Simple date handling in Moodle

Tsedey Terefe
Abstract

With an overview how Moodle plugin works, this paper aims to focus on those areas where open source software impact has on learning management system software which is distributed for free. One of the essential goal of this project is to develop plugin tool for Moodle, which aid users by providing less interaction time while adjusting date related tasks mainly with courses and assignments, which currently in the Moodle system is accomplished by a lot of steps. The project concentrated on two application functions, namely assignment_daterollover and course_daterollover. These two application plugin features in Moodle learning environment allows instructors to automatically update all date items in their course at the beginning of each semester. Date adjustments can easily be made to each assignments in a course through one centralized screen, rather than having to go into each individual learning activity. The two plugin tool options are one which will change the course start date by which all assignments can be simultaneously set forward by a specified number of days which is called course_daterollover and the other option is to adjust individual assignment items within a course which is called assignment_daterollover. Through the methods, the number of steps (clicks) has been counted and compared with the existing date adjustment method in Moodle for course and assignments, to evaluate and identity UI limitations Heuristic evaluation method is used. The plugin is implemented using block-plugin, because of its appearance to simply reside the content on the right or left side of each main course page, which also gives users the right to turn it on or off if they don't want to use it. In addition, the tool can be used as an option associated with the existing Moodle date adjustment system from the point of view of having a different design. The result is a user-friendly plugin tool interface for the stated two purposes which have less number of clicks than the Moodle date adjustment system for courses and assignments. The plugin tool can be used by anyone since it has been published on Github. As a conclusion, Moodle with its cost-effectiveness can serve a lot of users around the world.

Keywords: PHP, Moodle, plugin, open source, LMS, assignment daterollover, course daterollover.
Acknowledgment
For the continues help and advice, my gratitude goes to Magnus Eriksson.
# Table of Contents

Abstract .............................................................................................................. iv  
Acknowledgment ............................................................................................. v  
Terminology ....................................................................................................... viii

1 Introduction ..................................................................................................... 1  
1.1 Background and problem motivation ....................................................... 3  
1.2 Overall aim ............................................................................................... 3  
1.3 Scope ......................................................................................................... 3  
1.4 Detailed problem statement ...................................................................... 3  
1.5 Outline ....................................................................................................... 4  
1.6 Contributions ............................................................................................ 4

2 Moodle ........................................................................................................... 5
  2.1 Open source Licenses ............................................................................... 6  
  2.1.1 Strong Copyleft Licenses .................................................................. 7  
  2.1.2 Weak Copyleft Licenses .................................................................. 7  
  2.1.3 Non Copyleft Licenses ..................................................................... 8  
  2.2 Moodle Core ........................................................................................... 8  
  2.3 Moodle API ............................................................................................ 9  
  2.4 Plugin types ............................................................................................ 10  
  2.5 Capabilities ............................................................................................. 11  
  2.6 Definition of terms and abbreviations ................................................... 15

3 Methodology .................................................................................................. 16
  3.1 Analysis .................................................................................................... 16  
  3.1.1 Analysis of the project design ............................................................ 17  
  3.2 System Requirements ............................................................................ 17  
  3.3 Prototype ................................................................................................ 18  
  3.4 Number of Clicks .................................................................................... 18  
  3.5 Usability test method ............................................................................. 18  
  3.6 Response test method ............................................................................ 19  
  3.7 Revision control system ......................................................................... 19

4 Implementation ............................................................................................... 21
  4.1 Prototyping .............................................................................................. 22  
  4.2 Install a plugin in Moodle ........................................................................ 23  
  4.2.1 Administrator .................................................................................... 25  
  4.2.2 Mathematical formula ....................................................................... 28

5 Results ............................................................................................................ 31
  5.1 Response Test Result ............................................................................. 31  
  5.2 Number of clicks ..................................................................................... 32

6 Discussion ...................................................................................................... 35
## Terminology

### Abbreviations

<table>
<thead>
<tr>
<th>Abbreviation</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Moodle</td>
<td>Modular Object-Oriented Dynamic Learning Environment</td>
</tr>
<tr>
<td>LMS</td>
<td>Learning Management System.</td>
</tr>
<tr>
<td>CMS</td>
<td>Course Management System</td>
</tr>
<tr>
<td>VLE</td>
<td>Virtual Learning Environment</td>
</tr>
<tr>
<td>LDAP</td>
<td>Lightweight Directory Access Protocol</td>
</tr>
<tr>
<td>LTI</td>
<td>Learning Tool Interoperability</td>
</tr>
<tr>
<td>SCORM</td>
<td>Sharable Content Object Reference Model</td>
</tr>
<tr>
<td>GPL</td>
<td>General Public License</td>
</tr>
<tr>
<td>ADL</td>
<td>Advanced Distributed Learning</td>
</tr>
<tr>
<td>RCS</td>
<td>Revision Control System</td>
</tr>
<tr>
<td>RDMS</td>
<td>Relational Database Management System</td>
</tr>
</tbody>
</table>
1 Introduction

The term 'Open Source' brings up something that can be freely used, modified, and shared because it is meant to be publicly accessible by anyone. Meanwhile open source software development appoints the source code availability for modification as well as improvement by anyone at no charge.

Open source effect on software development is huge, the chance it gives to develop on the existing base software elements, makes developers to get educational benefit by working on the source code. Furthermore the right to make changes and its visibility, has made it easy to be imported to users target platform. The impact of open source software on education has evolved from the beginning of digitization of education through online courses, educational portals and also virtual universities. To meet the global accessible requirement goal, most universities rely on virtual learning environment which has a big load on finances. Another option is adaptation to the open source software development which is more cost-effective way of meeting the global accessibility goal. The advantage of having an open source learning management system gives continuous improvement on the software products along with contributing freely to projects by adding new tools by the open source developers. By this we come to the aim of this project development, which is made on one of the open source learning management system, Moodle[1].

Moodle (Modular Object-Oriented Dynamic Learning Environment)[1], is learning management system software which is distributed for free. It was originally designed to create an online courses, with it first version released on August 20, 2002 under General Public License (GPL) with its documentation of wiki content. With its rapid growth and acceptance it has now 64,945 registered sites around 234 countries through out the world[1].
“Every product that is used by someone has a user experience.”[2], this cited line shows how people feelings and products are associated, meaning getting a pleasure and satisfaction when using the product, like when opening, closing, updating and also looking at it. By this we mean that how does Moodle interaction design optimize user interaction mainly for instructors or course creators. Considering what might help instructors the way they currently do the task, we need to identify the weakness and strength of Moodle Learning Management system(LMS), interactive system. Then we can understand the need for this project as something usable. For this project we have identified one inefficiency that requires improvement that is currently done by number of steps for the basic task to be accomplished. The task is setting forward course start date and assignment items that include; due dates, cutoff date and allowsubmissionfrom, into a new academic calender term with a less interaction step or in other word a minimal step. Taking into account all that is designed using this project (assignment_daterollover and course_daterollover block plugins), is already in use in the current Moodle system but it is carried out with a number of steps (clicks) to adjust course and assignments for upcoming terms, the reason for developing such plugin comes into the table so that assignment_daterollover and course_daterollover blocks will provide a better user interaction.

