California State University Northridge	**Information Technology** Application Development Standard	**Page** 1 of 3	
		SOP#: 90-09-029	**Revision#:** Version 0.7

Prepared by: Kathryn Sharron **Date:** April 15, 2009	**Approved by:** Chris Olsen, ISO **Date:** June 29, 2009
Last revised by: Kevin Krzewinski **Date:** January 31, 2016	**Last approved by:** Kevin Krzewinski, ISO **Date:** January 31, 2016

1.0 PURPOSE

To provide campus application developers with the baseline standards for developing applications that will be Internet-facing.

2.0 SCOPE

CSUN-developed applications intended for enterprise-wide use, use by the public, a college or a division, or any application that will transact or store protected data. Class projects and research projects that do not transact/store protected data are not required to follow the standard.

3.0 RESPONSIBILITY

	Role (Title)	**Responsibility**
1	Application Developers	Request vulnerability scans prior to deploying Internet-facing applications. Mitigate risks discovered by the vulnerability scans.
2	Information Security Analyst	Conduct vulnerability scans of Internet-facing applications; coordinate with application developers to identify/remediate vulnerabilities.
3	Information Security Officer (ISO)	Review and approve changes to the application development standards.

4.0 STANDARDS

4.1 Documentation and Environmental Requirements

- Document the business requirements; ideally using use cases.
- Obtain sign-off for the business requirements from the requester.

4.2 Development

- Use a test or development environment to develop the application; rather than developing in production.

- Implement access-segregation controls; a role-based approach is recommended. In addition, enforce access control at the "GET" level to prevent a user from bypassing access-controls and accessing a web page containing protected data.
- Use server-side validation rather than client-side validation;
- Require a secure (SSL/https) connection for transmittal of protected data.
- Encrypt account credentials and other Level 1 data either by using an encrypted database or an encrypted file. The encryption required is 256-bit encryption. Recommended: AES, RSA or Blowfish. Use robust input validation techniques to avoid SQL injections and Cross-Site Scripting.
- Write loop statements so that they have a way out if the intended conditions aren't met, and use timeouts when the application will need to wait for another event to occur. Both will protect against Denial of Service or Distributed Denial of Service attacks.
- Write user and administrative error messages to provide sufficient information to identify the cause of the error and that do not disclose protected information such as user roles, credentials, system vulnerabilities, etc.
- Use established web development conventions and frameworks whenever possible (MVC, JSON, JQuery, etc.). They speed application development, address commonly-needed functions like form validation, and usually include built-in security.

4.3 Testing

- Create and document a test plan driven by the business requirements. This test plan may include accessibility, automated, cross-browser, load, regression and/or user-acceptance testing, or other forms of testing not listed here.
- Conduct tests according to the plan and fix bugs. Have a potential end user of the application, a fellow developer, or the business owner (or designee), conduct the final round of testing. Document and retain the completed test plans for a period of at least one year.

4.4 Pre-production/Deployment Checklist

- Document final sign-off by the business owner (or designee) to confirm that the application functions as required, that the test plan has been completed successfully, and to facilitate project close-out and reassignment of resources.
- Identify known and potential vulnerabilities:
    - Request an application vulnerability scan from the Information Security Office by submitting a request through the campus case management system or sending an email to it.security@csun.edu. Remediate vulnerabilities disclosed in the vulnerability scan report.
    - Conduct a review of the code to identify known and potential vulnerabilities not discoverable from the vulnerability scan (for example: look for hard coded passwords). Use automated

code checkers whenever possible, such as those included in IDEs. Remediate vulnerabilities disclosed in the review.

- Retain documented evidence of identified vulnerabilities and remediating actions for a period of at least one year.

- Follow a Change Management Review (CMR) process for the release to production of any application alteration that significantly affects functionality or has the potential to negatively or positively affect system users; this includes new versions, modules, significant changes to the look and feel, etc. The campus IT CMR process is available for this purpose. To join a weekly Wednesday CMR meeting, send an email to rca@csun.edu.

4.4 Post- deployment Patches/Updates

- See Campus-wide Patch Management Process.

4.5 3rd Party Applications

- In addition to adhering to the above standards, Application Service Providers must comply with CSUN's policy on "IT SECURITY REQUIREMENTS FOR APPLICATION SERVICE PROVIDERS."

5.0 DEFINITIONS:

6.0 REFERENCES:

CSUN POLICY 500-14 APPLICATION SERVICE PROVIDER SECURITY REQUIREMENTS

CSUN Accessibility Best Practices - http://www.csun.edu/accessibility/bestpractices/

SANS Top 25 Errors - http://www.sans.org/top25errors/

OWASP Application Development Standards - http://www.owasp.org/

CSUN Protected Information - http://www.csun.edu/it/security/protecteddata.html

Change Management Review Process: IT/CMR File Share folder

7.0 FURTHER INFORMATION: