

Success Story

“Enosis’ Testing and QA team formulated and executed a structured and disciplined software test cycle for an Enterprise Scale Web Application”



Background

While developing an Enterprise Scale Web Application to serve millions of customers, the client faced significant challenges in formulating a strategic direction for Quality Assurance (QA) and Testing. To guarantee superior quality a wide variety of tests had to be run to check the functionality of the application and to rectify the bottlenecks. There was a lack of standardization in QA and an absence of a centralized testing organization. The client turned to Enosis’ Testing and QA team to ensure in time knowledge acquisition, adequate test coverage & early defect detection. Enosis leveraged the proven expertise of its enterprise-wide Quality Assurance & Testing team and produced significant benefits in terms of cost savings, process improvement & enhanced productivity for the client.



The Client

SpoonByte is a US based “software on demand” service provider offering iPhone & Web app that allows restaurants to create promotions (e.g., meal discounts, menu specials, or events) to increase revenue. It’s the first app that serves the food industry by focusing on *Match-Based Deal Dissemination* for the physical marketplace. SpoonByte intelligently sends each promotion only to consumers that are near the restaurant and are predicted to act on the promotion and visit the restaurant.

- Focusing on testing iteratively, as often as stable code base is available and until quality is achieved from end customer’s perspective
- Conducting a large variety of tests to check the functionality of the application and to rectify the bottlenecks
- Executing intensive load testing to check the hardware sufficiency of the web server, its ability to handle increasing number of users and its capability to survive a massive spike in user traffic
- Ensuring the web application would run properly on major browsers

THE APPLICATION

Restaurant owners were struggling to fill empty tables even during peak dinner time and consumers lacked the ability to find the appropriate deals at the right time. To address the needs of the food service industry the client collaborated with Enosis to develop a customized Business Intelligence Web based tool that uses the client’s high end predictive modeling technology **to connect the right customer to the right restaurant with the right discounts.** Restaurants are able to use SpoonByte to directly impact demand by creating real time deals that SpoonByte routes directly to end consumers. SpoonByte intelligently routes these limited time deals to users by using information on their preferences and proximity.

CHALLENGES

Enosis SQA team had to overcome many complex obstacles in order to fulfill the client’s stringent quality requirements, since the application has to support Restaurant owners at one end and hundreds of thousands of consumers at the other. The key features of the QA process are:

- Running intricate web security testing which ensures proper data protection and checks if web based application requirements are met when exposed to malicious input



SpoonByte Web Application

ENOSIS’ APPROACH

The Enosis approach comprises of vigorous testing activities that are performed in a well structured method and offer several benefits through governance, standardization and reusability. This approach provides a single governance umbrella that leverages best practices and standardization across different functions. Enosis’ Web Application testing services include:

- Analysis of requirements and posting queries to eliminate confusion
- Setting up a rapid communication channel inside the team regarding feedbacks
- Functional Testing (Sapient and Automated approach)
- Security testing
- Data-driven Testing
- Integration Testing
- System Testing
- Usability testing
- User Interface testing
- Compatibility and Multi Language Support Testing
- Cross Browser Testing
- Exploratory and Smoke Testing
- W3C Compliance and Regression Testing
- Performance testing
- Load/Stress Testing
- User Acceptance Testing

The key testing activities that form the backbone of the Enosis' QA process are mentioned below:

- Analyze UCD workflows, identify risk areas, build scenarios and post questions to validate the requirements
- Explore other web applications and compare the features with SpoonByte UI prototypes
- Analyze database design, structures and post suggestions to build an optimized design
- Verify database objects naming pattern against database naming convention rules and report the problems
- Verify Nhibernate objects mapping file against class diagram and post the problems
- Build automated Unit and Integration Tests on Domain Repository classes and Controller classes
- Verify Save/Update/Find operations output and match against corresponding database storage

- Verify JSON output data and map against corresponding Mobile UI to identify the redundant or missing data field.
- Generate and Test Data using tool and validate it
- Prepare Test Document Suite (Functional Test Scripts, Requirement Traceability Matrix, Summarized checklists for login and Registration module) for each Subsystem
- Test User interface controls, user interactions and expected functionalities and find out defects between the interfaces of the integrated modules, with the aid of simulated scenarios
- Test both the Web based and Mobile based application in terms of functionality, usability, security, compatibility and performance aspects
- Build Automated Functional Regression test suite
- Review production code to identify logical errors and to ensure proper error handling mechanism
- Exercise every independent execution path through the component to ensure all statements in the program are executed at least once
- Test all conditional statements for both true and false cases
- Organize Release Procedure (Automated build process, Semi automated Regression Test Suite execution, Release Note)
- Measure code blocks performance
- Test web server load and configuration issues

THE SOLUTION

The Enosis QA Team has a verified set of actions for Web application testing that ensure the ability to track down all the potential bugs and errors in an application and contribute to the overall improvement of the quality of the Web Application.

