**사 용 권 한**

**Client Approval:**

본인은 서명으로써 본 문서가 본 프로젝트 범위 내에서 사용될 것을 인가함.

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **승인자 :** | 한국저작권위원회 차태원 팀장 | (인) | **일자 :** | \_\_\_\_/\_\_/\_\_ |

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **검토자 :** | 한국저작권위원회 문경도 선임 | (인) | **일자 :** | \_\_\_\_/\_\_/\_\_ |

**주식회사 굿씽크 & 엘에스웨어(주) Approval:**

본 문서에 대한 서명은 본 문서에 대하여 수행 및 유지관리의 책임이 있음을 인정하는 것임.

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **승인자 :** | 주식회사 굿씽크 곽종 부장 | (인) | **일자 :** | \_\_\_\_/\_\_/\_\_ |

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **검토자 :** | 엘에스웨어㈜ 김민 팀장 | (인) | **일자 :** | \_\_\_\_/\_\_/\_\_ |

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **작성자 :** | 엘에스웨어㈜ 신창권 상무 | (인) | **일자 :** | 2018/06/18 |

**제.개정 이력**

|  |  |  |
| --- | --- | --- |
|  |  |  |
|  |  |  |
|  |  |  |
|  |  |  |
|  |  |  |
|  |  |  |
|  |  |  |
|  |  |  |
|  |  |  |
|  |  |  |
|  |  |  |
| 1.0 | 제정 | 2018/06/18 |
| **개정번호** | **제.개정 페이지 및 내용** | **제.개정 일자** |

**목 차**

1 개요 5

1.1 표준화목적 5

1.2 표준화 범위 및 방향 5

1.3 표준화 대상 5

2 개발 환경 6

2.1 Web 애플리케이션(성능평가) 6

2.2 성능평가가 도구 아키텍쳐 7

2.3 Web 애플리케이션(저작권기술 R&D) 8

2.4 Java Architecture 9

3 개발 도구 10

3.1 개발 도구 및 버전 10

3.2 사용 환경 10

3.2.1 Client 10

3.2.2 성능평가 웹 Server 10

3.2.3 사업관리시스템 웹 Server 10

4 Application 구조 11

4.1 Application 구조도 11

5 Application 관리 규칙 11

5.1 소스 관리 파일 서버 11

5.2 기본 폴더 구성 11

5.2.1 TMIS WEB 서버 11

5.2.2 표준서비스 폴더 구성 13

5.3 소스 파일 명명 방법 및 작성자 13

5.3.1 일반사항 13

5.3.2 명명표기법 유형 14

5.3.3 디렉토리 14

5.3.4 파일 14

5.4 일반 명명 방법(Naming Rule) 16

5.4.1 변수 16

5.4.2 함수 16

5.4.3 클래스 16

5.4.4 인터페이스 16

5.4.5 메소드 16

5.5 구성 관리 17

5.6 자료 백업 17

6 시큐어 코딩 가이드 18

6.1 웹 어플리케이션의 필수 보안 가이드 18

6.2 유형별 시큐어 코딩 상세가이드 19

7 용어 표준 20

7.1 일반적 메시지 사용규칙 20

7.2 어미 사용 규칙 20

7.3 Exception Message 20

8 공통코드정의 20

# 개요

## 표준화목적

본 사업의 모든 개발업무에 있어 응용 프로그램 및 웹 개발의 표준을 제시함으로써 개발자들의 생산성을 향상시키고, 유지보수를 용이하게 하며, 사용자들의 편의성을 도모하고, 시스템에 대한 일관성 유지로 사용자 요구사항에 신속 대응함을 목적으로 한다.

## 표준화 범위 및 방향

* 전체페이지(화면) 구성에 대한 기본 템플릿을 제시한다.
* 프로그램 개발시의 산출물(소스 파일 등)에 대한 Naming Rule을 정한다.
* 프로그램 및 개발 소스에 대한 관리 표준을 정한다.

## 표준화 대상

* 파일명 및 파일의 표준
* 명명규칙(Naming Rule)
* 주석처리(Comment)
* 문서/소스의 관리
* 용어 표준
* UI 표준
* 공통 코드 정의
* 약어집

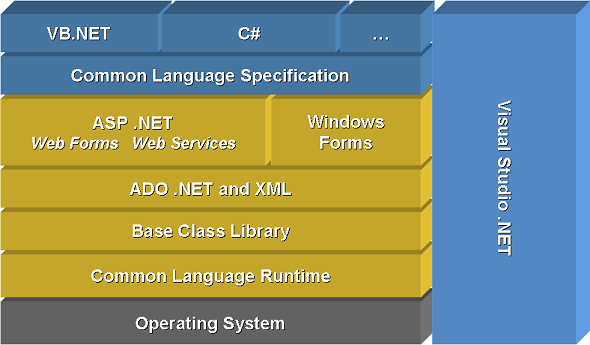
# 개발 환경

## Web 애플리케이션(성능평가)

웹 애플리케이션 JBOSS Web Application 을 활용하여 설계되어 있다

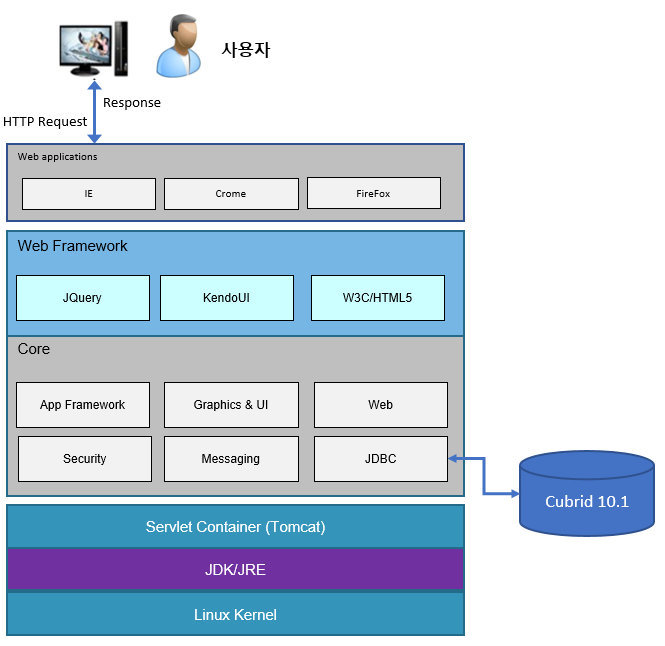
## 성능평가가 도구 아키텍쳐

Windows 애플리케이션은 닷넷 프레임워크를 사용하여 Windows OS 기반의 클라이언트 사용자들의 사용 편의성 및 확장성과 OS 제약 사항을 최소화를 지향하도록 설계하였다



## Web 애플리케이션(저작권기술 R&D)

웹 애플리케이션 Tomcat Web Application 을 활용하여 설계되어 있다



## Java Architecture



**Controller(Struts)**

**Business Logic 분기**

**Actions**

**View**

**(Jsp Page )**

**Data Acess**

**(Model)**

Sitemesh.xml

**Client Browser**

(1) Request

(11) Respond

(2)

(3)

(10)

Struts Config

**Business Logic**

**Service**

(8)

Application-context.xml

**DB**

(7)

(4)

(6)

(5)

(9)

# 개발 도구

프로그램 개발 시 사용될 개발 도구에 대한 특성, 버전, 툴을 사용할 수 있는 환경 요건 등을 기술한다.

## 개발 도구 및 버전

개발 도구별 버전 및 사용 용도에 대해 기술한다.

|  |  |  |  |
| --- | --- | --- | --- |
| Software | Ver | 구 분 | 용 도 |
| Eclipse Neon.3 | 4.6.3 | 개발도구 | Web App 개발도구 |
| Oracle Client | 11g | Database | 자료관리를 위한 Database |
| SQL Developer | 4.1 | DB 관리 툴 | Oracle DB 관리 및 쿼리 툴 |
| Adobe Photoshop | CS6 | Graphic Tool | Image Source 제작 |
| MS Office | 2013 | 문서작성Tool | 문서제작 |
| Erwin | 7.3.11.2812 | DB모델링 | DB모델링 제작 |
| Visual Studio | 2008 | 개발도구 | 검사 클라이언트 개발 |

## 사용 환경

### Client

* OS: Microsoft Windows XP/Vista, Internet Explorer 10이상
* 화면: 1024 x 768, High Color

### 성능평가 웹 Server

* Apache HTTPD Server 2.0
* JBOSS 5.0
* JDK 1.6
* Oracle 11g
* Oracle Client 11g

### 사업관리시스템 웹 Server

* Tomcat 9.0
* JDK 1.8
* Cubrid 10.1
* String 4.1
* eGoverment Framework 3.7

# Application 구조

## Application 구조도

**☞ 2180- Architecture\_정의서 .docx 참조**

# Application 관리 규칙

## 소스 관리 파일 서버

* 개발버전은 개발서버에, 운영버전은 운영 서버에 올려놓는다.
* 기존에 운영되었으나 사용하지 않는 디렉토리/파일은 운영서버에서 삭제하고 개발 서버에 백업 관리하는 것으로 한다.
* 운영서버에는 백업파일(.bak)및 테스트 파일이 존재하지 않도록 관리한다.

