

Client Requirement

The system is a web-based application for nonprofit organization that focuses on assessment and curriculum development of basic skills for youth and adults. CASAS is used by federal and state government agencies, business and industry, community colleges, education and training providers, correctional facilities and technical programs.

Client required a system which could manage basic skills tests developed by them that are administered to more than five million youth and adults. The products and services should certify language and basic skills proficiency and promote the transition to higher education and the workforce.

Following is the list of modules present in the application:

- Workshops
- Workshop trainers
- Training Registration
- Online Registration
- Global email templates
- Email confirmation text for registration
- Contacts and VContacts

Project Challenges

Team faced the following challenges:

- Developing a system to track all workshops, workshop trainers and participants.
- Developing and implementing a framework that smoothly handles posting workshops in training registration for online registration.
- Electronic training materials.
- Assign descriptions and email confirmation text per workshop.
- Password protected content for special audiences.
- Manage Summer Institute online registration.

Technologies

Used

Operating System & Server Management	<i>Windows OS, Multi-Server Architecture with Staging & Production Environment through Version Controlling releases, Server Optimization, Security & SSL Implementation, Scheduler for Back-ups, Alert Monitoring System Integration, Server Performance Tuning at regular intervals, Software Firewall Configuration & Maintenance</i>
Development Tools & Environments	<i>ASP.Net, Sitefinity CMS, MVC, C# etc.</i>
Database	<i>MS-SQL Database Server 2008, Report Service (SSRS), DB Clustering, DB Optimization, Master Slave Replication, Query Optimization, Scheduler for Backups</i>

Manpower

Project Leader	<i>1</i>
Developers	<i>3</i>
Designers	<i>1</i>
Integrator	<i>1</i>
DBA	<i>1</i>
Quality Assurance Testers	<i>1</i>

Planning

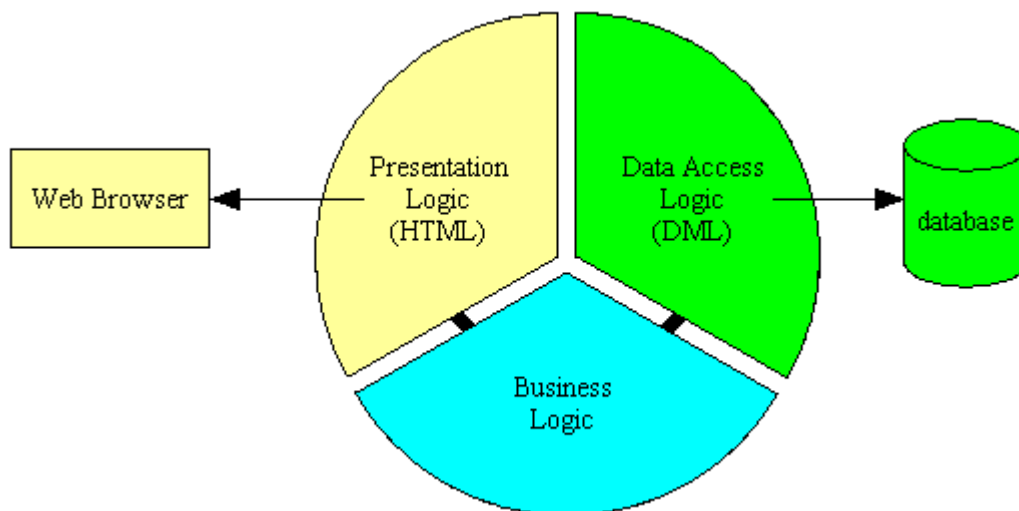
Keeping in mind the enormous structure of the website and the challenges involved, a complex approach was adopted consisting of:

- The Database layer containing MSSQL Server Database, Tables, Stored Procedures and so on.
- The Data Access layer containing the libraries responsible for accessing data from the database.

