|  |  |  |  |
| --- | --- | --- | --- |
| **Project Name** | Gift Portal | **Developer** | Lakshmi Annaparthi/Boga Swetha |
| **Review Date** |  | **Project Manager** | Rachana Brahmbhatt |
| **Reviewer** |  |  |  |

# Introduction

This checklist (template) is provided for .NET developers and architects to review .NET application code. The intent of the checklist is to define the series of steps taken to perform a code review. Many of the specific details are provided in the references. Not all items listed below are applicable to a given review. The application’s components and type of data being processed typically which of the following sections apply.

The checklist is organized by these topic areas:

|  |  |
| --- | --- |
| * Section 3 - .NET General | * Section 8 - Web Services |
| * Section 4 - Persistence | * Section 9 - Architectural Stategy |
| * Section 5 - ASP.NET General | * Section 10 - Code Review Tools |
| * Section 6 – ASP.NET WebForms * Section 7 – ASP.NET MVC | * Section 11 - PCI Analysis * Section 12 - File Details * Section 13 - Follow-up Items |

**Severity Level Key**

Each audit category is scored in accordance with a severity level as defined in the table below. In general, S1 defects are material threats to the hosting environment and/or violate fundamental best practices or Payment Card Industry (PCI) standards, S2 defects relate to performance and support risks, S3 defects point out areas which affect the long term maintainability or troubleshooting.

|  |  |
| --- | --- |
| **Level** | **Description** |
| **S1** | **Critical.** Issues must be fixed prior to productions. Issues in this category pose a high risk to hosting servers due to stability or performance problems. Items may include flawed architecture, missing fundamental best practices, inability of the application to handle the expected load, not adhering to PCI standards. |
| **S2** | **High.** The solution **might** be able to proceed to production under exception rules, but fixing these issues prior is highly recommended. Problems pose moderate risk to the application and hosting servers. Without remediation, the application is likely to experience periodic performance and stability issues, difficulty with maintenance and very high support costs. |
| **S3** | **Medium.** Solution may proceed to production. It is recommended that issues be addressed during the development cycle. Risk to solution is moderate; risk to hosting servers is low. Items in this category violate lower level best practices or guidelines related to optimization or maintainability. Without remediation, the solution may suffer from performance issues and higher than normal support costs. |
| **N/I** | **No Issues Found.** |
| **N/A** | **Not Applicable.** No action required. |

# References

[OWASP](http://www.owasp.org/index.php/Top_10_2007) - The Open Web Application Security Project (OWASP) is a worldwide free and open community focused on improving the security of application software. Also, here’s a link to a good document on [Top 10 Rules for Secure Coding](file:///C:/COE-MS/MSArchitecture/AB%20Document%20Library/1/Top%2010%20Rules%20for%20Secure%20Coding.doc).

# Modified Source Listing

List the change sets or source files that were modified in this section.

# .NET General

Insert the appropriate severity level in the level cell. Add comments to explain the nature of the issue.

| **Id** | **Level** | **Review Item** | **Comments** |
| --- | --- | --- | --- |
|  | **N/I** | Application should adhere to the project’s architecture and design specifications. | Yes. Verified. |
|  | **N/I** | Application should adhere to [Microsoft’s Design Guidelines](http://msdn.microsoft.com/en-us/library/ms229042.aspx). | Yes. Verified. |
|  | **N/I** | A unit test should be available for every public method. | Yes. Unit tests are created |
|  | **N/I** | Are there instances of duplicated code? “Copy and Paste”. Duplicated code should be re-factored into a new method or class. | No duplicated code is present. |
|  | **N/I** | Search source files for use of the “new” keyword. Review to make sure these items are being disposed when finished. | Yes. Verified. |
|  | **N/I** | Search source files for hard coded strings. Review to insure these are appropriate (no connection strings, etc). | Yes. Verified. Connection String is present in Config file only. |
|  | **N/I** | Manual inspection for proper error handling. Refer to [Microsoft’s Exception Handling Guidelines](http://msdn.microsoft.com/en-us/library/ms229005.aspx). | Yes. Used HandleException of Vault instead of just throw |
|  | **N/I** | If application has strict performance constraints, the tips found at [.NET Performance Tips](http://msdn.microsoft.com/en-us/library/ms973839.aspx) have been utilized. | Not applicable |
|  | **N/I** | Look for segregation of database dependencies from UI code. Most often caused by binding a DataSet directly to a UI page – this can be OK for small applications, but prefer better isolation on large applications. This really should be on the architecture review checklist, but it is easier to spot when you have the code. | Data sets are not used |
|  | **N/I** | Review implementation of any system interfaces (databases, web services, etc). Verify proper authorization / authentication and error handling is in place. | Verified. |
|  | **N/I** | Check for classes or methods that are never used in the application? Dead code should be removed to increase the maintainability and supportability of the application. | Removed unnecessary code |

