IITB Summer Internship 2013

Software Requirement Specification (SRS)

Optimizing Moodle LMS for Improving User Response Time

Principal Investigator
Prof. D. B. Phatak

Project In-charge
Mr. Nagesh Karmali

Project Mentors
Mr. Anoop Dobhal
Mr. Dhanajay Ambekar
Miss. Furuza Aibara (PMO)
Mr. Nilesh Birari
Mr. Raju.B
Miss. Silpa T.

Project Team Members
Miss. Akshaya Muralidharan
Mr. Mevin Dominic
Mr. Pawan Deshmukh
Miss. Priyanka Manchanda
Miss. Shabna T. R
Mr. Surya Pratap Singh
Miss. Vintha Slipa Reddy
# Table of Contents

1. **Introduction** ................................................................................................................................. 4
   1.1 Purpose.............................................................................................................................................. 4
   1.2 Document Conventions.................................................................................................................. 4
   1.3 Acronyms and Abbreviations.......................................................................................................... 4
   1.4 Intended Audience and Reading Suggestions.................................................................................. 4
   1.4.1 Optimized Moodle LMS and Quiz Plugin with Long Polling Mechanism:................................. 4
   1.4.2 Policy Enforcement Framework................................................................................................. 5
   1.5 Project Scope.................................................................................................................................... 5
   1.5.1 Optimized Moodle LMS.............................................................................................................. 5
   1.5.2 Policy Enforcement Plugin.......................................................................................................... 5
   1.6 References ........................................................................................................................................ 6

2. **Overall Description** ......................................................................................................................... 7
   2.1 Product Perspective.......................................................................................................................... 7
   2.1.1 Optimized Moodle LMS.............................................................................................................. 7
   2.1.2 Policy Enforcement Framework................................................................................................. 7
   2.1.3 Quiz Plugin with Long Polling Mechanism................................................................................ 7
   2.1.4 Description of random batch-mode connectivity algorithm....................................................... 8
   2.2 Product Features............................................................................................................................ 8
   2.2.1 Optimized Moodle LMS.............................................................................................................. 8
   2.2.2 Policy Enforcement Framework................................................................................................. 8
   2.2.3 Quiz Plugin with Long Polling Mechanism................................................................................ 8
   2.3 User Classes and Characteristics................................................................................................... 9
   2.3.1 Optimized Moodle LMS and Quiz Plugin with Long Polling Mechanism .................................. 9
   2.3.2 Policy Enforcement Framework................................................................................................. 9
   2.4 Operating Environment................................................................................................................. 9
   2.5 Design and Implementation Constraints...................................................................................... 9
   2.6 User Documentation....................................................................................................................... 9
   2.7 Assumptions and Dependencies...................................................................................................... 10

3. **System Features** ............................................................................................................................... 11
   3.1 Login Page ...................................................................................................................................... 11
   3.1.1 Description and Priority.............................................................................................................. 11
   3.1.2 Stimulus/Response Sequences ................................................................................................... 11
   3.1.3 Functional Requirements........................................................................................................... 11
   3.2 View Course................................................................................................................................... 11
   3.2.1 Description and Priority.............................................................................................................. 11
   3.2.2 Stimulus/Response Sequences................................................................................................... 11
   3.2.3 Functional Requirements........................................................................................................... 12
   3.3 Attempt Quiz Activity ..................................................................................................................... 12
   3.3.1 Description and Priority.............................................................................................................. 12
   3.3.2 Stimulus/Response Sequences................................................................................................... 12
   3.3.3 Functional Requirements........................................................................................................... 12
   3.4 Participate in Chat Activity ............................................................................................................. 12
   3.4.1 Description and Priority.............................................................................................................. 12
   3.4.2 Stimulus/Response Sequences................................................................................................... 12
   3.4.3 Functional Requirements........................................................................................................... 12
   3.5 View News Forum ........................................................................................................................... 13
   3.5.1 Description and Priority.............................................................................................................. 13
   3.5.2 Stimulus/Response Sequences................................................................................................... 13
   3.5.3 Functional Requirements........................................................................................................... 13
   3.6 View User Profile ........................................................................................................................... 13
3.6.1 Description and Priority .......................................................... 13
3.6.2 Stimulus/Response Sequences .................................................. 13
3.6.3 Functional Requirements .......................................................... 13
3.7 Logout ......................................................................................... 13
3.7.1 Description and Priority .............................................................. 13
3.7.2 Stimulus/Response Sequences .................................................... 13
3.7.3 Functional Requirements ............................................................ 13
3.8 Long Polling Requirements on the Client-Side .................................... 14
3.8.1 Description and Priority .............................................................. 14
3.8.2 Functional Requirements ............................................................ 14
3.9 Long Polling Requirements on the Server-Side .................................... 14
3.9.1 Description and Priority .............................................................. 14
3.9.2 Functional Requirements ............................................................ 15
3.10 Settings to use for random batch-mode connectivity algorithm ............ 15