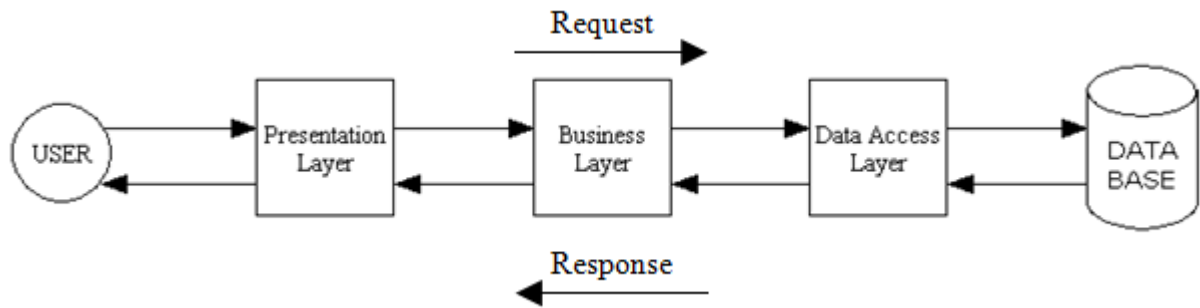
- The Business Logic layer consisting of all the business logic procedures for modules like user profiles, groups and registration info etc.
- The User Interface layer which forms the Graphical User Interface of the website.

Architecture

- We used a 3-tier architecture which includes Presentation Layer (PL), Business Access Layer (BAL) and Data Access Layer (DAL).
- Business Access Layer: This layer contains the business logics, validations related to data. This layer also contains Model-View-Controller which efficiently relates the user interfaces with the underlying data models.
- Data Access Layer: This layer contains MSSQL Server Database, Tables, Stored Procedures and so on. It also helps Business Layer to connect the data and perform required action by returning data or manipulating data (insert, delete, update).
- Presentation Layer: This layer contains .asp or html where data is presented to the user and inputs are taken from the user.



- Note here that the presentation layer has no direct communication with the data access layer - it can only talk to the business layer.
- The cycle of requests and their associated responses can be shown in the form of a simple diagram:



- The Presentation layer requires skills such as HTML, CSS and possibly JavaScript plus UI design.
- The Business layer requires skills in a programming language so that business rules can be processed by a computer.
- The Data Access layer requires SQL skills in the form of Data Definition Language (DDL) and Data Manipulation Language (DML) plus database design.

The main advantages of the 3-Tier Architecture are:

- Flexibility - By separating the business logic of an application from its presentation logic, 3-Tier architecture makes the application much more flexible to changes.
- Maintainability - Changes to the components in one layer have no effect on any others layers. Also, different layers require different skills (such as HTML/CSS is the presentation layer, .Net in the business layer, SQL in the data access layer) so these can be managed by independent teams with skills in those specific areas.
- Reusability - Separating the application into multiple layers makes it easier to implement re-usable components. A single component in the business layer, for example, may be accessed by multiple components in the presentation layer or even by several different presentation layers (such as desktop and the web) at the same time.
- Scalability - 3-Tier architecture allows distribution of application components across multiple servers thus making the system much more scalable.
- Reliability - 3-Tier architecture if deployed on multiple servers, makes it easier to increase reliability of a system by implementing multiple levels of redundancy.

Deployment Architecture

For deployment we are using Windows Server 2008 with IIS7. We are using MVC architecture. Benefits of MVC are given below:

1. Enables the full control over the rendered HTML.
2. Provides clean separation of concerns (SoC).
3. Enables Test Driven Development (TDD).
4. Easy integration with JavaScript frameworks.
5. Following the design of stateless nature of the web.
6. Restful urls that enables SEO.
7. No View-State and Post-Back events.
8. It provides RAD development.
9. Easy development models for developers those coming from WinForm development.

The application provided the client with a system capable of handling all the data and with ease of access for everyone. The detailed requirement gathering done on-site helped setting up a well-designed system with all modules integrated into it along with control over user access.

Registration data was easily transported from machine to the database and in turn shown to user on web interface. The control of site was given to administrator who can set up users, user roles and assign data-level access to each user. The Sitefinity Content Management System was fairly straightforward to set up and very easy to use which made the implementation process of the site very smooth and quick.