Software Architecture (COM3041-N)
Assignment 1

Details

Title:	*Davison Store*
Assessment Type:	Summative (70%)
Submission Date:	Monday 2nd March 2015
Submission Method:	*Blackboard*
Feedback Method:	You will receive group electronic feedback on your work. This feedback will include comments that are provided to help you improve. You will also receive individual marks. These will not be final until they are combined with the marks from the other assignments and then approved by the assessment board at the end of the academic year.
Feedback Date:	Monday 30th March 2015

Introduction

This is assessment component 1 of 2 for this module. More details on the entire assessment can be found in the module guide.

This is a group-based assignment. You will be assigned a development team composed of members from within your tutorial sessions. Teams will ideally consist of four students.

Within your team you will design, construction and verify an enterprise-standard, distributed, multi-component software application in a professional manner (tools, techniques and practices). The precise details and requirements are given in the remainder of this document.

Although this is a group assignment you will still receive individual marks. Your individual mark will be generated from the marks for the group's work (as detailed in the criteria section) via a peer assessment and tutor moderation process (details to follow).

Brief

Your team work as software developers for fictitious *Davison Corporation*. Our company aims to:

- Find and sell the cheapest products to our customers from a variety of online suppliers.
- Demonstrate how much we have saved our customers by showing them price comparisons.
- Provide a social online shopping experience for our customers, as a minimum allowing them to leave comments and ratings about the products available.
- Provide a price reduction incentive to our customers for purchasing multiple types of products within a single transaction.
- Stock and sell our own range of products.
- Exploit bulk purchasing cost savings where possible.

We want customers to be able to:

- Browse all products with suitable filtering and searching to improve the experience.
- Purchase one or more types of product, specifying a quantity for each, within a single guaranteed transaction regardless of how many online suppliers are involved.
- Add social value (e.g. comments, ratings) to the products; these relate to the products themselves, not to the product from a particular supplier.

Your team is responsible for designing and constructing a new system (suite of software components) to enable our company to meet its aims and deliver its customer experience. The system should be fully functional and ideally be well engineered. You are required to utilise only tutor-approved technologies. Also devise a procedure for deploying the system into use.

Your solution must include:

- A customer web app capable of browsing and purchasing products following the above guidelines.
- A customer Windows 8 Store app (either JavaScript/HTML or C#/XAML) with at least the same functionality as the above web app.
- Sourcing of products from the tutor-provided online web services.
- Sourcing of products from the tutor-provided Davison database without modification to its schema.
- Restocking of Davison products via the tutor-provided legacy software component.
- Suitable security to authenticate and authorise users.
- Resilient handling of multiple concurrent users.
- Resilient handling of failure of one or more software components.

You are expected to demonstrate that:

- **The above function requirements have been met.**
- **Suitable software development frameworks and tools have been considered and selected.**
- **Suitable software and architecture patterns have been considered and selected.**
- **Software components have been constructed and thoroughly unit tested.**
- **A procedure has been devised to deploy the system into a production environment.**

Guidance

Use an agile / iterative development method. Pick a particular method (e.g. SCRUM) and stick to it. Participate in the meetings, do the work, and above all be professional and courteous to each other.

Utilise the tutors as *product owners* to give external drive and direction to the project. This will help avoid in-fighting about the "vision" of the product and let you focus on building it.

Use source control (e.g. SVN or GIT) and other appropriate development tools (e.g. bug trackers). Use source control formally (i.e. branch, merge, and tag) as appropriate to your dev method.

Start the assignment's required documentation early and develop it iteratively. Use it productively to structure the work and communicate with each other, rather than as something to develop just for marking purposes at the end.

Accept that you're a team of people with different skillsets. Play to each other's strengths by structuring and developing the work sympathetically to those skillsets.

Avoid working on overlapping areas of the system. Agree the interfaces between components early and don't change them without full consultation with those involved. Change is inevitable, be do it in a controlled manner.

Deliverables

You must submit your work as a ZIP file via *Blackboard* with the following directory structure:

- ***Demo***: this directory should contain a working release build of the system. All the required data files and instructions of how to configure the system should be included. Also remember to include any copyright notices necessary for third-party frameworks, tools, libraries, etc. This directory is used to demo your work to others.
- ***Media***: this directory should contain full resolution screen shots and short (30-60 seconds) full resolution movie captures of the application. This directory is used for marketing your work.
- ***Source***: this directory should contain the entire source code for the system. Ensure that you have *cleaned* (i.e. deleted intermediate and non-essential files) the project by deleting the *release* and *debug* directories, empting the *bin* directories, and deleting all of the *packages* sub-folders. This directory is used for assessing your coding.
- ***Documentation***: this directory should contain any documents created to demonstrate and evidence that your work meets the requirements of the marking criteria (e.g. architecture diagrams, reports). This directory is used for assessing your architecture.

Include a *readme* file in the root directory containing the names and user ids of each member of your team. Also include the name of an SCM lab machine that you have successfully tested your work on.

Include the *peer assessment* files in the root directory containing your weighting of each other's contribution to the assignment (details to follow).

Marking Criteria

Functional requirements	20%	Can products be browsed? Can products be purchased? Are business rules being supported? (e.g. costing model) Has added value being demonstrated? (e.g. comments and ratings) Is more than one client app demonstrated? Has legacy component integration being demonstrated? (e.g. re-stocking) Is the system responsive under failure? Is the system secure? Are concurrent users attempted?
Choice and effectiveness of the software development frameworks and tools	25%	Are they appropriate? Has it led to SOLID coding? Was it an informed choice? Are failings and difficulties with them understood?
Suitability of the software architecture and the correctness of its implementation	25%	Has the architecture been documented? Is the documentation in agreement with the implementation? Does the architecture clearly show layering and separation of responsibilities? Is the architecture an informed choice? Are the failings and difficulties with architecture understood?
Thoroughness of the verification testing performed	20%	What proportion of the software components are unit tested? Are there unit tests for all functionality? Have mocks being used to isolate testing? Has failure been tested? Has concurrency been tested?
Quality of the devised deployment method	10%	Is the build process described? Is the deployment sequence described? Are the pre-requisites for deployment identified? Has a roll-back method been given? Has a verification method been given? Have the necessary tools been explained?

Document History

This is the initial version of the 2014 assignment.