# Persistence

| **Id** | **Level** | **Review Item** | **Comments** |
| --- | --- | --- | --- |
|  | **N/I** | If Entity Framework is being used, review use of Deferred Loading and ensure unnecessary database communication is minimal. | Not applicable |
|  | **N/I** | Prefer stored procedures over embedded SQL. Stored procedures are more secure and usually provide better performance. | Yes. Verified. No SQL statements are used. |
|  | **N/I** | Query plans should be documented for SQL statements. | Not applicable |
|  | **N/I** | Use parameters for your database queries when working with Oracle or SQL\*Server. | Not applicable |
|  | **N/I** | Be wary of DataSets due to their complex internal object hierarchy. Try to avoid serialization of DataSets. | Data sets are not used. |
|  | **N/I** | Objects implementing dispose() should have the dispose() method called as soon as possible. This could have been listed under the General topic, but most often this problem occurs with ADO.NET objects (DataSet, DataTable, DataView, Command). | Verified. |

# ASP.NET General

| **Id** | **Level** | **Review Item** | **Comments** |
| --- | --- | --- | --- |
|  | **N/I** | If using ASP.NET session, review for adherence to [ASP.NET State Management Recommendations](http://msdn.microsoft.com/en-us/library/z1hkazw7.aspx) | Coded as per standards |
|  | **N/I** | Review settings in the web.config file. Ensure that the settings will not conflict with the machine.config settings for the application's destination test and production environments. Be sure no sensitive data is stored in the file. | Yes. No sensitive data is present in the files. |
|  | **N/I** | Review use of the methods in the Global.asax code behind class, especially Application\_Start(), Session\_Start() and Application\_Error(). | Yes. Verified. |
|  | **N/I** | Look for opportunities to use various ASP.NET caching options (server, page, etc.) | Not applicable |
|  | **N/I** | Review security requirements for ASP.NET Web Services (Basic Auth, SSL, etc.) | Not applicable |

# ASP.NET Web Forms

| **Id** | **Level** | **Review Item** | **Comments** |
| --- | --- | --- | --- |
|  | **N/I** | Review use of viewstate. Only store the minimum number of objects in view state. Disable viewstate if you don’t need it. | Not applicable |
|  | **N/I** | Be wary of using DataGrid controls. This control can generate a lot of viewstate. Consider using a DataList or Repeater control as a lighter weight alternative | Not applicable |
|  | **N/I** | Use the Application object to share static, read-only data. | Not applicable |

# ASP.NET MVC

| **Id** | **Level** | **Review Item** | **Comments** |
| --- | --- | --- | --- |
|  | **N/I** | All business logic should be called through actions to the controllers | Not applicable |
|  | **N/I** | Data should be bound to views by use of the model | Not applicable |
|  | **N/I** | Use the Application object to share static, read-only data. This should also be avoided in MVC. | Not applicable |

# Web Services

| **Id** | **Level** | **Review Item** | **Comments** |
| --- | --- | --- | --- |
|  | **N/I** | Services should have minimal functionality in them and should use a controller class to handle most of the work | Not applicable |
|  | **N/I** | Namespaces should be assigned to all Service and Data Contracts | Not applicable |
|  | **N/I** | Soft titling should be assigned to Service, Data, and Operation Contracts | Not applicable |
|  | **N/I** | Complex Types should be used for input and output parameters. | Not applicable |