Adjusting dates on Mid Sweden University website have been done in the old learning management system WebCT easily; but now WebCT has been entirely replaced by the new Learning Management system(LMS) Moodle, where such capability was missing from it by which instructors automatically update all dates for the items in their course through one centralized screen, rather than having to go into each individual learning activity.

In this project we find user centered approach in which the end users using the product being at the core gets less interaction time with the system. The project design code has met users requirements correctly and the design will be modified according to the evaluation feedback and a new set of prototypes will be developed and then evaluated. The other approach used in this project for evaluation is usability testing which will be discussed in chapter 3. Various authors agree that usability is a weak point in an open source software. With the new 'assignment_daterollover' and 'course_daterollover' plug-in, it will be possible that instructors can perform

- Changing assignment hand in dates in one centralized place.
- Change course start-date that will automatically change each assignment date related items without going into each assignment activity.

The final scene of 'assignment_daterollover' is tidying up the old assignment duedate, cutoffdate, allowsubmissionfrom for the new semester according to
the selected date in a centralized screen for a single assignment whereas the final scene of 'course_daterollover' will be tidying up the old course start date according to the selected date in a centralized screen for a new semester but it will also set forward all assignments hand in date within that particular course.

1.1 Background and problem motivation

To provide additional feature to Moodle LMS, which updates all dates for course and assignments. At the end of each academic term, instructors want to process an existing course material for the new upcoming term. The Date Rollover feature is not available now in Moodle where instructors can automatically update all dates for the items in their course at once. There is a need to have a tool for assignments, and assessments, that can simultaneously set forward by a specified number of days so that instructors can then make date adjustments to individual items through a centralized screen, that is easy to navigate. To adjust all dates in a course, for example if the course start-date was 2014/02/22 and if a user wishes to move it into a new term into 2014/05/21, all assignment dates for the course will be set forward 200 days. So there is no need to adjust dates for each learning activity since this new feature will enable to adjust it at one time.

1.2 Overall aim

The project's overall aim is to investigate and add an additional feature into Moodle, by focusing on a better interaction system which is a less steps to adjust all course item dates and also by highlighting the minimal time wastage when the task is done. It is well known that, it is impossible to create a user experience, but it is possible to create the design feature that can invoke the sensual experience.

1.3 Scope

In this paper, the reader will have an introduction to Moodle, a closer look at the most commonly used plugin types and an overall understanding on how to develop a block plugin and in depth look on the assignment_daterollover and course_daterollover plugins.

1.4 Detailed problem statement

The central concern of this project besides building an extension for Moodle is to develop a usable product, which means easy and effective to use while providing enjoyable user experience. By identifying the specific weakness in Moodle and its poorly designed date adjustment we could find that the problematic areas are:

- it is inefficient, requires a number of steps for a basic task.
It is not plain what to do, for first time users: the instructions are generated by the system.

It is confusing.

1.5 Outline

Chapter 2 describes the Moodle core API, focusing on the tools that will help when developing Moodle plugin scripts, and drawing attention to their functions so that developers will determine which API to use in each circumstance. It also describes the open-source licenses in particular about the copyleft licenses and lastly the different Moodle plugin types, which enables additional functionality to Moodle. Chapter 3 discusses about the chosen method used to accomplish this project, and the needed system requirements. In chapter 4, the paper describes how the method is implemented and with the result shown in chapter 5. Finally conclusion based on the result obtained will be discussed by showing some ways that could be improved in the future.

1.6 Contributions

Magnus Eriksson has a substantial contribution to the concept and design of this project and also for the interpenetration of the project into a real time work. I have conducted the entire php code which enables the 'assignment_daterollover' and 'course_daterollover' Moodle - plugin to make changes to the course item dates.
2 Moodle

Moodle with the acronym for Modular Object-Oriented Dynamic Learning Environment, is the most widely used learning platform first developed by Australian named Martin Dougiamas, then continued to be developed by Australian project development team, with a worldwide users which now exceeds 65 million. Moodle is an online Open Source, interactive Course Management System (CMS), is also known as Learning Management System (LMS) and Virtual Learning Environment (VLE). Moodle is created to provide a personalized learning environment and deliver a web based content for courses and course related sections a technology lesson in a manner where traditional courses are given but moved it into an online elements with an option to give from a range of a completely online courses or with a limited in-person interaction to courses given on-campus to provide a secure and integrated system [1]. With the capability to real-time communication tools, it is extended with assignments, quizzes, grades, certification, manage course and course contents, social and collaborative learning in an engaging manner, by uploading course materials such as audio, video, word documents and a link into other sites.

Moodle with its ongoing advancements is an open source software, under GNU GPL (GNU General Public License )[3]. As a software which works as an open source, anyone can copy, use and modify it for any project target, without any licensing fee by agreeing to provide the source to others without removing the original license. More explanation about the open sources Licenses are given in chapter 2.

Moodle can be customized for an individual needs, developers can develop plugins to add specific functionalists according to their needs. Moodle has an interface which is easy to use, with it's drag and drop features for all users regardless of any background, and on the other-hand for developers there is an instruction to follow and later on to accept feedback from experts.

Moodle is used by institutions and individuals such as Universities, primary and high-schools, governments organizations , airlines, independent educators and many more which makes it scalable and flexible to any size. It has multilingual capability which breaks the linguistic limitation. It is not mandatory to register Moodle site on moodle.org, but registering allows the Moodle site to be on the mailing list to get notification about recent Moodle releases and security alerts.

Moodle has a dedicated full time developers and an international community which support the development.
Moodle as a standard System

Moodle has achieved international standards which are mentioned below.

✔ **Open source:** under the GNU License a free open source software which can be modified and distributed[3].

✔ **IMS certified:** The Learning tool interoperability (LTI) certification is a technical standard for learning applications with a concept of establishing a standard way of integrating remotely hosted tools with LMS. In other words to allow seamless connection of web based tools to LMS. For example, connecting chat application or virtual labs to the learning management system platform[4].

✔ **SCORM complaint:** Sharable Content Object Reference Model (SCORM) is a specification for e-learning software products. Its an independent third party testing. SCORM enables interoperability between e-learning content and LMS. It clarify the act between the run time environment and the client side content by defining a specification and standards about content data models and protocols across systems by promoting the reusable and interoperability of LMS including how content should be packed into a transferable ZIP file which is Package Interchange Format. In short one content which has been published on one LMS should be able to be used in any other LMS if it is SCORM certified. Moodle complies with SCORM 1.2 specification version[5].

✔ **Open Badges:** is a standard to recognize and verify learning, which is an online representation of the accomplishments one earn. It is integrated in Moodle so any University or school can create and issue badges to learners so that learners can show the digital badges they earn across the web[6].

Figure 2.1 Icons for the international standards that are compliant with Moodle[7].

2.1 **Open source Licenses**

Licenses that abide by the work of open source definition are called open source licenses. It allows free use, modification and share. To be approved by the open source initiative, a license must go through a review process and ac-
According to the free software definition, open-source license must fulfill the following freedoms:

1. to run the program for any purpose.
2. give access to the source code so that it can be used according to the needs and can be studied.
3. give access to the source code to improve the program and release it to the public.
4. to redistribute copies to help others.

Popular open-source initiative licenses are:

- Apache License 2.0
- BSD 3-Clause "New" or "Revised" license
- BSD 2-Clause "Simplified" or "FreeBSD" license
- GNU General Public License (GPL)
- GNU Library or "Lesser" General Public License (LGPL)
- MIT license
- Mozilla Public License 2.0
- Common Development and Distribution License
- Eclipse Public License

2.1.1 Strong Copyleft Licenses

Provisions are imposed into all works that will be copied and distributed, any derivative work have to be distributed under the original open-source license. In strong copyleft, the first creator of the work will have the most rights. The code under strong copyleft cannot be linked with code under a non-strong copyleft. Examples include GNU General Public License v2 and v3, Java, Linux kernel and Sleepycat license[8].

2.1.2 Weak Copyleft Licenses

Weak copyleft license allow free distribution, use and sell of the code. Weak copyleft licenses require that any software that descend from it should be distributed under weak copy left license. The derivative work can be distributed under a different license like non open source code. Examples are GNU Lesser General Public License (GNU LGPL), Version 2.0 of the Artistic License, Mozilla public License (MPL) and GPL with an Exemption Clause[8].
2.1.3 Non Copyleft Licenses

Non Copyleft Licenses which is also known as BSD licenses are licenses that allow copy, modification and use of the source code which also considers selling the program for any type of use. Non copyleft licenses don't require the original license to be on the derived work. It allows to use the software to be incorporated with other software’s and use it in a basis for proprietary software. Non Copyleft licenses after being distributed under proprietary software, don't permit to remove the original copyright notice and bring any legal battle to the software damage[8].

![Copy-left symbol](image1)
![GNU GPL icon](image2)

Figure 2.2 From left to right: Copy-left Licenses icon[9] and GNU GPL icon[10].

Moodle is organized as an application core, where several other plugins surrounding it to give an additional functionality, Moodle can be customized without the modification of the core libraries, since that will create a problem when upgrading.

2.2 Moodle Core

All the infrastructures that are important to build the LMS are incorporated in the Moodle core, that a plugin needs to work on. These are

**Course and activities:** A Moodle course is a sequence of activities and resources grouped into sections. Courses are organized into a directory called category within Moodle site.

**User:** anyone who uses Moodle system is called user. Participation in a course depend on enrollment with a given system role as student or teacher.

**Course enrollment:** according to the role given as student and as a teacher.
2.3 Moodle API

Moodle has core APIs that are used between the components to interact with each-other and they are important when writing plug-ins.

The most used APIs are shown from the Figure 2.3 above and below are the description for each of the most critical and nearly used by every Moodle plugin.

1. Access API: it gives functions so that the developer can determine what the plugin user is allowed to do and it also allows the modules to extend with new capability. Entities in Moodle which are system, users, course categories, courses, modules and blocks are represented by contexts. The contexts are organized in a hierarchy called context tree as it is seen from Figure 2.6 below. For each logged-in user or guest a role capability is set, the role assignment is a representation of the ability to do something.

2. Data manipulation API: it gives the function to access the Moodle database.

3. File API: it is used to manage the files system stored by Moodle. It is related with Repository API which lets users to get contents into the Moodle system from external repositories.

4. Logging API: is used to add log and specify the way they are showed in reports.

5. Navigation API: the navigation system is defined by the navigation API, which shows how the user browse to the other pages. The navigation system in Moodle is contextual to the page the user is viewing.
6. **Page API:** defines how the current page is displayed to the user.

7. **Form API:** html forms and elements are configured and are created with the form API.

8. **Output API:** contains the meta data that are used by the renderer and represent elements to be output.

9. **String API:** is used by user interface to get the language strings.

10. **upgrade API:** new installations and upgrades are tracked here.

### 2.4 Plugin types

There are Moodle functionalists which are common to all plugins those are installation, permission configurations and upgrade.

Moodle plugin mainly holds a directory containing files with php, styles sheet, JavaScript as well icon images so that Moodle core finds the entry point in this directories to communicate most commonly the file called *lib.php*. Plugin has a type and name, by the naming system called "Frankenstyle", which gives the naming system by combining plugin type and plugin name, 'plugin-type_plugin-in-name', for this project the Frankenstyle name would be, 'block_assignment_daterollover' where 'block' indicates the plugin type and 'assignment_daterollover' indicates the plugin name, Figure 2.4 shows the Frankenstyle naming system and folder use in Moodle.

![Figure 2.4 example of plugin types](image)

Moodle requires a directory to store all the files that include; all the sites uploaded files, caches and session data. To develop our own plugin we have to create a directory with the appropriate files in it as mentioned on Table 4.1. and put it on the suitable plugin type directories, since placing the plugin directory in the wrong plugin type make our Main LMS site not to function properly.

### The basic plugin types are

**Activities and resources**: Activities include quizzes, assignment forums these are the components that make up a course and are important for the teaching and learning. Activities and resources are installed in the *mod* folder.
Blocks: its an interface that can be added into the sides of a page, They are added into the blocks folder.

Themes: the modality style of the site or a style for a particular course can be changed. They are added into the theme folder.

Language packs: to provide a varied forms of language.

Course formats: activities grouped into sections and the way they are structured is controlled by the course formats, it is added into course/format folder.

Authentication plugins: log-in control which are stored in Lightweight directory access protocol (LDAP). It is added on the auth folder.

Enrollment plugins: controls course enrollment, it is added on enrol folder.

Repository plugins: it is the way to get files into Moodle, files could be uploaded from different sources. It is added into repository folder.

Others: other plugins which are not basic but exist in the system includes text filters, admin reports, course reports, question types, web service, plagiarism detection. All are added into lib/moodlelib.php file.

As we have discussed above one of the common Moodle functionalists which is common to all plugins is permission configuration, which we will discuss in capabilities.

2.5 Capabilities

Site administrators configure which user should be given permission using capability. When functions gets the relevant context where the permission is checked, it performs

1. get context from the current opened module.
2. get roles a user has from the context
3. figure out permission for each role in current context.

Roles: are named set of permission given to a user. When one user logs into Moodle the user will have an authenticated user role in the system context, and the role applies to every place since the system is the higher hierarchy. In a course context, for one student the role may apply to the course and the module contexts within it.

Permissions: permission is defined according to the role given or in other words role defines permission for each capability[11]. As we can see from the
code example below from Figure 2.5; 'block/assignment_daterollover:change-date', teacher, editingteacher, coursecreater all have a role which allow them to change assignment dates, however students don't have ALLOW capability.

```php
defined('MOODLE_INTERNAL') || die();
$capabilities = array(
    'block/assignment_daterollover:changedate' => array(
        'captype' => 'write',
        'contextlevel' => CONTEXT_COURSE,
        'archetypes' => array(
            'teacher' => CAP_ALLOW,
            'editingteacher' => CAP_ALLOW,
            'coursecreater' => CAP_ALLOW,
            'manager' => CAP_ALLOW
        ),
    ),
    // New standard capability 'addinstance'.
    'block/assignment_daterollover:addinstance' => array(
        'captype' => 'write',
        'contextlevel' => CONTEXT_COURSE,
        'archetypes' => array(
            'editingteacher' => CAP_ALLOW,
            'manager' => CAP_ALLOW
        ),
    ),
'clonepermissionsfrom' => 'moodle/site:manageblocks'
);
```

Figure 2.5 example code from assignment_daterollover showing capability.