## 기본 폴더 구성

### TMIS WEB 서버

* 웹서버 : [tmis.copyright.or.kr](http://www.olis.or.kr) (222.231.43.106)
* 시작 경로 : /app/copyright/deploy

|  |  |  |  |
| --- | --- | --- | --- |
| **Depth 1** | **Depth 2** | **Depth 3** | **Description** |
| /copyright | admin |  | 관리자 페이지 |
| code | 평가항목 코드 관련 페이지 |
| common | 화면 메뉴 관리 및 session관리 |
| content | 콘텐츠 관리 |
| copyrighter | 권리자 관리 |
| cpc | Iframe 파일 |
| css | 디자인 css 파일 |
| db |  |
| edit | tmail editor |
| forum | 토론방 |
| images | 공통또는메인 화면 이미지 |
| js | 자바스크립트 파일 |
| mbr | 관리자 관리 |
| moni | 시스템 모니터링 |
| osp | Osp 사업자 관리 |
| performance | 성능평가 신청 관리 |
| pes | 성능평가 결과 관리 |
| pesGraph | 성능평가 결과 통계 그래프 |
| pesLog | 성능평가 로그 관리 |
| plaza | 게시판 관리 |
| research | 게시물관리 |
| stat | 표준서비스 관리 |
| tmail | Tmail 관리 |
| upload | 업로드 파일 보관 |
| jsp |  | 로그인관련 페이지 |
| cert | 보안 키 과련 |
| css | 디자인 관리 |
| images | 이미지 파일 |
| js | 자바스크립트 파일 |
| tpms |  | 사용자 페이지 |
| board | 게시판 관리 |
| common | 공통화면 |
| contackus | 신청접수 관련 페이지 |
| css | 디자인 관리 |
| TMIS | 저작권 기술 화면 |
| downLoad | 파일 다운로 로직 파일 |
| error | 에러 관리 |
| images | 공통 이미지 |
| js | 자바스크립트 파일 |
| mail | 메일관련 템플릿 관리 |
| perform | 성능평가신청 페이지 |
| performdb | 특징점 DB 사용신청 |
| sitemap | 사이트맵 화면 |
| WEB-INF |  |  |
|  | classes | Java 컴파일 파일 |
|  |  | lib | 라이브러리 관리 |

### 표준서비스 폴더 구성

* 표준서비스 매니저

|  |  |  |  |
| --- | --- | --- | --- |
| **Depth1** | **Depth2** | **Depth3** | **설명** |
| /R-FIMS | /STDManager | / | 표준서비스 모듈 및 필터링 업체 모듈 프로그램이 설치되어 있는 폴더 |
| /log | 표준서비스 로그가 저장된 폴더 |
| /db | / | Database 폴더 |
|  | /backup | Database 백업용 폴더 |

|  |  |  |  |
| --- | --- | --- | --- |
| **Depth1** | **Depth2** | **Depth3** | **설명** |
| /media | /ftpdata | / | FTP 루트폴더로 콘텐츠 및 메타정보가 올라오는 폴더 |
| /complate | 표준 서비스 모듈 및 필터링된 콘텐츠가 최종 적으로 옮겨지는 폴더 |

## 소스 파일 명명 방법 및 작성자

이 규칙은 컴퓨터가 효율적으로 실행될 수 있는 코드를 만들어야 한다는 목표와 인간이 쉽게 이해 할 수 있는 코드를 만들어야 한다는 서로 상충되는 두 가지 목적을 모두 만족시키기 위한 것이다

이러한 지침들의 효과는 바로 지속적인 사용에 있다. 모든 팀원들이 그 지침에 따라야 한다. 위 사항을 만족하기 위해서는 원하는 문자를 조합할 수 있도록 기준을 마련해야 하며, 새로운 클래스나 사용자 정의 데이터 유형을 만들 때에는 새로운 약자를 만들어야 한다.