4. External Interface Requirements .......................................................... 16
4.1 User Interfaces .............................................................................. 16
4.2 Hardware Interfaces ..................................................................... 16
4.3 Software Interfaces ...................................................................... 16
4.4 Communications Interfaces ........................................................... 16

5. Other Nonfunctional Requirements .................................................... 17
5.1 Performance Requirements ............................................................. 17
5.2 Software Requirements ................................................................. 17
5.3 Safety Requirements ...................................................................... 17
5.4 Security Requirements ................................................................. 18
5.5 Software Quality Attributes .......................................................... 18

6. Other Requirements ........................................................................ 19

Appendix A: Glossary ........................................................................ 19
6.1 Definitions .................................................................................... 19
1. **Introduction**

1.1 **Purpose**

The purpose of this section is to provide the reader with a general, background information and an insight into the web application of “Optimized Moodle Learning Management System”, to specify the requirements for policy enforcement framework for Moodle and the requirements for developing a long polling mechanism for quiz module in Moodle.

All the techniques long-polling, random number generation, are experimentations over Moodle aimed either to improve optimization, throughput, or increase number of users those operating simultaneously to take quiz or other activity. This technique random number generation to Moodle is also aimed at testing over Moodle and see if number of users or the throughput is improved.

1.2 **Document Conventions**

1.3 **Acronyms and Abbreviations:**

- **AJAX**: Asynchronous JavaScript and XML
- **PHP**: Hypertext Preprocessor (originally "Personal Home Page")
- **LMS**: Learning Management System
- **SRS**: Software Requirements Specification
- **UC**: Use Case
- **SDD**: Software Design Description
- **Policy**: A proposed or adopted course or principle of action
- **Context**: Defines the circumstance, such as time and location in which the policy is valid.

1.4 **Intended Audience and Reading Suggestions**

1.4.1 **Optimized Moodle LMS and Quiz Plugin with Long Polling Mechanism:**

The intended audience for the Optimized Moodle LMS and Quiz Plugin with Long Polling Mechanism is the web designers, Academicians and the Education Industry comprising of Institute Heads, students and faculty who are currently using or are interested to use the Moodle LMS.

For a growing number of applications, however, this form of polling isn’t enough. AJAX polling almost inevitably introduces a delay between the occurrence of an event on the server and the notification of the event to the client (since there is a new request from the client after request before the event).

Long polling places a smaller number of requests to the server compared with AJAX polling.
1.4.2 Policy Enforcement Framework

**Developer:** If the developer wants to read, change, modify or add new requirements into the existing system, he/she must first consult this document, update the requirements in an appropriate manner and pass the information correctly to the other phases of the development process.

**User:** The user of this program reviews the specifications presented in this document determines if the software has all the suitable requirements and whether the software developer has implemented all of them.

1.5 Project Scope

1.5.1 Optimized Moodle LMS

Optimized Moodle LMS is a very useful web application for education industry. This application supports each and every feature of Moodle LMS but it significantly reduces the average user response time for each activity. Numerous Optimization Techniques have been applied to optimize the Moodle LMS.

1.5.2 Policy Enforcement Framework

Policy enforcement deals with users security and privacy concerns, by allowing them to define policy rules.

**The main goals are:**

- To restrict the usage of resources
- To prevent privilege escalation attack
- To provide fine-grained access control

When integrated with Moodle as a plugin, these policy enforcements can help the teacher/admin to block some applications, such as Wikipedia during a quiz/test

1.5.3 Quiz Plugin with Long Polling Mechanism

While developing an interface which requires continuous, real-time access to fast-changing data on the server-side, polling over HTTP is incredibly inefficient as these connections take time to establish and they also add stress on the server. With long polling, the client places the request and the server doesn’t reply until it has information to return. The Web client keeps a pending connection that’s closed only when some valid response can be returned.
1.6 References

- http://www.it.iitb.ac.in/arndg/dokuwiki
- High Performance MySQL Optimization Backups Replication And More - OReilly Publication
  Web Applications: A Proposal to Improve Response Time and Its Application to MOODLE
- www.xphp.info/php-tutorial/long-polling/
2. Overall Description

2.1 Product Perspective

2.1.1 Optimized Moodle LMS

Moodle (acronym for Modular Object-Oriented Dynamic Learning Environment) is a free source e-learning software platform, also known as a Learning Management System, or Virtual Learning Environment (VLE). As of June 2013 it had a user base of 83,008 registered and verified sites, serving 70,696,570 users in 7.5+ million courses with 1.2+ million teachers.