# Architectural Strategy Adherence

| **Id** | **Level** | **Review Item** | **Comments** |
| --- | --- | --- | --- |
|  | **N/I** | Logging adheres to the logging strategy defined in the architecture specifications. | Logging done as per Vault standards |
|  | **N/I** | Exception handling adheres to the exception handling strategy defined in the architecture specifications. | Exception Handling done as per Vault standards |
|  | **N/I** | Integrations adhere to the strategy defined in the architecture specifications | Verified |
|  | **N/I** | Integrations points are reusable as service | Not applicable |
|  | **N/I** | If caching is being utilized, it adheres to the caching strategy defined in the architecture specifications. | Not applicable |
|  | **N/I** | Security adheres to the architectural specifications. | Not applicable |
|  | **N/I** | Review use of third party and open source tools: - if any tools have been used that were not specified in the architecture, they must be reviewed for use at Maritz following the [open source policies and review](http://mgts.maritz.com/development/MLM/wikit/Wiki%20Pages/ARCH_OPENSOURCE_POLICIES_AND_REVIEW.aspx) document. | Not applicable |

# Code Review Tools

| **Id** | **Level** | **Review Item** | **Comments** |
| --- | --- | --- | --- |
|  | **N/I** | Code Analysis from Visual Studio has been collected and all critical, high, and medium alerts have been remediated. | In progress |

# PCI Analysis

PCI Requirement #6 states “Develop and maintain secure systems and application.” For custom developed applications, numerous vulnerabilities can be avoided by using standard system development processes and secure coding techniques. Below is a list of common coding vulnerabilities present in web based applications (taken from the [OWASP Top 10 2007](http://www.owasp.org/index.php/Top_10_2007)). Appendix A lists the details associated with this PCI Analysis section.

| **Id** | **Level** | **Review Item** | **Comments** |
| --- | --- | --- | --- |
|  | **N/I** | A1 – Cross Site Scripting (XSS)  XSS flaws occur whenever an application takes user supplied data and sends it to a web browser without first validating or encoding that content. XSS allows attackers to execute script in the victim's browser which can hijack user sessions, deface web sites, possibly introduce worms, etc. | Verified and used Server.HTMLEncode for textbox inputs |
|  | **N/I** | A2 – Injection Flaws  Injection flaws, particularly SQL injection, are common in web applications. Injection occurs when user-supplied data is sent to an interpreter as part of a command or query. The attacker's hostile data tricks the interpreter into executing unintended commands or changing data. | Not applicable |
|  | **N/I** | A3 – Malicious File Execution  Code vulnerable to remote file inclusion (RFI) allows attackers to include hostile code and data, resulting in devastating attacks, such as total server compromise. Malicious file execution attacks affect PHP, XML and any framework which accepts filenames or files from users. | Not applicable |
|  | **N/I** | A4 – Insecure Direct Object Reference  A direct object reference occurs when a developer exposes a reference to an internal implementation object, such as a file, directory, database record, or key, as a URL or form parameter. Attackers can manipulate those references to access other objects without authorization. | Not applicable |
|  | **N/I** | A5 – Cross Site Request Forgery (CSRF)  A CSRF attack forces a logged-on victim's browser to send a pre-authenticated request to a vulnerable web application, which then forces the victim's browser to perform a hostile action to the benefit of the attacker. CSRF can be as powerful as the web application that it attacks. | Not applicable |
|  | **N/I** | A6 – Information Leakage and Improper Error Handling  Applications can unintentionally leak information about their configuration, internal workings, or violate privacy through a variety of application problems. Attackers use this weakness to steal sensitive data, or conduct more serious attacks. | Error Handling done as per Vault standards |
|  | **N/I** | A7 – Broken Authentication and Session Management  Account credentials and session tokens are often not properly protected. Attackers compromise passwords, keys, or authentication tokens to assume other users' identities. | Not applicable |
|  | **N/I** | A8 – Insecure Cryptographic Storage  Web applications rarely use cryptographic functions properly to protect data and credentials. Attackers use weakly protected data to conduct identity theft and other crimes, such as credit card fraud. | Not applicable |
|  | **N/I** | A9 – Insecure Communications  Applications frequently fail to encrypt network traffic when it is necessary to protect sensitive communications. | Not applicable |
|  | **N/I** | A10 – Failure to Restrict URL Access  Frequently, an application only protects sensitive functionality by preventing the display of links or URLs to unauthorized users. Attackers can use this weakness to access and perform unauthorized operations by accessing those URLs directly. | Not applicable |

# File Details

The following comments by file are intended to help identify patterns of coding issues and help with specific problems. It is not necessarily comprehensive coverage of all files, but should contain detailed comments about representative issues.