Functional Testing

- **Links verification:**
 - Testing all internal links

- Testing the outgoing links from all the pages from a specific domain
 - Testing links jumping on the same pages
 - Testing links used to send email to the administrator or other users from web pages
 - Checking if there are any orphan pages
 - Checking for broken links in all the above-mentioned links
- **Forms validation:** Forms are the integral part of any web site. Forms are used to obtain information from users and to interact with them. Form validation includes:
 - Checking all validations on each field
 - Checking for the default values of fields
 - Testing if wrong inputs to the fields in the forms are restricted
 - Checking options to create, delete, view or modify the forms
 - All these validations are checked in both manual and automated ways
 - **Cookie testing:** Cookies are small files stored on user end machines. These are used to save the session credentials, especially those of login sessions. Cookie testing involves:
 - Test the application by enabling or disabling the cookies from the browser options
 - Testing if the cookies are encrypted before writing to user machine
 - Testing the session cookies (i.e. cookies that expire after the session ends) for login sessions and user statistics after session ends
 - Checking effect on application security by deleting the cookies
 - **HTML/CSS validation:** To optimize the site for Search Engines, HTML/CSS validation is very important. The site is validated for HTML syntax

errors and checked to ensure crawlers of different search engines are able to find it. Furthermore, the site is checked for scripting (e.g. JavaScript) errors

- **Database testing:** Data consistency is very important in web application. Data integrity and errors that occur while user edit, delete, add forms or do any DB related functionality are checked thoroughly. Database query execution, data retrieval and update are also checked. Then the retrieved results are cross checked by running the queries in database and verifying the results

Usability and UI Testing

- **Navigation checking:**
 - Website is checked for ease of navigation
 - Checking if the instructions provided serve the purpose
 - Checking if main menu is consistent on each page
- **Content checking involves checking:**
 - Simplicity of the layout
 - Spelling mistakes
 - Presence of unnecessary links/button/input fields/form elements etc
 - Clarity of contents on each page
 - Presence and content of sitemap
 - Images for descriptive alt information (i.e. value in "ALT" tag)
 - Color combination/pattern etc should be lightweight e.g. image(s) should not be large
- **Connectivity Testing:** The major interfaces are Load Balancer - Application Server interface and Application Server - Database Server interface. Connectivity testing involves:
 - Checking if all the interactions between these servers are executed properly and errors are

handled properly. If database or web server returns any error message for any query by application server then application server must display these error messages to the user

- Checking cases such as users interruption to any transaction and web server connection reset

▪ **Compatibility Testing:**

- **Browser compatibility testing:** Web applications are very dependent on the browsers. As different browsers have different configurations and settings, the compatibility of the web application must be checked to ensure that web coding is cross-browser platform compatible. As web application is using Javascript or AJAX calls for UI functionality and performing security checks or validations, more stress is given on browser compatibility testing. The web application is tested on different browsers like IE, Firefox, Chrome, Safari, Opera browsers with different versions
- **OS compatibility testing:** Some functionalities of the web application may not be compatible with all operating systems. New technologies used in web development like graphics design, interface calls etc. may not be available in all operating systems. The web application is tested on different operating systems like Windows, Unix, MAC, Linux etc.
- **Cell phone testing:** The web application is further tested for compatibility with browsers available on cell phones
- **Testing Printing options:** As the application provides page-printing options, fonts, page alignment, page size compatibility, page graphics etc. must be tested for proper printing

Security Testing

Testing the Web application security is a vital segment of the testing process. Security testing involves:

