EX-SELL

Design for Software Maintainability

Version 1.0 approved

Prepared by Zenith

Yiu Hong Sum	U1621435G
Lim Kian Hock Bryan	U1620949L
Cheng Yang Zhen	U1521618K
Lim Yan Jun	U1622311B
Soong Jie Ming	U1521535B
Chen Zhenni	U1622603D
Chang Keat Lueng Aaron	U1622360F

29th March 2018

Revision History

Revision	Description of Change	Approved by	Date
0.10	Design Strategies	Yang Zhen	21/03/2018
0.40	Architectural Design	Jie Ming	24/03/2018
0.60	Software Configuration Management Tools	Aaron	25/03/2018
1.00	Compilation	Bryan	29/03/2018

Table of Content

1. Design Strategies

1.1. Planning

During each meeting, our group will discuss the most important features which we should include in the web application to enhance the user experience for each student using Ex-Sell. We will then analyze and predict the improvements after we release the application.

Thus, we will mainly be targeting Usability of Ex-Sell for the NTU community, as our web application foresees a broad user base. Thus, software must flow according to the user's expectation, offer needed information, and provide navigation controls which are clear and obvious.

Scalability of our system is also needed as we continually expand to test our web applications from a smaller to bigger user base. Security is also essential that web software applications handle customer data and other electronic information as securely as possible, and thus Security measures are vital as well.

To cope with these, our team has decided to adopt the 3 Tier Architecture design pattern.

1.2. Development Process

For the initial runs, we will be rolling out our web application in in a small and test-driven environment. Firstly, we will start with a small cohort of people (e.g. SCSE students) to test their responses when using Ex-Sell. After a set period, we will then collate their feedback based on how we can improve the design and usability of our application. Thereafter, we will expand to the entire college, collating feedback once again to enhance the application before rolling it out as the official web application.

1.3. Correction by Nature

For the official testing of Ex-Sell, we will be implementing the following procedures to look out for, namely:

1.3.1. Corrective Maintainability

Reactive modification of a software product performed after delivery to correct discovered problems

1.3.2. Preventive Maintainability

Modification of a software product after delivery to detect and correct latent faults in the software product before they become effective faults.

1.4. Enhancement by Nature

We will enhance our application while testing the application. And this is what we will look out for:

1.4.1. Perfective Software Maintenance

After our web application is up and running, we are able to quickly detect errors and correct it when needed, which helps to reducing maintenance costs and time wasted.

2. Architectural Design - 3 Tier Architecture

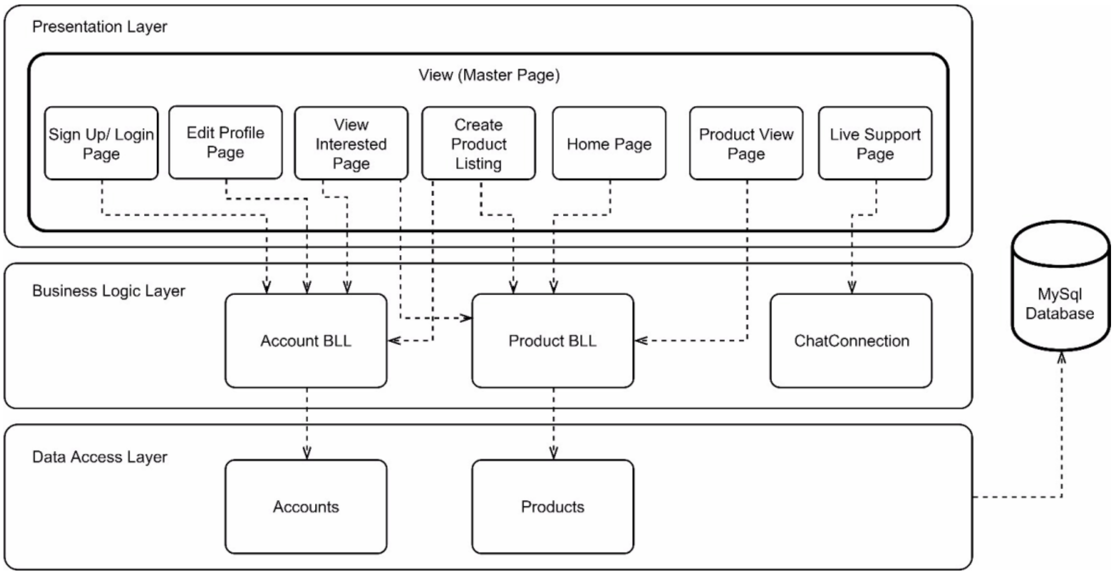
Ex-Sell is using the 3 tier Architecture pattern, which consists of the Presentation Tier, Business Logic Tier and the Data Access Tier respectively. We have chosen the following architecture as it allows for the logical separation between all three layers.

Presentation Layer: It contains .aspx pages where data is presented to the user or input is taken from the user. In this case the ASP.NET web site is called the Presentation Layer.

Business Logic Layer: Contains business logic, validations or calculations related to the data. The Business Layer is vital in that it validates the input conditions before calling a method from the data layer.

Data Access Layer: It contains methods that helps the Business Layer to connect the data and perform required actions, whether to return data or to manipulate data (insert, update, delete) on the sql server.

Below shows the following design for our architecture:



3. Software Configuration Management Tools

Having the right Software Configuration Management Tools(SCMT) allows us as a group to do the following:

- Traceability between software and test ware and different versions or variants.
- To build and release management.
- Base lining (e.g. all the configuration items that make up a specific release).
- Access control (checking in and out).

Below is a list of SCMT that we have utilized for this project:

3.1. MediaWiki

MediaWiki is a free and open-source application. This service is used as it is easy for beginners to pick up. There are many FAQs provided which can teach users the functions required by the users. There is a wide range of functions which allows users to create their information in different styles. It also allows users to concurrently edit the page at the same time. Hence, editing of the page will not result in a loss of information.

3.2. TortoiseSVN

SVN is used for versioning and for revision control. This service is used as it helps us to keep track of the different versions of source code files and documents being uploaded and edited. It allows users to rename, move, remove, and copy the file while keeping the full revision history. TortoiseSVN (the svn that we are using) can be used to integrate windows explorer and hence, allowing us to view and edit the files easily.

3.3. Google Drive

Google Drive service is used as a file storage and for the backup of documents initially created. This service allows users to share and store files within the group easily. Also, this service allows users to edit documents concurrently. Version control is also available in this service.

3.4. GitHub

Github is used as a repository to track all the changes in code whenever there is a Sprint meeting. This allows us to push and pull the relevant code from the repository easily while keeping track of past code changes.