Generally capabilities have four types of permissions, which are

1. **CAP_INHERIT**
2. **CAP_ALLOW**
3. **CAP_PREVENT**
4. **CAP_PROHIBIT**

Capabilities with no permission will take or inherit a capability permission from a higher hierarchy level context. The description to the contexts are shown below in the Figure 2.6 and Table 2.1, in the description it can be viewed that the different capability permissions, where one context which doesn't have a defined permission takes a general higher context. For example if Activity module has no defined permission, it takes CONTEXT_COURSE.
The top level hierarchy is the system context which has category context inside. Category contexts are contexts that are created to hold courses and sometimes they may hold another categories also. Lets say in Information Technology category we may have different courses such as Java, networking I, networking II, TCP/IP and so on. As a nested form category context will have course context and each activity will have its own context. For figurative description about nested category see Figure 2.7 [12].

In Moodle a user may have different permission according to the roles assigned, a user in one course as a student may have another role when he/she becomes a teacher in another course so difference permission should be given to each user in different places, these places are called contexts. It should be noted that a capability could be overridden when it is given different permission in different contexts.

From the above four types of permissions mentioned, there is a slight difference between CAP_PREVENT and CAP_PROHIBIT. CAP_PREVENT allows overriding permission for example if a student can be made to have only a read-only capability by overriding mod/wiki:edit in that module context but CAP_PROHIBIT can not be overridden in sub-contexts. CAP_PROHIBIT can be used to totally disable users capability to post unnecessary posts in the forum.
Contexts with their area and level is shown in the following table.

<table>
<thead>
<tr>
<th>CONTEXT_NAME</th>
<th>CONTEXT AREA</th>
<th>CONTEXTLEVEL</th>
</tr>
</thead>
<tbody>
<tr>
<td>CONTEXT_SYSTEM</td>
<td>the whole site</td>
<td>10</td>
</tr>
<tr>
<td>CONTEXT_USER</td>
<td>another user</td>
<td>30</td>
</tr>
<tr>
<td>CONTEXT_COURSECAT</td>
<td>a course category</td>
<td>40</td>
</tr>
<tr>
<td>CONTEXT_COURSE</td>
<td>a course</td>
<td>50</td>
</tr>
<tr>
<td>CONTEXT_MODULE</td>
<td>an activity module</td>
<td>70</td>
</tr>
<tr>
<td>CONTEXT_BLOCK</td>
<td>a block</td>
<td>80</td>
</tr>
</tbody>
</table>

Table 2.1 Context list from General to specific[12].

As it is shown in Figure 2.7 the School system has different faculties which are Humanities, Science and so on. Under each Faculty there are departments. Under each departments there are a number of courses and each course has activities. Which clearly defines the meaning of context and its hierarchy.
2.6 Definition of terms and abbreviations

**Requirement** : a requirement is a statement about an intended product that specifies what it should do or how it should preform[13].

**Date rollover**: a feature in the course settings where instructors can move forward dates for course items by a number of days for a new semester, instructor can also change dates by editing individual dates.

**Agile software development** is a group of software development methods based on iterative and incremental development, where requirements and solutions evolve through collaboration between self-organizing, cross-functional teams[14].

**GNU General Public License** is a free, Strong copyleft license for software and other kinds of works. GNU General Public License is intended to guarantee freedom to share and change all versions of a program to make sure it remains free software for all its users. It requires the derived work to be distributed under the original license[3].

**Copyleft License**: Copyleft is a type of license that attempts to ensure that the public retains the freedom to use, modify, extend and redistribute a creative work and all derivative works[15].

**Usability testing**: an evaluation approach to evaluation that involves measuring users' performance and evaluation their satisfaction with the system in question on certain tasks in a laboratory setting[13].

A **role** is an identification of user's status in contexts[12].

A **capability** is a description which determines what the current use allowed to do. For example, *block/assignment_daterollover:changedate* is a capability[12].

A **permission** is the capability given to do a specific task for example, allow or prevent[12].

A **context** is a space in Moodle where it contains other contexts within it for example category is a context where it contain courses and so on[12].

**Frankenstyle**: refers to the naming convention that is used identify a Moodle plugin based on the type of plugin and its name[16].
3 Methodology

3.1 Analysis

As the aim of this design project is to add a new feature to Moodle system, there were an initial set of requirement in-order the plugin to work:

1. **Identifying needs:** this phase is about understanding users, their work and the context of their work while they update course start and assignment end dates and to explore more for sufficient detail[13].

2. **Requirement analysis:** from the needs identified to set of stable requirements by analyzing the initial set of requirements. In this phase, the requirement statement were identified about the plugin product, which specifies how it should perform. Two approaches has been identified [13].
   - change of course start date which in-turn changes assignment date related items by x number of days at one time.
   - Adjust individual dates for assignment items.

3. **Data gathering for requirements:** The aim of data gathering is to collect sufficient and relevant data to set the requirements. Even-though the initial requirement is to build a plugin for Moodle, data gathering is needed to clarify and expand the initial requirements. There is a need to gather how instructors now perform daterollover task and the principle for performing such a basic task. Among the common forms of data gathering in this project the techniques used are observation how instructors perform and studying Moodle documentation. The observation technique was accomplished the fact that I was given 'teacher' role in one course from Miun University, supervised by Magnus Eriksson so that I can study how Moodle handles activities when the user is logged in as a teacher in a real time system.

   Interpretation of facts that will inspire the design was taken with Agile software development. Since Agile software development method allow iterative development and adaptive planning, flexible methods were employed because this project has changed requirements even late in development stage, to comply with the needs. Throughout the project I have had weekly meetings with my supervisor for 30 minutes, so there were changes in the requirements according to the feedback given[14].
3.1.1 Analysis of the project design

<table>
<thead>
<tr>
<th>The power of appearance</th>
<th>What people see influences, how they interact and feel. Simple html form implementation is used.</th>
</tr>
</thead>
<tbody>
<tr>
<td>Accessibility</td>
<td>The block content is separate from navigational elements</td>
</tr>
<tr>
<td>Navigational depth</td>
<td>The maximum number of clicks it takes to execute is only one.</td>
</tr>
<tr>
<td>Security</td>
<td>The database is appropriately safeguarded against an authorized access.</td>
</tr>
<tr>
<td>Block loading time for first time installment</td>
<td>Moodle site loads 'assignment_daterollover' and 'course_daterollover' block quickly in less than a minute</td>
</tr>
</tbody>
</table>

Table 3.1. analysis of the program.

3.2 System Requirements

Moodle can be run on different operating systems such as Windows, Mac, Linux and also Mobiles devices that support iphone Operating system (iOS) and Android.

A Moodle installation comprises the Moodle code executing in a PHP-capable web server; a database managed by MySQL, PostgreSQL, MariaDB or Oracle; and a file storage for uploaded and generated files (the moodledata folder), so it can work on any computer that can run PHP.