**General Notes** –

* Below is the list of modified objects as a part of Gift Portal project

**DDL and Data:**

$/MLM\_Suntrust/SunTrust/Database/Dev/4.0.3/Database/DDL/SunTrust\_GiftPortal\_DDL.sql

$/MLM\_Suntrust/SunTrust/Database/Dev/4.0.3/Database/DDL/SunTrust\_GiftPortal\_DDL\_Rollback.sql

$/MLM\_Suntrust/SunTrust/Database/Dev/4.0.3/Database/Data Scripts/SunTrust\_GiftPortal\_Data.sql

$/MLM\_Suntrust/SunTrust/Database/Dev/4.0.3/Database/Data Scripts/SunTrust\_GiftPortal\_Data\_Rollback.sql

**Stored Procedures:**

$/MLM\_Suntrust/SunTrust/Database/Dev/4.0.3/Database/Sprocs/Vault.SaveNewGiftCode.sql

$/MLM\_Suntrust/SunTrust/Database/Dev/4.0.3/Database/Sprocs/Vault.SaveExistingGiftCode.sql

$/MLM\_Suntrust/SunTrust/Database/Dev/4.0.3/Database/Sprocs/Vault.DeleteGiftCode.sql

$/MLM\_Suntrust/SunTrust/Database/Dev/4.0.3/Database/Sprocs/Vault.GetCustomerGiftCodeByGiftCode.sql

$/MLM\_Suntrust/SunTrust/Database/Dev/4.0.3/Database/Sprocs/Vault.GetCustomerGiftCodeListByGiftCode.sql

$/MLM\_Suntrust/SunTrust/Database/Dev/4.0.3/Database/Sprocs/Vault.GetCustomerGiftCodeListByExternalAccount.sql

$/MLM\_Suntrust/SunTrust/Database/Dev/4.0.3/Database/Sprocs/Vault.GetCustomerGiftCodesAll.sql

$/MLM\_Suntrust/SunTrust/Database/Dev/4.0.3/Database/Sprocs/Vault.GetCustomerByGiftCode.sql

$/MLM\_Suntrust/SunTrust/Database/Dev/4.0.3/Database/Sprocs/Vault.GetAwardGiftCodeListBySegmentValueCode.sql

$/MLM\_Suntrust/SunTrust/Database/Dev/4.0.3/Database/Sprocs/Vault.GetCustomerGiftCodeListByLastFirstPostal.sql

$/MLM\_Suntrust/SunTrust/Database/Dev/4.0.3/Database/Sprocs/Vault.GetCustomerGiftCodeListByPhoneNumber.sql

$/MLM\_Suntrust/SunTrust/Database/Dev/4.0.3/Database/Sprocs/Vault.Unload\_ZustekHourlyEvent5.sql

$/MLM\_Suntrust/SunTrust/Database/Dev/4.0.3/Database/Sprocs/Vault.Unload\_ZustekHourlyEvent6.sql

$/MLM\_Suntrust/SunTrust/Database/Dev/4.0.3/Database/Sprocs/Vault.Batch\_EventDrivenEMailCampaignUnload.sql

Batch processing:

$/MLM\_Suntrust/SunTrust/Database/Dev/4.0.3/Database/Sprocs/Vault.GetCustomerGiftCodeByGiftCode.sql

$/MLM\_Suntrust/SunTrust/Database/Dev/4.0.3/Database/Sprocs/Vault.SaveExistingGiftCode.sql

$/MLM\_Suntrust/SunTrust/Database/Dev/4.0.3/Database/Sprocs/Vault.GetCustomerGiftCodeListByGiftCode.sql