### 일반사항

* 임의의 약어 대신 완벽한 영어 단어를 사용한다.
* 업무영역에 적합한 정의된 용어를 사용한다.
* 명칭의 가독성을 높이기 위해 대소문자를 원칙에 따라 혼용한다.
* 약어 사용시 정의된 약어를 사용한다.
* 너무 긴 명칭을 삼가 한다. (15자 이내 권장)
* 비슷한 명칭이나, 대소문자만 다른 명칭 사용을 삼가 한다.

### 명명표기법 유형

* Pascal 표기법
  + 각 단어의 첫 문자는 대문자로 표기
  + 클래스 명명규칙은 Pascal 표기법에 따른다
  + 사용 예) StandardCode
* Camel 표기법
  + 첫 문자는 소문자로 시작하여 그 이후 단어의 첫 문자는 대문자로 표기
  + 파일명 명명규칙은 Camel표기법에 따른다.
  + 함수 명명규칙은 Camel 표기법에 따른다.
  + 사용 예) standardCode
* Upper Case 표기법
  + 모든 문자는 대문자로 표시
  + 단어수가 1~2자로 적은 문자수를 가지는 식별자로 표기
  + 단어 사이에는 UnderLine(\_)을 사용하는 것으로 한다.
  + 사용 예) MAX\_VALUE

### 디렉토리

* 모든 디렉토리의 명칭은 영문자의 조합으로만 구성되어야 한다.
* Java 디렉토리의 명칭은 소문자로 한다.
* C# 디렉토리의 명칭은 첫 문자는 대문자로 한다.
* WEB-INF 처럼 특별한 경우일 경우 대문자를 사용할 수 있다.

### 파일

* jsp 파일 명은 관리를 쉽게 할 수 있도록 업무에 관련된 이름을 부여한다.
* jsp 파일명은 Camel 표기법을 사용한다
* 모든 파일명은 영문을 사용하고 숫자는 가급적 사용하지 않는다.
* 길거나 자주 사용되는 용어의 길이는 약어를 사용하여 적절히 줄인다.
* 외부 범위의 이름과 동일한 이름을 내부 범위에서 사용하지 않도록 한다.
* 공백을 허용하지 않는다.
* 가급적 서술어의 경우 아래의 Postfix를 기준으로 한다
* 구분코드는 파일의 의미를 알 수 있는 단어의 조합으로 한다.

(구분코드를 emp로 한 경우)

|  |  |  |
| --- | --- | --- |
| 구분 | Postfix | 예제 |
| 목록 | List | empList.jsp |
| 조회 | Detail | empDetail.jsp |
| 등록 | Insert, Form | empInsert.jsp / empForm.jsp |
| 수정 | Modify | empModify.jsp |
| 삭제 | Delete | empDelete.jsp |
| 처리 | Proc | empInsertProc.jsp |
| 성공 | Success | empInsertSuccess.jsp |
| 실패 | Fail | empInsertFail.jsp |

* 파일 확장자는 다음의 규정을 따른다.

|  |  |
| --- | --- |
| 파일구분 | 확장자명 |
| Java Source File | java |
| JavaScript File | js |
| Hyper Text Markup Language File | html |
| Cascading Style Sheet File | css |
| Java Server Page File | jsp |
| Shock Wave File | swf |
| GIF File | gif |
| JPEG File | jpg |
| C# Source File | cs |
| eXtend Markup Language | xml |
| Icon File | Ico |
| Visual C++ File | cpp |
| C File | C |
| Header File | h |

* XML, XLS 등이 스크립트 기반 언어에 의해 작성되는 경우에 한해 확장자 규정을 예외로 할 수 있다.
* 파일 확장자는 소문자이어야 한다.
* 기타 정의되지 않은 파일 확장자는 일반적인 사용례에 따른다.

## 일반 명명 방법(Naming Rule)

### 변수

* 기본적으로 변수는 소문자로 표기한다.
* 특별한 경우 소문자와 대문자 조합으로 표기할 수 있다.

### 함수

* 기본적으로 Camel 표기법에 따른다.
* 함수명은 함수가 수행하고자 하는 바를 명확하고 간결하고 빠르게 이해할 수 있어야 한다.
* 함수명 뒤의 괄호는 붙여서 사용한다.

### 클래스

* 기본적으로 Pascal 표기법에 따른다.
* 단어들의 첫 문자는 반드시 대문자로 하여 두 단어 이상이 결합되어 사용되어야 하는 경우 새로운 단어의 시작 역시 반드시 대문자로 한다.
* 클래스명에 약어를 사용할 경우 대문자를 사용
* Class의 이름은 간단하면서도 Class에 대한 설명이 되도록 작성한다. 가능하면 너무 긴 이름이 되지 않도록 한다.
* Class 이름은 정해진 약어를 제외하고 가능한 완전한 단어로 쓰는 것을 원칙으로 한다.