**Moodle requires a**

- free disk space of 160MB
Simple data handling in Moodle – assignment and course daterollover

Tsedey Terefe

2014-04-04

- backups to keep backups
- 256MB minimum memory to concurrently support other users.
- Operating system such as Linux and windows, Linux being the optimal choice for Moodle.
- Apache web server which can run PHP version 5.3.3
- database such as MySQL min version 5.1.33 and PostgreSQL min version 8.3, MSSQL min version 9.0, MariaDB min Version 5.3.5. Oracle is not fully supported.

This project has been done on:

Windows 8 and Linux Virtual Box.

- The Linux Virtual-box has 60 GB storage size.
- Windows 8, which has 8 GB RAM, 64 bit operating system processor, 1.60 GHz and 500 GB Hard disk drive.
- Moodle 2.6 version installed for development
- Moodle 2.2 minimum requirement for the plugin to work
- PHP script on MySQL database, Apache local web server

3.3 Prototype

Paper based mockup of the design interface was used, to communicate and adjust the interface with my supervisor, see Appendix B.

3.4 Number of Clicks

The number of clicks in the current Moodle standard date adjustment for both Assignment and course.

3.5 Usability test method

To determine the level of the block usability, instructors will perform and evaluate the project by testing it on course and assignments. The evaluation method is heuristic evaluation that will help to identify problem related with the user interface design. It is useful to test the efficiency on how many steps it requires for the complete task, how many mistakes do instructor make when performing the task to determine the accuracy and it helps to determine product acceptance.
The heuristic evaluation check-list has been sent to four instructors since the result of this project will benefit mostly instructors. A brief description about the questionnaire used in the evaluation can be seen from Appendix C.

### 3.6 Response test method

The criteria that will be used to test the response time will be the number of assignments against the total amount of time it takes to accomplish the task. In order to determine execution time, the measurement starts the moment the button is pressed until it finish and redirects the page, this method enables to determine the time it takes the PHP script to execute the task.

1. Click the button

2. take the current time in unix timestamp in microseconds using microtime() method and assign it into a new variable.

3. Start updating the database tables

4. finish updating the database tables

5. take a new value of the current time in unix timestamp using microtime() method and save it in a new variable.

6. Subtract the value obtained in number 5 from the value obtained in number 2.

7. output the total time it takes from the value obtained in number 6.

### 3.7 Revision control system

Revision control system (RCS), is a version control system in program development, it will keep the system software and configurations consistent throughout many versions. RCS is a software tool that is used to manoeuvre the task of keeping the software system which had many versions consistent. It manages revisions of source code programs[17].

Currently there are two strategic ways that are used by the revision control systems, centralized and distributed. In the centralized mode, there exists a central server where everyone uploads and downloads to changes. It is an ideal way of doing a group project with a requirement to be connected with the server whenever there is a need to pull and push a change into the server. On the other-hand with the distributed system, someone may pull from another person the source code or the file and make the necessary changes to it and tell others or group members to pull the changes made from him/her. Using a distributed system is ideal for groups working in disconnected environments [18].
In this project a distributed revision control system is used and published using github server so that anyone can revise it.

**The method's possible weaknesses**

Even if using block method meets the initial requirements, which says to accomplish the tasks using one centralized screen, it has one possible weakness if the need to have more HTML forms arise, since it is impossible to expand horizontally and vertical expansion doesn't seem to be ideal design attraction. This can be viewed from Implementation Chapter Figure 4.2 and Figure 4.3.
4 Implementation

Analysis of the code shows the method by which block plugin in Moodle will attempt to resolve the problem of having to do the task from multiple steps into one step. To implement that has been identified via the analysis method in chapter three of this project, the approach requires two completely separate plugin blocks in-order to change one for course start date and the other with individual assignment due-date, cutoff and allowsubmission from date, that means if an instructor wants to change a course start date which simultaneously set forward assignment item dates by a specified number of days he/she should add block_course_daterollover, which adjust all dates at once or if the instructors intention is to adjust individual date of assignments due date, allowsubmission date and cutoff date then the other block, block_assignment_daterollover, should be added. To implement this project, the user should have admin privilege to put the two blocks under the following path www/moodle/blocks in Linux and htdocs/moodle/blocks in windows. Then a successful installation will show a success message as it can be seen on Figure 4.1, if the plugin is saved in a wrong directory the system will show up error.

Figure 4.1 block first installation.
4.1 Prototyping

Having all the necessary information collected about what the Moodle plugin should and shouldn't do, there is a need to build a prototype. I have consulted Jan Erik about the final product should look like, the prototype was made in a paper based outline or mock-up, which explains how the interface should look like. The mock-up was an important communication device when I wanted to discuss the ideas with my supervisor. The mock-up serves to answer some vague requirements, it shows also what functions and buttons it should have; including positioning, labelling and the overall shape. The mock-up can be seen from Appendix B.

Figure 4.2 an overview block_course_daterollover.

Figure 4.3 an overview of block_assignment_daterollover.
4.2 **Install a plugin in Moodle**

Moodle provides a capability to add plugins which will work together with the core system in order to have additional functionality which the standard and default plugins doesn't offer. It has an advantage to any developer to make his/her own plugin to a particular need. The installation consists of three parts.

1. **Main application code**: it is located under at www/moodle directory in Linux or /htdocs/moodle in windows. This directory should not be writable by the web server. According to the type of plugin as discussed in Theory section 2.4, plugin types , 'assignmentdaterollover' plugin will belong to blocks and is located in www/moodle/blocks or /htdocs/moodle/blocks.

2. **Moodledate directory**: it's writable by the web server. Moodle generated and uploaded files are stored in this directory.

3. **database course materials**: which resides in Moodle database, it is supported by RDMS(relational database management system). Moodle handles database tables by adding a prefix 'mdl' to all table names, which will give a possibility to use the database with different applications.

Under the root directory, there is a file called config.php, moodle/config.php, which holds the overall configuration information about the installation which has been described above in three points. As it can be seen from Figure 4.4, below the following file is created when Moodle is installed for the first time.

```
<?php // Moodle configuration file

unset($CFG);
global $CFG;
$CFG = new stdClass();

$CFG->dbtype = 'mysql';
$CFG->dblibrary = 'native';
$CFG->dbhost = 'localhost';
$CFG->dbname = 'moodle';
$CFG->dbuser = 'moodleuser';
$CFG->dbpass = 'moodledb';
$CFG->prefix = 'mdl_';
$CFG->dboptions = array ( 
    'dbpersistence' => 0,
    'dbport' => '',
    'dbsocket' => '',
); 

$CFG->wwwroot = 'http://localhost/moodle';
$CFG->dataroot = '/var/moodledata';
$CFG->admin = 'admin';

$CFG->directorypermissions = 8777;
require_once(dirname(__FILE__) . '/lib/setup.php');
```

Figure 4.4 configuration information
Context usage

One of the tools used to give access privileges to users is context, as it has been discussed in the theory part in detail, in Moodle context is a space where roles are assigned. Roles identifies user's status and are used to define list of permissions, in other word it gives permission for different actions.

The whole point of describing context is to show that unauthorized persons cannot make any changes into assignments or courses since permission are given according to the role given in each context.

Block Location

Where the block should appear will be determined by the context and users permission.

- **Original block location**: block information as to where originally created.

- **Display on page types**: depending on the user permissions it allows the context that the block can appear in.

- **Default region**: option for right or left position

- **Default weight**: when there are other blocks in that column, it will position the block according to the numbers and element given like. top=-10, bottom=10, 0=neutral.

