EdutorialOnline:

Software Architectural Description

Brett Lee

Introduction

EdutorialOnline provides K-12 educational institutions and parents an option for online supplementary education through the use of cutting-edge games, puzzles, interactive social media interfaces and other fun and engaging instructional modules, and reporting tools using the Adaptive Instruction Module Engine (AIME). In addition to AIME, educators and content developers are able to use the Module Development Kit (MDK) to provide content, which can be accessed through an API by game and web-application developers for delivery through a number of fun and interactive interfaces.

Key Terms

AIME
: The Adaptive Instruction Module Engine: the core platform for generating, and reporting individualized education experience.

API
: Application Programming Interface: the necessary libraries, services and conventions used by web and game developers [1] when accessing AIME user data.

Availability
: Continuous system functionality and the consequences of system failure (e.g., up-time and recovery). [5]

Conceptual Integrity
: The "underlying theme or vision that unifies the design at all levels" [5].

Correctness and Completeness
: The ability of the architecture "to allow for all of the system's requirements and runtime resources constraints to be met" [5].

Cost and Benefit
: The level of benefit, or the quality of product, which is achieved for a given budget. [5]

Framework (or, *web application framework*)
: A foundation to aid developers in the implementation of (web) applications, providing common core functionality [2].

LAMP
: A web development platform combining **L**inux (operating system), **A**pache (web server), **M**ySQL (Database), and **P**HP (hypertext preprocessing, server-side language), commonly abbreviated **LAMP** [8].

MDK

The Module Development Kit: the tools provided to content/curriculum creators to easily create and manage knowledge areas, standards and assessment levels, content, and related elements used by AIME for generating adaptive instruction modules.

Modifiability

The cost or difficulty, related to change in a system. [5]

Performance

The timing of user-facing interruptions/delays, and resources necessary for maintaining a high level of functionality. [5]

PHP

"[A]n open source, server-side, HTML embedded scripting language used to create dynamic Web pages" [3].

Security

The system's ability to prevent unauthorized access. [5]

Targeted Market

Refers to whom and how the product will be made available. [5]

Testability

The difficulty in exposing software or usability faults. [5]

Time to Market

The ability for development time to be condensed in response to pressure from external sources [5].

Usability

The ease for a user to complete a task. [5]

Yii

A web application development framework written in PHP [4].

Environment

The software systems detailed in this architectural description are built upon the Yii PHP framework, deployed within a LAMP environment. Specific details relating to the Yii framework architecture and the features it provides should be referred to the official Yii Framework documentation [7].

Stakeholders

- Curriculum and Content developers
- End users (students)
- Parents
- Educational institutions
- Database Administrators
- Software Engineering/Developers
- Hosting Services/Systems Engineers/Support
- Sales and Marketing/Management

Stakeholder Concerns

Stakeholder concerns are the focus of the architectural description. Specific concerns to be covered include system quality attributes and crosscutting concerns.

System Quality Attributes

System quality attributes are those qualities, which concern most system stakeholders as they relate to product qualities.

Availability

Primary Stakeholders

- Curriculum and Content developers
- End users (students)
- Parents
- Educational institutions
- Hosting Services/Systems Engineers/Support

Response Measure

Availability can be measured in the amount of time the system is completely or partially non-functional.

Assessment

Yii provides exception handling built upon the existing exception framework of the PHP processing modules [6]. System availability is contingent upon availability measures and best practices taken up within the LAMP stack.

Modifiability

Primary Stakeholders

- Software Engineering/Developers
- Sales and Marketing/Management
- Hosting Services/Systems Engineers/Support

Response Measure

Modifiability can be measured by the effort necessary to make changes to the software system for maintenance and introduction of new functionality.

Assessment

The Yii framework provides tools that automatically generate much of the code necessary to support a new feature or change to functionality [6].

Performance

Primary Stakeholders

- End users (students)
- Parents
- Educational institutions
- Hosting Services/Systems Engineers/Support
- Sales and Marketing/Management

Response Measure

Performance response is measured in the length of delays to accomplishing a user-initiated task.

Assessment

The Yii framework has placed high importance on issues of performance and has implemented a number of features for high performance out-of-the-box as well

configurable elements for performance tuning [7]. Performance tuning best practices should also be applied to the LAMP stack for optimal results.

