for the G/L, listing all the accounts; sub-controllers to view, add or update

- **ap/welcome** homepage for the A/P, listing all the vendors; sub-controllers to view, add or update

- **ar/welcome** homepage for the A/R, listing all the customers; sub-controllers to view, add or update

- **po/welcome** homepage for the P/O, listing all the purchase orders; sub-controllers to view, add or update

- **ic/welcome** homepage for the I/C, listing all the products; sub-controllers to view, add or update

- **oe/welcome** homepage for the O/E, listing all the orders; sub-controllers to view, add or update

9. Suggested folder structure: as for lab 3.

```
index.php
.htaccess
application/
assets/
```

10. I suggest separate folders inside the controllers and views folders, for each module, though the webapp is simple enough that we can get away with everything directly inside each of those.

11. Non-RDB data will be kept in an external folder, /data/xxx. Specifically, there should be a properties.xml file, with provision for module properties (gl.xxxxxx) as well as business properties (xxxxxx)

12. Your site does not have to be exceptionally aesthetically beautiful, but then again we do not want it exceptionally ugly. There will be a subjective component of the grade for aesthetics.

13. Provide a `readme.txt` in your project root, with anything I need to know about your assignment before reviewing it. This would be a good place to note which of the requirements were fully met, partially met, or not met, in your opinion.

Submission

Nothing has to be handed in to share-in. I will download the entire hostpapa site in order to view your source code.

Marking Guideline

The assignment requirements are reflected in the marking breakdown (out of 20) below:

Component	Marks
Controllers	2
Models	2
Views	2
Aesthetics & consistency	2
Data completeness & validity	2
Works properly	3
Useful summary / readme	1
Comments & programming style	2

Teams

Team members do not have to get the same grade if they do not contribute equally. You need to advise me of such circumstances. If a team ends up really fractured, then it will have to split in two, and one of the “groups” will be assigned a different module to work on.

Plan on working with the same team members for all three assignments.