2.1.2 Policy Enforcement Framework

This section considers an idea of policy enforcement framework for Moodle on Aakash tablet.

Some of the use-cases explaining why a policy enforcement framework is required are as follows:

- When Moodle is being used for conducting quizzes or exams in schools, only the quiz or exam related apps should get open. Any request to start any of the remaining apps should be blocked.
- During school-time, students can open a limited set of apps.
- List of allowed apps would be defined by schools (teachers). For example, students should not be able to open social networking apps or gaming apps during school-time (say, 9:00am to 3:30pm) while a quiz is being conducted.

Context Attributes
The policy framework allows users to define context attributes such as time and location. If the system time and location matches with the context attributes, then the policy is valid. Otherwise it is invalid and the next policy is considered.

2.1.3 Quiz Plugin with Long Polling Mechanism

AJAX Polling
AJAX polling almost inevitably introduces a delay between the occurrence of an event on the server and the notification of the event to the client. Long polling places a smaller number of requests to the server compared with AJAX polling.

Long-Polling
With long polling, the client places the request and the server doesn’t reply until it has information to return. The Web client keeps a pending connection that’s closed only when some valid response can be returned. That’s exactly what we want. Long polling places a smaller number of requests to the server compared with AJAX polling, when real-time communication required, but each request could take much longer.
2.1.4 Description of random batch-mode connectivity algorithm:-

In an client server, communication for a certain upper limit of during simultaneous access, to further increase this limit of number of users, this technique can be used as algorithm which intakes number of users simultaneously accessing and assign them with a random number generated in such a way that users get grouped by same random number and this is equal to the round in which he shall download the quiz.

2.2 Product Features

2.2.1 Optimized Moodle LMS

This web application is optimized using the following techniques:

- Use of Image Maps
- Changing Hardware Configuration to SSD
- Removing Irrelevant Data Displayed During Quiz Activity
- Use of far future expires headers

2.2.2 Policy Enforcement Framework

The main features of policy framework are:

- Add more context

A policy can have more than one context. The details like context name, from time, to time and location attributes like latitude, longitude and range are entered.

- Edit policy

The policy attributes can be modified by the teacher.

- Rearrange priority

The policy priority can be altered easily by using drag and drop facility to sort the policies.

2.2.3 Quiz Plugin with Long Polling Mechanism

Start: make the quiz available and wait for students to join.(for instructor)

Join:-join to give the quiz which is made available.(for student)

Next:-pass a question to student screen.
2.3 User Classes and Characteristics

2.3.1 Optimized Moodle LMS and Quiz Plugin with Long Polling Mechanism

- **Web Designers**
  
  A web designer is someone who engages in web design. Web design encompasses many different skills and disciplines in the production and maintenance of websites. The web designers can employ the techniques discussed in this document to optimize their websites.

- **Education Community**

2.3.2 Policy Enforcement Framework

This application is of great use to the education community who are currently using or are interested in using Moodle LMS.

- **Physical Actors:**
  
  Student: The student is the one who login to moodle and then gets himself/herself enrolled in a course and take part in a quiz

  Teacher: The teacher is the one who enforce policy on certain application while a quiz is being conducted

2.4 Operating Environment

OE-1: All the three products shall operate with the any web browser.

OE-2: All the three products shall operate on Windows 7 and Ubuntu 12.10 operating system.

OE-3: All the three products are free of cost and can be accessed by any user.

2.5 Design and Implementation Constraints

CO-1: The system shall be developed using Firebug, HttpFox and I Macros extension for Firefox.

CO-2: The system shall be tested using Apache JMeter.

2.6 User Documentation

UD-1: The system shall provide an online link to the learning resources available on the Moodle Docs for the understanding of the user.
2.7 Assumptions and Dependencies

AS-1: The server used for Optimized Moodle LMS is Apache 2.22.

AS-2: The database used for Optimized Moodle LMS is MySQL 5.5.31.

AS-3: The Moodle Version 2.5 is used to develop the system.