Security

Primary Stakeholders

- End users (students)
- Parents
- Educational institutions
- Software Engineering/Developers
- Hosting Services/Systems Engineers/Support

Response Measure

Security is measured by the level of effort necessary to circumvent protections to gain unauthorized access to the system or sensitive data.

Assessment

Yii offers some security features to aid in the prevention of certain types of attacks, [7] but security best practices should be employed in the design as a crosscutting concern.

Testability

Primary Stakeholders

- Software Engineering/Developers

Response Measure

Testability is measured by the level of effort necessary to find defects in the software system.

Assessment

The Yii framework comes standard with features that can allow for some powerful testing capabilities [7]. Leveraging these features to implement a test-driven approach to development can improve testability by ensuring that each component is built with relevant unit tests.

Usability

Primary Stakeholders

- Curriculum and Content developers
- End users (students)
- Parents
- Sales and Marketing/Management

Response Measure

Usability is measured by how effectively (and efficiently) a user is able to complete a task.

Assessment

User interface design should follow best practices of usability. Compartmentalized UI components (PHP files) that are commonly used should be created according to modern web standards, utilizing cascading style sheets (CSS) to provide a consistent user experience.

Crosscutting Concerns

Significant design effort should be place on the following crosscutting concerns, which are not limited to a single component within the web application stack.

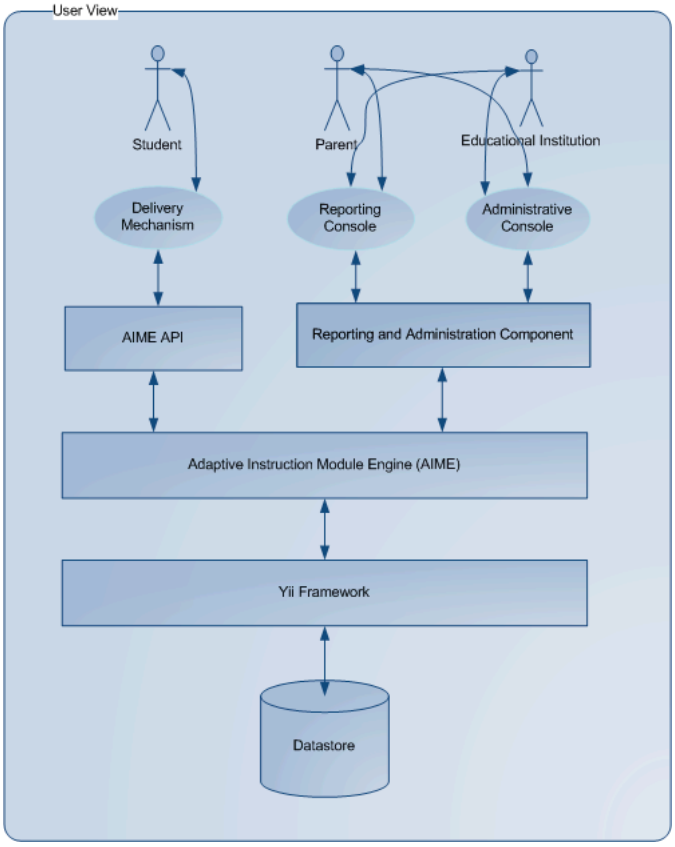
- Security
- Logging
- Fault detection
- Reporting

Architectural Views

User View

The user view provides a look at the architecture from the viewpoint of the student parent and educational institution end users.

Primary Presentation



Elements

Users

Students, Parents and Educational Institutions are represented here as end users connecting to the system via a web browser.

Delivery Mechanism

The Delivery Mechanism can consist of a social web application, a game, a puzzle, Flash media, or other interface.

Reporting Console

A suite of reporting tools provided through a web interface for querying data related to student performance.

Administrative Console

A portal for the general administration of user accounts, billing and security settings, among others.

AIME API

A series of PHP and JavaScript libraries and documentation to provide access to AIME resources and services for developers.

Reporting and Administrative Component

A web application technology stack to support the Reporting and Administrative Consoles.

Adaptive Instruction Module Engine (AIME)

The core logic built within the Yii framework for generating individualized instructional modules.

Yii Framework

The Yii Framework is more than just a layer here, but the foundation for the development of all the other components, providing services available throughout the stack.

Datastore

Represents the database (or database farm) where the actual data resides.