The block name is “block_assignment_daterollover”, with the frankenstyle naming system[16], which shows 'plugin type_ plugin name'. To refer to Moodle home directory, path to Moodle files and directories always start with a slash. In order to have a working Moodle block there are four basic php files, where they will be placed under blocks directory. The blocks directory exists under the installed Moodle, /path/to/moodle. The examples given below are according to the **assignment_daterollover** block.

Those four basic files are:

1. block_assignment_daterollover.php
2. db/access.php
3. lang/en/block_assignment_daterollover.php
4. version.php
In addition there are two more files which are used for this project particular use.

- edit.php
- assignment_daterollover_form.php

<table>
<thead>
<tr>
<th>Name</th>
<th>explanation</th>
</tr>
</thead>
<tbody>
<tr>
<td>block_assignment_daterollover.php</td>
<td>used to display the block and determine the number of block appearance in one page</td>
</tr>
<tr>
<td>access.php</td>
<td>Holds functions to determine what the current user is allowed to do</td>
</tr>
<tr>
<td>lang/en/block_assignmentdaterollover.php</td>
<td>Holds language strings which help to display in the interface</td>
</tr>
<tr>
<td>version.php</td>
<td>Contains the version number of the plugin.</td>
</tr>
<tr>
<td>edit.php</td>
<td>Communicate with the database to retrieve and update user values</td>
</tr>
<tr>
<td>assignment_daterollover_form.php</td>
<td>Holds HTML elements like buttons, select options.</td>
</tr>
</tbody>
</table>

Table 4.1 PHP files from assignment_daterollover block with their use explanation.

### 4.2.1 Administrator

The site administration menu block is located at the left side of Moodle site, under Administrator block. It provides links to the site administration which is very helpful when developing a new plugin.
Figure 4.5 Administration option tool

- **Notifications**: this tool option is used when there is a need to upgrade a new plugin or to check any available updates. All plugins that require upgrade will automatically appear in this page.

**Advanced features**: contains the following main features that will be available if the site administrator enables them globally

- **Courses**: this option is used to add and edit courses which is used by those who have site administrator, course creator or manager roles.

- **Plugins**: using this option lists activities for each module, it is possible to see installed plug-ins and also it is possible to delete installed plug-ins.

- **Development**: this tool is essential to developers, when this tool is used cautious measures must be taken, since it can disrupt the site or it is better to use it on production sites.

**Debugging**: as it can be seen from Figure 4.5 above it is located under development tool, and it could be enabled to see error messages
when developing a plugin. It helps to diagnose problems since it shows the complete error message. Developers can go and set the debugging mode by clicking on the debugging link and choose from the drop down select option set as seen from Figure 4.6 as

Developer: extra Moodle debug messages for developers.

![Debugging tool option](image)

XMLDB Editor: it is a tool for making .xml files, and it is used to set up the database tables. Any database related task can be done from here after the option under the plugin is loaded. As it can be seen from Figure 4.7 below blocks/assignment_daterollover/db is where daterollover plugin database can be found.

![XMLDB Editor](image)
4.2.2 Mathematical formula

The following mathematical formula is used to calculate how many dates each assignment date related items should be set forwarded according to course start-date, in 'course_daterollover' block.

<table>
<thead>
<tr>
<th>Addition/Subtraction</th>
<th>Result</th>
</tr>
</thead>
<tbody>
<tr>
<td>(New course startdate) - (Old course startdate)</td>
<td>= date difference</td>
</tr>
<tr>
<td>(date difference ) + (old assignment duedate)</td>
<td>= new assignment duedate</td>
</tr>
</tbody>
</table>

insert (new assignment duedate ) into database 'assign' table column duedate

Table 4.2 Mathematical formula for adjust all dates

The above formula is used for duedate, cutoffdate and allowsubmissionfrom for Adjusting all dates in the 'course_daterollover'.

Example how the above formula changes dates.

Table for Adjust All Dates

<table>
<thead>
<tr>
<th>Title</th>
<th>Before change</th>
<th>After change</th>
</tr>
</thead>
<tbody>
<tr>
<td>Course start-date</td>
<td>01/01/14</td>
<td>01/08/14</td>
</tr>
<tr>
<td>Assignment 1 - duedate</td>
<td>02/01/14</td>
<td>02/08/14</td>
</tr>
<tr>
<td>Assignment 2 - duedate</td>
<td>02/15/14</td>
<td>02/22/14</td>
</tr>
<tr>
<td>Assignment 3 - duedate</td>
<td>02/22/14</td>
<td>03/01/14</td>
</tr>
<tr>
<td>Assignment 1 - Cutoff Date</td>
<td>03/01/14</td>
<td>03/08/14</td>
</tr>
<tr>
<td>Assignment 2 - Cutoff Date</td>
<td>03/15/14</td>
<td>03/22/14</td>
</tr>
<tr>
<td>Assignment 3 - Cutoff Date</td>
<td>03/22/14</td>
<td>03/29/14</td>
</tr>
<tr>
<td>Assignment 1- allowsubmissionfrom</td>
<td>01/01/14</td>
<td>01/08/14</td>
</tr>
<tr>
<td>Assignment 2- allowsubmissionfrom</td>
<td>01/15/14</td>
<td>01/08/14</td>
</tr>
<tr>
<td>Assignment 3- allowsubmissionfrom</td>
<td>01/22/14</td>
<td>01/08/14</td>
</tr>
</tbody>
</table>

Table 4.3 Adjust All Dates
Table For Adjust Individual Dates

<table>
<thead>
<tr>
<th>Title</th>
<th>Before change</th>
<th>After change</th>
</tr>
</thead>
<tbody>
<tr>
<td>Assignment 1 - duedate</td>
<td>02/01/14</td>
<td>05/08/14</td>
</tr>
<tr>
<td>Assignment 1 - Cutoff Date</td>
<td>03/01/14</td>
<td>05/10/14</td>
</tr>
<tr>
<td>Assignment 1 - allowsubmissionfrom</td>
<td>01/01/14</td>
<td>01/08/14</td>
</tr>
</tbody>
</table>

The example code displayed in Figure 4.8, is used to get the result that is shown above in the Table 4.3 Table for Adjust All dates, which iterate through each individual assignments that belong into the current course, it first adjusts items that belong to assign table; allowsubmissionfrom, duedate and cutoff date columns, then it adjusts events table, finally it adjusts the course table; startdate column.