### 인터페이스

* 클래스와 동일한 규칙을 따른다.

### 메소드

* 첫 문자는 소문자를 사용한다
* 중간에 들어가는 단어의 첫 글자는 대문자로 한다
* 메소드의 경우 보통 동사를 많이 사용한다
* 단어사이의 구분은 ‘\_’(underbar)를 사용하지 않는다
* 기능에 따라 적절한 어미를 붙이는 것을 권장한다.

|  |  |  |
| --- | --- | --- |
| 기능 | add | 추가를 처리하는 경우 |
| modify | 수정을 처리하는 경우 |
| remove | 삭제를 처리하는 경우 |
| is | 존재여부 확인하는 경우 |
| insert | 데이터 INSERT 처리 |
| update | 데이터 UPDATE 처리 |
| select | 데이터 조회 처리 |
| delete | 데이터 삭제 처리 |
| 세부기능 | 대상클래스명 or 처리로직을 표현할 수 있는 영문명 or 대상을 표현하는 영문명 or 대상 테이블명[컬럼명]  예) User, UserInfo | |
| 명명표준 | 기능 + 세부기능, 예) add + User = addUser | |
| 명명표준 예제 | addUser()  modifyUserInfo()  removeUser()  isManager()  insertUser()  updateUserName()  selectUserInfo()  deleteUser() | |

## 구성 관리

* Application 소스의 구성 관리는 프로젝트 기술서의 구성관리 계획서에 따른다.
* 인자가 바뀌거나 중요한 로직의 변경/추가시 해당 program source file의 history를 아래와 같이 history를 관리한다.
* 규칙 : {원래 이름}\_YYMMDD.{확장자}
* YY - 갱신년(예: 2008년 - 08)
* MM – 갱신월(예: 1월 – 01)
* DD – 갱신일(예: 9일 – 09, 15일 – 15)

예) app\_reg.asp를 1월 20일에 대폭 수정한 경우 원래 source파일을 app\_reg\_080120.jsp로 수정

## 자료 백업

* 단위 프로그램의 소스는 해당 개발자가 개인 PC의 Project 폴더에 수시로 백업하며, 프로그램 작성 및 테스트는 개발 상황에 따라 서버 및 개인PC에서 진행한다.
* 최종 단위 프로그램은 모듈 관리자가 관리하는 파일 Server에복사한다.
* 모듈 관리자는 당일 업무종료 시간에 백업을 한다.
* 단, 모듈 관리자가 부재 시에는 위임을 받은 자가 백업을 한다.
* 백업 파일의 보존 기간은 백업 받은 일로부터 프로젝트 종료일까지 보존한다.

# 시큐어 코딩 가이드

본 과업에서는 시큐어 코딩을 준수하되, 웹어플리케이션에서 가장 많이 취약하다고 알려진 부분에 대해서는 반드시 시큐어 코딩 가이드를 준수하는 것을 그 원칙으로 한다.

