Software Requirement Specifications

For

Activity Tracker Moodle Plug-In

Table of Contents

Table of Contents

[Description 3](#_Toc171898170)

[Operating Environment 3](#_Toc171898171)

[Assumptions and Dependencies 3](#_Toc171898172)

[External Interface Requirements 4](#_Toc171898173)

[User Interface 4](#_Toc171898174)

[Hardware Interface 4](#_Toc171898175)

[Software Interface 4](#_Toc171898176)

[System Features 6](#_Toc171898177)

[Description 6](#_Toc171898178)

[Stimulus/Response Sequences 6](#_Toc171898179)

[Functional Requirements 6](#_Toc171898180)

[Nonfunctional Requirements 7](#_Toc171898181)

[Performance Requirements 7](#_Toc171898182)

[Scalability 7](#_Toc171898183)

[System and Maintenance 7](#_Toc171898184)

[Security 8](#_Toc171898185)

# Description

The system will provide support to Moodle educators through a plugin that captures and displays students’ performance on evaluation material created by the educator for the course on the Moodle platform.

## Operating Environment

The system can operate on any hardware platform that can support the Moodle LMS 3.11 version. This includes operating systems Windows versions 7,8, and 10, Mac OS X, and Linux Ubuntu distribution. The system will be using MariaDB 10.3 as its database server along with Apache 2.4.6 as the web server.

## Assumptions and Dependencies

* Assumptions
  + Moodle API can be used to rearrange the information Moodle already provides.
* Dependencies
  + Moodle Local Instance
  + PHP version
  + XAMPP for image management.

For this plug-in we assume that Moodle API will help us fulfill meet client expectations. API like data and page management are expected to be used for the functionality. The biggest change desired is the incorporation of graphs to present the information of the course to the educator, but if the API mentioned does not provide what is expected to implement such change, another approach will need to be made.

# External Interface Requirements

## User Interface

The activity tracker will allow the educator to monitor student’s activities. In the first window, the educator will have a general view of how the students are doing in the class in general while also have the option of selecting an activity. If selected, they will be taken to a window where they can track each student’s performance in that activity. Back in the first window, the teacher also has the possibility of selecting a student instead of an activity. If selected the teacher will then be taken to a window displaying the student’s performance in general given the teacher the option of tracking his participation in each activity.

## Hardware Interface

Development will be on separate servers regarding the web front end and the database as it is easier to modify to specifications that way. Because of the potential of the server’s size, development work will be done as if the university will be implementing the features for its community which is why the options for an 8GB memory server was made. A 2GHz dual core processor will be used and, for the disk space, the source code requires approximately 200MB, but additional space will be allocated to be able to save course content which is why the established target will be for a minimum of 5GB of space.

## Software Interface

The activity tracker will provide a specific UI tailored to facilitate task management for educators, heavily depending on visual queues. The sole dependency will be on Moodle Core API, relying on services such as the Message API to relay to students when they've successfully completed or missed an activity, the Preference API from the educator’s perspective, to highlight students who are not completing the specified tasks.

# System Features

## Description

The activity tracker currently established at Moodle UPRM will be overhauled to show graphics that represent a student’s performance. These graphics can be sorted by student or by activity.

## Stimulus/Response Sequences

|  |  |
| --- | --- |
| **Stimulus** | **Response** |
| Educator requests student data progress. | System creates data to be exported. |
| Educator selects specific activity to get information from. | System creates data of the given activity. |
| Educator selects specific student to get information from. | System creates data of the given activity. |

# Functional Requirements

1. The plugin will showcase the activity completion for each activity in the course.
2. The plugin will let the educator view the information only of a given student.
3. The educator can filter the information they want to see by activity type in the Activity Tracker’s Main Logger page. This only applies to activity types ‘assign’, ‘book’, ‘page’, and ‘quiz’.
4. The educator can filter the information they want to see by activity type in the Activity Tracker’s Student Logger page. This only applies to activity types ‘assign’, ‘book’, ‘page’, and ‘quiz’.
5. The educator can choose the activities they want to see information about in the Activity Tracker’s Main Logger page. Only applies to activity types specified in requirement c.
6. The educator can choose which activities they want to see information about in the Activity Tracker’s Student Logger page. Only applies to activity types specified in requirement d.

# Nonfunctional Requirements

## Performance Requirements

### Scalability

1. Generated plugins must comply with Moodle Plugin development standards and must not surpass a limit of 250Mb in size.

### System and Maintenance

1. The use of and SQL database server from PostgreSQL (v9.6), MariaDB (v.10.2.29), My SQL (v5.7), Microsoft SQL Server (2012 release)
2. PHP Required Minimum version is PHP 7.2.0. Support extends of up to version 7.4.

### Security

1. Minimum Secure Distribution is Moodle 3.5.
2. All users must be properly authenticated at the beginning of a session with regards to Moodle.
3. Upon initial request to the server for the JSON files in reference to Google Apps Scripts, users must be properly authenticated through a Google Account.
4. Before any request to the Database, verify the session tokens and
5. Students must be granted modification capabilities for the scheduler in regard to the activities of 1 specific course by a teacher or administrator.
6. Educators must be granted modification capabilities for the activity tracker in order to overrule custom completion rules of an assignment.
7. The system must implement HTTPS security to protect from Main in the Middle (MITM)
8. The system must verify user all input before generating queries to update or manage information in the database; this prevents any type of injection.