The following line of code will get the course start-date with the current course id.

```php
$sql2=$DB->get_fieldset_select('course','startdate','id=?',array($courseid));
```

$sql2 value set the old course start-date and will be used to calculate how many days needs to be subtracted from the new inserted course start-date (New course startdate) - (Old course startdate), in-order to add the date difference into the whole assignment date related items.
Simple data handling in Moodle – assignment and course daterollover

Tsedey Terefe

2014-04-04

```php
$assignments = $DB->get_records('assign', array('course' => $courseid));

foreach ($assignments as $assignment)
{
    $record = new stdClass();
    $record->id = $assignment->id;
    $currenttime = time();

    $stradd = ($date->date) - ($sql1[0]);
    $record->due_date = ($assignment->due_date) + ($date->date) + ($sql1[0]);

    $record->allow_submission_from_date = ($assignment->allow_submission_from_date) + ($date->date) + ($sql1[0]);
    $record->cut_off_date = ($assignment->cut_off_date) + ($date->date) + ($sql1[0]);
    $record->time_modified = $currenttime;
    $DB->update_record('assign', $record);
}

$upcomingevents = $DB->get_records('event', array('courseid' => $courseid));

foreach ($upcomingevents as $upcomingevent)
{
    $eventrecord = new stdClass();
    $eventrecord->id = $upcomingevent->id;
    $currenttime = time();

    $eventrecord->time_start = ($upcomingevent->time_start) + ($date->date) + ($sql1[0]);
    $eventrecord->time_modified = $currenttime;
    $DB->update_record('event', $eventrecord);
}

$sas = $DB->get_records('course', array('id' => $courseid));

foreach ($sas as $a)
{
    $courserecord = new stdClass();
    $courserecord->id = $a->id;
    $courserecord->start_date = $data->date;
    $DB->update_record('course', $courserecord);
}
}
}
```

direct($CFG->wwwroot . '/course/view.php?id=' . $courseid, '', 0);

direct($CFG->wwwroot . '/course/view.php?id=' . $courseid, '', 0);

Figure 4.8. Example code
5 Results

5.1 Response Test Result

![Figure 5.1 performance measure relative to the number of assignments within a course](image)

As it can be seen from Figure 5.1, the performance measure was taken on a local server to test the response time in seconds against the total number of assignments in a course, by this we mean the time it takes the method to accomplish and affect three database tables with one click. The tables that will be affected are:

- course table
- assign table
- event table

The average response time for the above test is around 0.217623 secs.

Another test was also taken for, assignment daterollover block, to measure the response time for individual assignments, and the response time taken at different interval time ranges between 0.02- 0.25 secs.
5.2 Number of clicks

Figure 5.2 the number of clicks to adjust one assignment in the current Moodle system.
As we can see from Figure 5.2, in the current Moodle system, the number of clicks it takes to adjust a single assignment is seven, which is shown in five steps.

Figure 5.3 the number of clicks to adjust course start date in the current Moodle system.

From Figure 5.3, we can see that in the current Moodle system, it takes three clicks to adjust course start-date. Which means it takes ten clicks to adjust one course and one assignment, note that one course might have several assignments in it.

Whereas if course_daterollover block is used,

1. it handles date adjustments which are carried in two separate places into one.

2. by taking only two clicks it can adjust both settings from Figure 5.2 and Figure 5.3.

Totally it can save eight clicks if the course only has one assignment but it can save more than that if the course have more assignment in it. The good thing about course_daterollover is, it only takes two clicks no matter the number of assignments the course have.
To adjust individual assignment dates, in the current Moodle system it takes, seven clicks as it can be seen above from Figure 5.2, whereas in the assignment_daterollover block, it takes four clicks, to adjust one assignment as it can be seen from Figure 5.4.

Figure 5.4. the number of clicks to adjust individual assignments
6 Discussion

This report provides a basic understanding of what Moodle is, in which type of open source copyleft category it exists, the international standard compatibility it acquired, the most common core API that are very useful when writing a Moodle script, and how the project overall aim has been achieved with the method it used to solve the problem identified.

One of the very important social aspect of using Moodle is cost effectiveness in contrast to other LMS like Blackboard besides allowing different types of content to be uploaded for use by students and instructors with the different option and tools. On the ethical aspect users can pay for a proprietary software with a high cost if they are willing and can afford it, but for those who can't afford the high cost of proprietary software, let them choose the open source software. The other point on the ethical part would be users from a proprietary software will be locked up with their files which resides in the proprietary software since switching to other software will force them to pay more.

Some critics about Moodle include the problem if too much customization is done and there are many steps(clicks) to accomplish simple tasks such as upload a file, change a menu heading, change course and assignment dates. Keeping the discipline not to change the Moodle core API will save a great amount of time when developing for Moodle, since too many changes to the core code will impact the site when new Moodle version releases appear, that will affect the developers time since each time they upgrade someone has to do a tremendous amount of time even a couple of days to merge to the upgrade to the newer version. Controlling the core code changes will help to have the custom plugins and local plugins to work without break at any form of upgrade.

The problem stated at the beginning of this project was met since date adjustments are done through a centralized screen and also the steps it takes to adjust a course item now is less than the standard way Moodle uses now. Based on the result obtained from adjusting all dates for course items the response time determines, whenever the number of assignments increase the response time increases, and with the individual assignments the response times are all similar which doesn't show any circumstance affecting the result, over all the average response time from adjusting a course which intern adjusts assignment date related items is around 0.217623 secs, which almost is similar to the response time obtained from adjusting individual assignments which is between 0.02-0.25 secs, from this we can conclude that the time it takes to adjust all course date related items is almost similar to adjusting an individual assignment and it is reasonable since the response time is less than current Moodle course date adjustment system.
Usability test method was used with instructors so that they can determine the level of the block usability unfortunately it was impossible to gather relevant information via questionnaires and make usability test according to the Heuristic Evaluation sent to some instructors, since only one person was willing to respond and make usability test. The other method used to test response time was microtime() php function, which will start counting when button is clicked to update the database tables and stops counting when it finish updating. Revision control system method is also used to control the source code frequently for further revision, on Github.

My suggestion for future Moodle plugin development is, to make a local plugin that will incorporate what I have done in two blocks into one and also that will hide assignment materials that the students shouldn't see when they first access the course, until the date is set, this option was not made since it needs another third block development in this project, and it was not ideal for instructor to add three blocks at once to adjust a certain task. So for anyone who would like to contribute to the Moodle plugin development I would suggest that it's good to incorporate all of them together, by creating a link under Administrator block and put them all in one table. Better to peruse on with the basics of 'local plugin' development and apply the core Moodle API, I encourage to have a look at Appendix D for my future suggestion. Note that, it's hard to cooperate together the Moodle library which holds Moodle form(html form elements) and html_table together and also there is a problem with not having enough documentation for how things work around and some naming terminologies in Moodle are also confusing.

As a final point, a recommended way to add additional feature to Moodle is to create standard plugins such as activity module, blocks, enrol and so on but local plugins are appropriate for those tasks that don't fit the standard plugins.
References


[8] Nicholas A. Solter, Gerald Jelinek, and David Miner, OpenSolaris, Indianapolis, wiley publishing., 2009


[10] Creativecommons, “Under the following terms”, [http://creativecommons.org/licenses/by-sa/4.0/deed.en_US](http://creativecommons.org/licenses/by-sa/4.0/deed.en_US), Retrieved 2014-06-21


http://smartbear.com/products/qa-tools/what-is-agile-testing/,
Retrieved 2014-04-10


[18] University of Mennesota, “Revision Control Systems” ,
Appendix A: User Guide to add assignment_daterollover plugin into moodle site