## 웹 어플리케이션의 필수 보안 가이드

|  |  |  |
| --- | --- | --- |
| **웹 애플리케이션의 10대 취약점** | | |
| **A1** | **입력 값 검증 부재** | 웹 요청 정보가 웹 애플리케이션에 의해 처리되기 이전에 적절한 검증이 이루어지고 있지 않다. 공격자는 이 취약점을 이용하여 웹 애플리케이션의 백 엔드 컴포넌트를 공격할 수 있다. |
| **A2** | **취약한 접근 통제** | 인증된 사용자가 수행할 수 있는 작업을 적절히 제한하지 않고 있다. 공격자는 이 취약점을 이용하여 다른 사용자의 계정에 접근하거나, 민감한 정보가 담긴 파일을 열람하거나, 허용되지 않은 작업을 수행할 수 있다. |
| **A3** | **취약한 인증 및**  **세션 관리** | 계정 토큰과 세션 토큰이 적절히 보호되고 있지 않다. 공격자는 암호나 키, 세션 쿠키, 기타 인증 관련 토큰을 공격하여 인증을 우회하고 다른 사용자의 ID를 가장할 수 있다. |
| **A4** | **크로스 사이트**  **스크립팅(XSS)**  **취약점** | 웹 애플리케이션이 다른 사용자의 브라우저를 공격하는 도구로 사용될 수 있다. 공격이 성공하는 경우 일반 사용자의 세션 토큰이 노출되거나, 사용자의 컴퓨터를 공격하거나, 다른 사용자를 속이기 위해 위조된 컨텐츠를 보여주게 된다. |
| **A5** | **버퍼 오버플로우** | 웹 애플리케이션 컴포넌트가 사용자의 입력값을 적절히 점검하지 않는 언어로 작성되어 다운될 수 있다. 특수한 경우에는 공격자가 해당 프로세스의 권한을 획득할 수 있다. 이 컴포넌트로는 CGI, 라이브러리, 하드웨어 드라이버, 웹 애플리케이션 서버 컴포넌트 등이 포함된다. |
| **A6** | **삽입 취약점** | 웹 애플리케이션이 외부 시스템이나 자체 OS에 접근할 때 입력받은 인자를 그대로 전달한다. 공격자가 해당 인자로 악의적인 명령어를 삽입하는 경우, 해당 외부 시스템은 웹 애플리케이션으로 인해 입력받은 명령어를 실행할 수 있게 된다. |
| **A7** | **부적절한 에러 처리** | 일상적인 운용 과정 중에 발생하는 에러 상황에 대해 적절한 처리가 이루어지지 않는다. 공격자가 웹 애플리케이션이 처리하지 못하는 에러가 발생하도록 유도하여, 해당 시스템에 대한 상세 정보를 획득하거나, 서비스를 방해하거나, 보안 메커니즘이 작동하지 않도록 할 수 있으며, 서버가 다운될 수도 있다. |
| **A8** | **취약한 정보**  **저장 방식** | 웹 애플리케이션은 정보나 인증 관련 토큰을 보호하기 위해 암호화를 자주 사용한다. 암호화 관련 기능이나 코드는 적절하게 구현하기가 어려움이 이미 증명되었으며, 많은 경우 오히려 보안상 바람직하지 않은 결과를 초래한다. |
| **A9** | **서비스 방해 공격** | 공격자가 다른 정당한 사용자가 사이트에 접속하거나, 애플리케이션을 사용하는 것을 방해하기 위해 웹 애플리케이션의 리소스를 고갈시킬 수 있다. 공격자는 또한 다른 사용자가 본인 소유의 계정을 사용하지 못하도록 계정을 잠글 수 있으며, 심지어 웹 애플리케이션 전체가 멈추도록 할 수 있다. |
| **A10** | **부적절한 환경 설정** | 강화된 서버 환경 설정 표준을 보유하는 것은 안전한 웹 애플리케이션에 있어 결정적으로 중요한 부분이다. 해당 서버는 보안에 영향을 미치는 다양한 환경 설정 옵션이 있으며, 벤더 출하시에는 기본적으로 안전하지 않은 상태로 출시된다. |

## 유형별 시큐어 코딩 상세가이드

유형별 시큐어 코딩에 대한 상세한 설명이나 예제, 점검 방법등은 별첨을 참고하도록 한다.

☞ 별첨 : 가장 심각한 10가지 웹 애플리케이션 보안 취약점.docx

# 용어 표준

## 일반적 메시지 사용규칙

* 사용자 메시지는 공통 메시지 페이지를 작성하여 사용한다.
* 사용자가 이해할 수 있는 사용자 위주의 문구를 사용하고, 간결하고 정확해야 한다.
* 메시지에 사용되는 문법, 용어, 약어 등을 일관성 있게 사용한다.
* 실패 메시지의 경우, 처리 결과와 조치방법이 제시되어야 한다.

## 어미 사용 규칙

|  |  |
| --- | --- |
| Avoid | Recommend |
| 하십시요. | 하십시오.  하시겠습니까? |
| 하세요. |
| 하시기 바랍니다. |
| 주세요. | 주십시오. |
| 했습니다. | 하였습니다. |
| 되었습니다. |
| 저장되었습니다. | 저장하였습니다. |
| 없습니다. | 존재하지 않습니다. |

## Exception Message

jsp, java, Exception 에러 처리는 try ~ catch ~ finally 를 사용을 기본으로 하고 finally는 생략이 가능하다.

C# Exception 에러 처리는 try ~ catch ~ finally 를 사용을 기본으로 한다.

# 공통코드정의

* **☞ 별첨 : 2110-개발-표준\_코드\_설계서-v1.0.xlsx**