DE-1: The operation of the Optimized Moodle LMS depends on the number of concurrent users and the activities accessed.
3. **System Features**

3.1 **Login Page**

3.1.1 **Description and Priority**

Provide the user with a page to login to Optimized Moodle LMS.
Priority = 9

3.1.2 **Stimulus/Response Sequences**

Stimulus: User clicks on Login Link.
Response: Login Page is displayed
Stimulus: User Enters Username and Password
Response: Username and Password are validated from MySQL Database.
Stimulus: User Clicks on Login Button
Response: Home Page is displayed if Username and Password is correct else Error Message is displayed.

3.1.3 **Functional Requirements**

REQ-1: The user shall be able to view and click on Login Link.
REQ-2: The user shall be able to enter the username and password
REQ-3: The database shall be able to validate username and password.

3.2 **View Course**

3.2.1 **Description and Priority**

Provide the user with a page to view courses and to view activities associated with each course.
Priority = 9

3.2.2 **Stimulus/Response Sequences**

Stimulus: User clicks on Courses Link
Response: Courses are displayed
Stimulus: User Clicks on a particular course
Response: The course page and associated activities are displayed.
3.2.3 Functional Requirements

REQ-1: The user shall be able to view and click on the Courses Link.

REQ-2: The user shall be able to view the activities associated with each course.

3.3 Attempt Quiz Activity

3.3.1 Description and Priority

The user shall be able to attempt the quiz activity.
Priority = 9

3.3.2 Stimulus/Response Sequences

Stimulus: User clicks on Quiz Activity
Response: The quiz is displayed.
Stimulus: User clicks on Next
Response: The subsequent quiz page is displayed.
Stimulus: User clicks on Finish Quiz
Response: The quiz result is displayed.

3.3.3 Functional Requirements

REQ-1: The user shall be able to view and answer the quiz questions

3.4 Participate in Chat Activity

3.4.1 Description and Priority

The user shall be able to participate in chat activity.
Priority = 9

3.4.2 Stimulus/Response Sequences

Stimulus: User clicks on Chat Activity
Response: Chat Room is displayed.
Stimulus: Chat Room is displayed.
Response: Message is displayed.
Stimulus: User posts messages.
Response: Message is displayed.

3.4.3 Functional Requirements

REQ-1: The user shall be able to view chat room and post messages.
3.5 View News Forum

3.5.1 Description and Priority
The User shall be able to view news forum.
Priority = 9

3.5.2 Stimulus/Response Sequences
Stimulus: User clicks on News Forum
Response: News Forum Page is displayed.

3.5.3 Functional Requirements
REQ-1: The system shall be able to view news forum.

3.6 View User Profile

3.6.1 Description and Priority
The User shall be able to view User Profile.
Priority = 9

3.6.2 Stimulus/Response Sequences
Stimulus: User clicks on Settings -> User Profile
Response: User Profile page is displayed.

3.6.3 Functional Requirements
REQ-1: The user shall be able to view user profile.

3.7 Logout

3.7.1 Description and Priority
The user shall be able to Logout.
Priority = 9

3.7.2 Stimulus/Response Sequences
Stimulus: User clicks on Logout link
Response: User is logged out and index page is displayed.

3.7.3 Functional Requirements
REQ-1: The user shall be able to logout from the System.
3.8 Long Polling Requirements on the Client-Side

3.8.1 Description and Priority

The requests made from the client-side will always feature a parameter describing the data set currently held by the client-side. In a messaging application, this data is the set of messages that the client-side has already received. The parameter describing it is the timestamp of the latest received message. Including this parameter will instruct the server-side to respond only when new data is available. When implementing long polling on the client-side, bear in mind that the polling request might terminate for any other reason.

The Long Polling Requirements on the Client must be satisfied.
Priority = 9

3.8.2 Functional Requirements

REQ-1: The requests made from the client-side shall feature timestamp of the latest received message, describing the data set currently held by the client-side

3.9 Long Polling Requirements on the Server-Side

3.9.1 Description and Priority

When it receives a request, your script needs to retrieve the parameter sent by the client-side which describes the required data set. In the case of the web messenger, this will be the timestamp of the last message received by the client. Your PHP script must then connect to its data source and poll it for messages with a higher, more recent timestamp. The PHP script will keep on querying its data source until it finds new information. This could be done in a while loop with the connection to the database established before entering the repetition. As soon as the information is found, your script will output it and finally terminate.

Ways to do:
1) Modification time of file on server-side.
2) URL time stamp method.
3) Change of a global variable by trigger from a third person.