$/MLM\_Suntrust/SunTrust/Database/Dev/4.0.3/Database/Sprocs/Vault.LoginGiftCodeUser.sql

$/MLM\_Suntrust/SunTrust/Database/Dev/4.0.3/Database/Sprocs/Vault.LoginGiftCodeUserGetCustomerInfo.sql

$/MLM\_Suntrust/SunTrust/Database/Dev/4.0.3/Database/Sprocs/Vault.FetchAccountSegmentHistoryTAIntervalDates.sql

$/MLM\_Suntrust/SunTrust/Database/Dev/4.0.3/Database/Sprocs/Vault.Batch\_GiftPortalCCGUnload\_Job.sql

$/MLM\_Suntrust/SunTrust/Database/Dev/4.0.3/Database/Sprocs/CmpMgr.GenerateOfferCodesForGiftPortal.sql

$/MLM\_Suntrust/SunTrust/Database/Dev/4.0.3/Database/Sprocs/CmpMgr.GenerateOneGiftPortalSegmentTA.sql

$/MLM\_Suntrust/SunTrust/Database/Dev/4.0.3/Database/Sprocs/CmpMgr.QualificationAllTargetQualifyGP.sql

$/MLM\_Suntrust/SunTrust/Database/Dev/4.0.3/Database/Sprocs/Vault.Batch\_ProcessGiftPortalEnrollment\_RunDate.sql

$/MLM\_Suntrust/SunTrust/Database/Dev/4.0.3/Database/Sprocs/Vault.Batch\_Select\_GiftPortalRedeemCodeFile.sql

$/MLM\_Suntrust/SunTrust/Database/Dev/4.0.3/Database/Sprocs/Vault.Batch\_Unload\_GiftPortalRedeemCodeFile.sql

**Triggers:**

$/MLM\_Suntrust/SunTrust/Database/Dev/4.0.3/Database/Triggers/Vault.trgManageAccountSegmentDeletes.sql

$/MLM\_Suntrust/SunTrust/Database/Dev/4.0.3/Database/Triggers/Vault.trgManageAccountSegments.sql

**Business Objects:**

$/MLM\_Suntrust/SunTrust/Web/Dev/4.0.3/Business/Membership/Customer.cs

$/MLM\_Suntrust/SunTrust/Web/Dev/4.0.3/Business/Membership/Customers.cs

$/MLM\_Suntrust/SunTrust/Web/Dev/4.0.3/Business/Membership/Factory.cs

$/MLM\_Suntrust/SunTrust/Web/Dev/4.0.3/Business/Membership/GiftCode.cs

$/MLM\_Suntrust/SunTrust/Web/Dev/4.0.3/Business/Membership/GiftCodes.cs

$/MLM\_Suntrust/SunTrust/Web/Dev/4.0.3/Business/TOOLKIT/EnumHelper.cs

$/MLM\_Suntrust/SunTrust/Web/Dev/4.0.3/Business/Commerce/Catalog/Rewards.cs

$/MLM\_Suntrust/SunTrust/Web/Dev/4.0.3/Business/Commerce/Order.cs

$/MLM\_Suntrust/SunTrust/Web/Dev/4.0.3/Business/Membership/Account.cs

$/MLM\_Suntrust/SunTrust/Web/Dev/4.0.3/Business/Membership/Customer.cs

$/MLM\_Suntrust/SunTrust/Web/Dev/4.0.3/Business/Membership/Customers.cs

$/MLM\_Suntrust/SunTrust/Web/Dev/4.0.3/Business/Membership/GiftCode.cs

**Call Center:**

$/MLM\_Suntrust/SunTrust/Web/Dev/4.0.3/Inetpub/wwwroot/SunTrustCallCenter/Javascript/FormValidation.js

$/MLM\_Suntrust/SunTrust/Web/Dev/4.0.3/Inetpub/wwwroot/SunTrustCallCenter/UserControls/DeclineGift.ascx

$/MLM\_Suntrust/SunTrust/Web/Dev/4.0.3/Inetpub/wwwroot/SunTrustCallCenter/UserControls/DeclineGift.ascx.cs