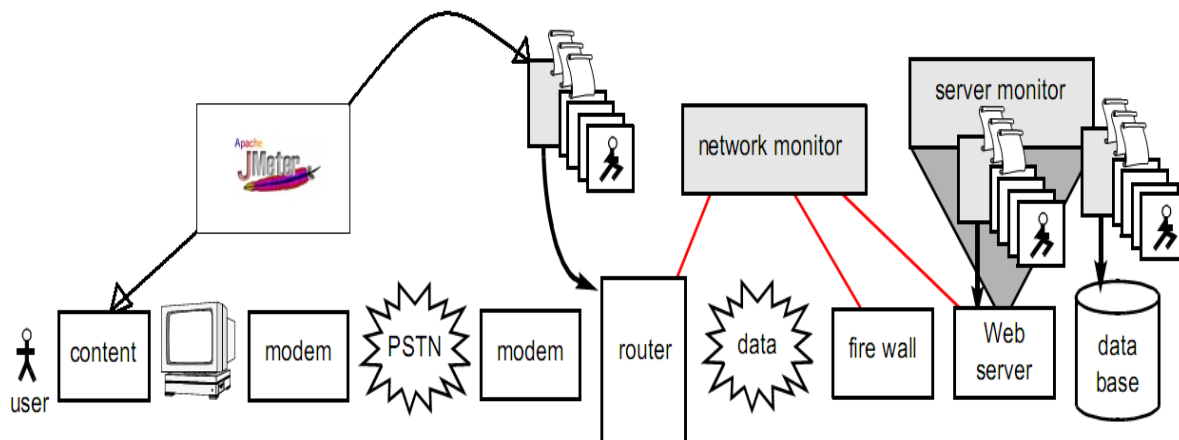
- **Looking at the application from every perspective:** A reconnaissance is performed on the Web application to see how Google presents the site and how it behaves with hacking tools such as Found Stone's Site Digger. Next, a Web vulnerability scanner such as HP's WebInspect and Acunetix Web Vulnerability Scanner is run on the application. Where possible the scans are run as both an unauthenticated and un-trusted outsider and as an authenticated and trusted user (via basic HTTP, NTLM or form authentication)
- **Test for underlying weaknesses:** One of the most commonly overlooked areas of Web application testing is to scan the underlying operating system and installed applications. Tools such as Nessus and Qualys Guard are used to identify problems such as missing patches and configurations errors in the operating system and other software that has been installed (including the Web server itself). Back-end databases and related network infrastructure systems are also checked for underlying weaknesses
- **Manually check for weaknesses:** Due to the scarcity of automated tools manual testing is compulsory on login mechanisms, form input validation and sensitive information buried in HTML and server directories. This removes risks from malicious users who might be inclined to attempt SQL and Script injection attacks
- **Testing source code:** The web application's source code must be analyzed regularly. Some tools such as DevInspect and Checkmarx are used to check for software flaws at the source

- Log files of all server requests/transactions, error messages and security breach attempts are maintained on web server
- Markup (HTML) and CSS are validated so that W3C standards are satisfied. It is essential to defend vulnerable Cross Site Scripting attacks
- Server directory listing is disabled
- It is verified whether the system is showing expected result in case of timeout limits
- Some decoder is applied to automatically decode CAPTCHA image
- Database tempering from an outside environment is restricted

- Maximum number of connections is getting access to the server simultaneously without a busy signal.
- The returned content is verified when full load is applied to the server
- Achieve zero downtime even when the application is running for long periods of time
- Web page load time is measured

TOOLS AND TECHNOLOGIES

- **Programming Language:** C#.
- **Frameworks:** Asp.NET MVC, WCF, Windows Service.



Load testing diagram

Load/Stress Testing

- Testing application performance on various internet connection speeds
- **Web load testing:** This involves testing the site with many simultaneous user requests, large input data from users, simultaneous connections to database, heavy load on specific pages etc. This checks if the system can sustain itself during peak hours
- **Stress testing:** Stress is generally exerted on input fields, login and sign up areas to check how the system responds and recovers from crashes

- **Web Technologies:** Html, XHTML, Javascript, CSS, AJAX.
- **Relational Database Management System:** PostgreSQL, Microsoft SQL Server.
- **Object Relational Mapping (ORM) Tool:** NHibernate.

Tools Used

- **Automated Functional Testing Tool** - Selenium, WATIN
- **XHTML & CSS Validator Tool** - Total Validator, W3C Validator, W3C mobileOK Checker, CSS Validator

- **Broken Link Checker Tool** - Xenu's Link Sleuth, W3C Link Checker
- **Spell Checker Tools** - Total Validator, IESPELL
- **Cookie Watcher/Viewer Tools** - IECookiesView v1.74, MozillaCookiesView v1.27, Safari Cookies, Fire Cookies
- **Webpage Control Inspectors** - Firebug (firefox), DebugBar (IE)
- **Webpage Performance Analyzers** - YSlow (Firebug extension & Add-on)
- **HTTP Packet Analyzer/Sniffers** - WebBug, WireShark, Live HTTP Header
- **Load/Stress/Performance Testing Tool** – Jmeter
- **SQL Injection** - Hackbar, SQL Injection!
- **CAPTCHA Decoder** - PWNtcha
- **Userscripts Executor** - GreaseMonkey
- **Maximum Length Input Generator** - PerlClip
- **Test Data Generator** - Advance Data Generator