The Long Polling Requirements on the server must be satisfied.
Priority = 9
3.9.2 Functional Requirements

REQ-1: The script shall retrieve the parameter sent by the client-side which describes the required data set.
REQ-2: The script shall connect to its data source and poll it for messages with a higher, more recent timestamp.
REQ-3: The PHP script shall keep on querying its data source until it finds new information.
REQ-4: As soon as the information is found, the script shall output it and finally terminate.

3.10 Settings to use for random batch-mode connectivity algorithm

Moodle quiz when done settings that all questions to be displayed at once, they are downloaded at one time, giving us scope to implement the Random number generation download method.

The quiz plugin folder, consists of php, css, java script files, with their flow as:-

view.php > startattemp.php > other supportive php files.

View.php : contains the attempt quiz button with its action file being startattempt.php is triggered.

Startattempt.php : authenticates the user and subsequently fetches the quiz and starts attempt.
4. **External Interface Requirements**

4.1 **User Interfaces**

UI-1: The system shall provide all functionalities and activities supported by Moodle LMS

UI-2: The pages shall permit complete navigation and item selection.

UI-3: Provide User interface to create a policy

UI-4: Provide User interface to add a context

UI-5: Provide User interface to edit policy

UI-6: Provide User Interface to attempt quiz.

UI-7: Provide User with a Quiz Plugin which Supports Long Polling

4.2 **Hardware Interfaces**

HI-1: **Processor** 1GHz or faster

HI-2: **RAM** 1GB (32bit) or 2GB (64bit)

HI-3: **Hard disk space** 16GB (32bit) or 20GB (64bit)

4.3 **Software Interfaces**

Apache Server 2.2 and MySQL 5.5.31 and PHP 5.4.6 must be installed

4.4 **Communications Interfaces**

CI-1: The system shall use MySQL Database named “Moodle”

CI-2: The system shall be able to send emails to the users.

CI-3: The system uses PHP 5.4.6 to generate Moodle pages.
5. Other Nonfunctional Requirements

5.1 Performance Requirements

PE-1: Responses to queries shall take no longer than 3 milliseconds to load onto the screen after the user submits the query for 1 user.

PE-2: The system shall display confirmation messages to users within 4 milliseconds after the user submits information to the system.

PE-3: The system should generate policy with an accuracy of 99%.

5.2 Software Requirements

If you’re using a MySQL database, your query will return immediately, even if there are no results. This will cause the script to use a lot of processing power as it’s running a loop. Add a sleep call just after the query inside the while loop.
Furthermore, define a maximum execution time.
When Moodle quiz is given settings, all questions are to be displayed at once, they are downloaded at one time, giving us scope to implement the Random number generation download method.

The quiz plugin folder consists of php, css, java script files, with their flow as:-

view.php > startattemp.php > other supportive php files.

view.php: contains the attempt quiz button with its action file being startattemp.php is triggered.

Startattemp.php: authenticates the user and subsequently fetches the quiz and starts attempt.

for your script. This should be slightly less than external time constraints configured on the server-side (Example: max Execution_time in php.ini).

Make sure you record the starting timestamp first thing when the script starts. After every iteration of the polling query, check the elapsed time since the script was started. Break the loop and return and empty response when the elapsed time approaches the maximum limit.

5.3 Safety Requirements

Consistency: Checking the fact that all clients must be attached to one server, so there is an appropriate control of the information.
5.4 Security Requirements

SE-1: The files generated by the user are only accessible by the admin and application should store these files in MySQL database and must no share them.

SE-2: The policy framework should be accessible only to teachers.

5.5 Software Quality Attributes

Availability-1 The system shall be available to users all the time.

Availability -2 The system shall always have something to function and always pop up error messages in case of component failure.

Efficiency-1: The system shall generate the correct pages with an accuracy of 99%.
Efficiency-2: The system shall provide the right tools to support all its features.
6. Other Requirements

All the requirements have been specified

Appendix A: Glossary

6.1 Definitions:

*Moodle*: Moodle (acronym for Modular Object-Oriented Dynamic Learning Environment) is a free source e-learning software platform, also known as a Learning Management System, or Virtual Learning Environment (VLE).

*Apache JMeter*: The Apache JMeter™ desktop application is open source software, a 100% pure Java application designed to load test functional behavior and measure performance. It was originally designed for testing Web Applications but has since expanded to other test functions.

*Web Application*: A web application is an application that is accessed by users over a network such as the Internet or an intranet.

*Web Design*: Web design encompasses many different skills and disciplines in the production and maintenance of websites.

*Web Designer*: A web designer is someone who engages in web design.

*Web Optimization*: Website optimization is a phrase that describes the procedures used to optimize – or to design from scratch