$/MLM\_Suntrust/SunTrust/Web/Dev/4.0.3/Inetpub/wwwroot/SunTrustCallCenter/UserControls/SideMenu.ascx.cs

$/MLM\_Suntrust/SunTrust/Web/Dev/4.0.3/Inetpub/wwwroot/SunTrustCallCenter/UserControls/TopBar.ascx

$/MLM\_Suntrust/SunTrust/Web/Dev/4.0.3/Inetpub/wwwroot/SunTrustCallCenter/UserControls/TopBar.ascx.cs

$/MLM\_Suntrust/SunTrust/Web/Dev/4.0.3/Inetpub/wwwroot/SunTrustCallCenter/basePage.cs

$/MLM\_Suntrust/SunTrust/Web/Dev/4.0.3/Inetpub/wwwroot/SunTrustCallCenter/CommonUtility.cs

$/MLM\_Suntrust/SunTrust/Web/Dev/4.0.3/Inetpub/wwwroot/SunTrustCallCenter/Global.asax.cs

$/MLM\_Suntrust/SunTrust/Web/Dev/4.0.3/Inetpub/wwwroot/SunTrustCallCenter/Search.aspx

$/MLM\_Suntrust/SunTrust/Web/Dev/4.0.3/Inetpub/wwwroot/SunTrustCallCenter/Search.aspx.cs

$/MLM\_Suntrust/SunTrust/Web/Dev/4.0.3/Inetpub/wwwroot/SunTrustCallCenter/BrowseRewards.aspx.cs

$/MLM\_Suntrust/SunTrust/Web/Dev/4.0.3/Inetpub/wwwroot/SunTrustCallCenter/CheckOut.aspx

$/MLM\_Suntrust/SunTrust/Web/Dev/4.0.3/Inetpub/wwwroot/SunTrustCallCenter/CheckOut.aspx.cs

$/MLM\_Suntrust/SunTrust/Web/Dev/4.0.3/Inetpub/wwwroot/SunTrustCallCenter/ViewCart.aspx.cs

**Gift Portal:**

All objects under the project $/MLM\_Suntrust/SunTrust/Web/Dev/4.0.3/Inetpub/wwwroot/SunTrustGiftPortal

**Unit Tests:**

$/MLM\_Suntrust/SunTrust/Web/Dev/4.0.3/SunTrustGiftPortalTest/GiftCodeTest.cs

$/MLM\_Suntrust/SunTrust/Web/Dev/4.0.3/SunTrustGiftPortalTest/GiftCodesTest.cs

$/MLM\_Suntrust/SunTrust/Web/Dev/4.0.3/SunTrustGiftPortalTest/CustomersTest.cs

$/MLM\_Suntrust/SunTrust/Web/Dev/4.0.3/Inetpub/wwwroot/SunTrustGiftPortalTest/AccountTest.cs

$/MLM\_Suntrust/SunTrust/Web/Dev/4.0.3/Inetpub/wwwroot/SunTrustGiftPortalTest/RewardsTest.cs

**Site Core:**

All content under

SiteCore/Content/SUNTRUSTREWARDS/GIFTPORTAL/ApplicationConfiguration/ParticipantWebConfig/LayoutInstances

SiteCore/Content/SUNTRUSTREWARDS/GIFTPORTAL/Content/ParticipantWeb/Pages

**Detailed notes by file** –

* Add notes here

# Code Review Disposition

## Follow Up Items

List here any items that require a post review follow-up. Always assign a follow up item to an individual and determine a due date.

Note: It is assumed that all S1 documented issues are to be remediated in the application code.

|  |  |  |  |
| --- | --- | --- | --- |
| **Topic** | **Comments** | **Assigned To** | **Due Date** |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |

## Mandatory Follow Up Code Review

Based on the findings of this code review, taking into consideration any severe issues, determine if another code review is required after application changes are completed. Check one of the boxes below:

Follow Up Code Review Required

Follow Up Code Review NOT Required

## Appendix A – Detailed PCI Analysis

Example: A2 – Injection Flaw -- Application uses Stored Procedures for data access and the database user has no rights () to the underlying database tables.