Interfaces

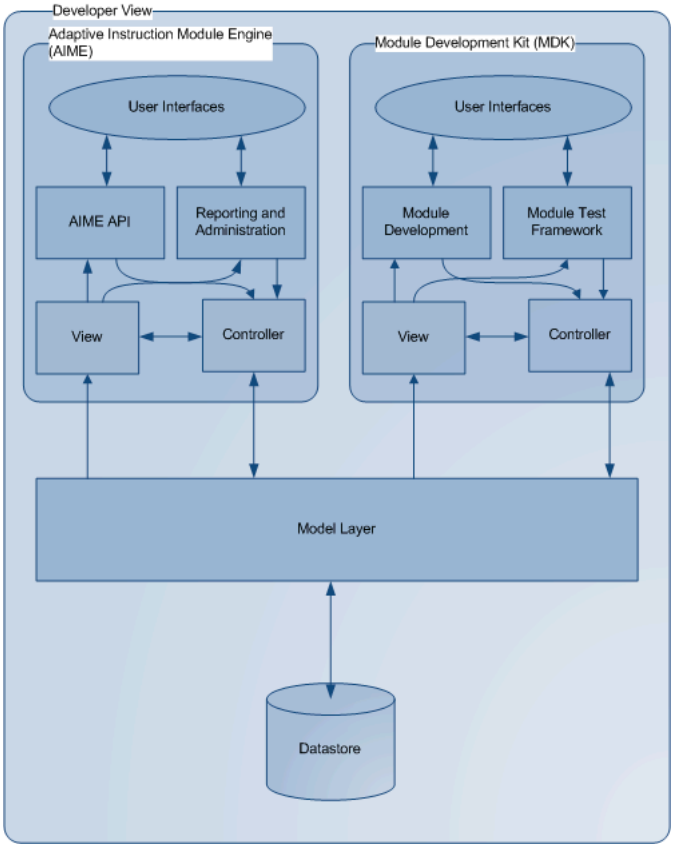
The interface from the users web browser to the web server is made via HTTP request. Other elements interface via HTTP request, both traditional and through AJAX calls, according to the design patterns implemented by the Yii framework.

Rationale

The design illustrated provides a layered-style view of the architectural elements as data makes its way to and from the users through the various layers to the database, and the relationships between components.

Developer View

Primary Presentation



Elements

User Interfaces

Generic representation of the user-facing components.

AIME API

A series of PHP and JavaScript libraries and documentation to provide access to AIME resources and services for developers.

Reporting and Administrative

A web application technology providing Reporting and Administrative functions.

Module Development

A suite of web-based tools for the creation of instructional modules.

Module Test Framework

A test suite providing module developers the ability to test modules created by adjusting input parameters to simulate a student's interaction through a delivery mechanism.

Controller

In the Model-View-Controller (MVC) architectural pattern, the controller interprets user input and applies application logic [6].

View

In the MVC, the view correlates to the user interface and works cooperatively with the controller, together comprising the presentation element [6].

Model Layer

In MVC, the model handles data abstraction and much of the business logic [6].

Datastore

Represents the database (or database farm) where the actual data resides.

Interfaces

The interface between the various elements largely follows design patterns implemented by the Yii framework.

Rationale

The design makes strong use of the MVC architectural pattern as implemented by the Yii PHP framework, adhering to the separation of concerns (SoC) principle.

References

[1] API, from Webopedia [2011]. Available: http://www.webopedia.com/TERM/A/API.html

[2] Web application framework, from DocForge [2011]. Available: http://docforge.com/wiki/Web_application_framework

[3] PHP, from Webopedia [2011]. Available: http://www.webopedia.com/TERM/P/PHP.html

[4] About Yii, from YiiFramework [2011]. Available: http://www.yiiframework.com/about/

[5] Len Bass, Paul Clements and Rick Kazman. Software Architecture in Practice, Second Edition. (2003) Addison-Wesley, pp 79-90.

[6] Martin Fowler. *Model View Controller*, from *GUI Architectures* (2006). Available: http://www.martinfowler.com/eaaDev/uiArchs.html

[7] Qiang Xue and Xiang Wei Zhuo. The Definitive Guide to Yii 1.1 (2010). Available: http://yii.googlecode.com/files/yii-docs-1.1.7.r3135.tar.gz

[8] LAMP, from Webopedia [2011]. Available: http://www.webopedia.com/TERM/L/LAMP.html