Plugin-Name: Assignment Date Rollover

Course Date Rollover

Turn editing on either from Administration block or top of the course page

After editing is turned on add a block will appear

block_assignment_daterollover

block_course_daterollover
Appendix B: Mockup

Course Start date

Current Start Date: 2014-01-01

New Start Date: 2014-06-14

Review

Assignment

start: 2014-01-01
end: 2014-01-20

Quizz
## Appendix C: Heuristic Evaluation Questionnaire and Answers

### Heuristic Evaluation - A System Checklist

**Project Name:** assignment_daterollover and course_daterollover

**Evaluators Name:** Franked Loan

**Date:** 2014-06-09

<table>
<thead>
<tr>
<th>#</th>
<th>Review Checklist</th>
<th>Yes</th>
<th>No</th>
<th>N/A</th>
<th>Comments</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>1 Visibility of System Status</strong></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>1.1</td>
<td>Does the block title display when you add the block for the first time?</td>
<td>☐</td>
<td>☐</td>
<td>☒</td>
<td></td>
</tr>
<tr>
<td>1.2</td>
<td>Is the block clearly visible when surrounded by other blocks?</td>
<td>☒</td>
<td>☒</td>
<td>☐</td>
<td></td>
</tr>
<tr>
<td>1.3</td>
<td>Can the block be added into one page multiple times?</td>
<td>☐</td>
<td>☐</td>
<td>☒</td>
<td></td>
</tr>
<tr>
<td>1.4</td>
<td>If pop-up windows have been used to display error messages, will it allow the field in error?</td>
<td>☒</td>
<td>☒</td>
<td>☐</td>
<td></td>
</tr>
<tr>
<td>1.5</td>
<td>Are there any delay which will be more than 15 sec in the response time(system) and notice that the system is in progress?</td>
<td>☐</td>
<td>☒</td>
<td>☐</td>
<td></td>
</tr>
<tr>
<td>1.6</td>
<td>Are response times appropriate to the tasks?</td>
<td>☐</td>
<td>☒</td>
<td>☐</td>
<td></td>
</tr>
<tr>
<td>1.7</td>
<td>Does it take less than 1 sec for adjusting dates?</td>
<td>☐</td>
<td>☒</td>
<td>☐</td>
<td></td>
</tr>
<tr>
<td>1.8</td>
<td>Does it take more than 1 sec for tasks?</td>
<td>☑</td>
<td>☐</td>
<td>☐</td>
<td></td>
</tr>
<tr>
<td>1.9</td>
<td>Is the menu-naming terminology consistent with moodle’s task domain?</td>
<td>☐</td>
<td>☐</td>
<td>☒</td>
<td></td>
</tr>
<tr>
<td>1.10</td>
<td>If users must navigate between multiple screens, like moodle uses now to change assignment due-dates, cut-offs or allow submission from, does that make a better method approach to the navigation?</td>
<td>☐</td>
<td>☒</td>
<td>☐</td>
<td></td>
</tr>
</tbody>
</table>

2. **User should have the power to select by their own instead of the system do it for them**

<table>
<thead>
<tr>
<th>#</th>
<th>Review Checklist</th>
<th>Yes</th>
<th>No</th>
<th>N/A</th>
<th>Comments</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>2 User Control and Freedom</strong></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>2.1</td>
<td>Can users cancel out of operations in progress?</td>
<td>☐</td>
<td>☒</td>
<td>☐</td>
<td></td>
</tr>
<tr>
<td>2.2</td>
<td>Can users easily reverse their actions?</td>
<td>☒</td>
<td>☐</td>
<td>☐</td>
<td></td>
</tr>
</tbody>
</table>

3. **Different words shouldn’t give the same meaning, and the users should not be confused.**

<table>
<thead>
<tr>
<th>#</th>
<th>Review Checklist</th>
<th>Yes</th>
<th>No</th>
<th>N/A</th>
<th>Comments</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>3 Consistency and Standards</strong></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>3.1</td>
<td>Have Moodle formatting standards been followed consistently?</td>
<td>☐</td>
<td>☒</td>
<td>☒</td>
<td></td>
</tr>
<tr>
<td>3.2</td>
<td>Does the menu structure match the task structure?</td>
<td>☐</td>
<td>☒</td>
<td>☒</td>
<td></td>
</tr>
<tr>
<td>3.3</td>
<td>Are field labels and fields distinguished typographically?</td>
<td>☐</td>
<td>☒</td>
<td>☒</td>
<td></td>
</tr>
<tr>
<td>3.4</td>
<td>Are attention-getting techniques used with care?</td>
<td>☒</td>
<td>☒</td>
<td>☒</td>
<td></td>
</tr>
<tr>
<td>3.5</td>
<td>Does the system follow Moodle standards for function key assignments?</td>
<td>☒</td>
<td>☒</td>
<td>☒</td>
<td></td>
</tr>
</tbody>
</table>

4. **Error messages should be displayed in plain language no codes.**

<table>
<thead>
<tr>
<th>#</th>
<th>Review Checklist</th>
<th>Yes</th>
<th>No</th>
<th>N/A</th>
<th>Comments</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>4 Help Users Recognize, Diagnose, and Recover From Errors</strong></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>4.1</td>
<td>If an error is does the system place the error in the block: field: sect?</td>
<td>☐</td>
<td>☒</td>
<td>☒</td>
<td></td>
</tr>
<tr>
<td>4.2</td>
<td>Do error messages suggest the cause of the problem?</td>
<td>☐</td>
<td>☒</td>
<td>☒</td>
<td></td>
</tr>
<tr>
<td>4.3</td>
<td>Do error messages indicate what action the user needs to take to correct the error?</td>
<td>☐</td>
<td>☒</td>
<td>☒</td>
<td></td>
</tr>
<tr>
<td>4.4</td>
<td>Does the system warn users if they are about to make a potentially serious error?</td>
<td>☐</td>
<td>☒</td>
<td>☒</td>
<td></td>
</tr>
</tbody>
</table>
5. Irrelevant or rarely used words should not be used on dialogues.

<table>
<thead>
<tr>
<th>#</th>
<th>Review Checklist</th>
<th>Yes</th>
<th>No</th>
<th>N/A</th>
<th>Comments</th>
</tr>
</thead>
<tbody>
<tr>
<td>5</td>
<td>Aesthetic and Minimalist Design</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>5.1</td>
<td>Are all icons in a set visually and conceptually distinct?</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>5.2</td>
<td>Does each icon stand out from its background?</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>5.3</td>
<td>Does each data entry screen have a short, simple, clear, distinctive title?</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>5.4</td>
<td>Are field labels brief, familiar, and descriptive?</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>5.5</td>
<td>Are menu titles brief, yet long enough to communicate?</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

6. System interaction should enhance users quality work.

<table>
<thead>
<tr>
<th>#</th>
<th>Review Checklist</th>
<th>Yes</th>
<th>No</th>
<th>N/A</th>
<th>Comments</th>
</tr>
</thead>
<tbody>
<tr>
<td>6</td>
<td>Pleasurable and Respectful Interaction with the user</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>6.1</td>
<td>Is each individual icon a harmonious member of a family of icons similar to the moodle system?</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>6.2</td>
<td>Has color been used with will it make it better?</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

7. The system should only be accessible to user who are managers, teachers and instructors.

<table>
<thead>
<tr>
<th>#</th>
<th>Review Checklist</th>
<th>Yes</th>
<th>No</th>
<th>N/A</th>
<th>Comments</th>
</tr>
</thead>
<tbody>
<tr>
<td>7</td>
<td>Accessibility</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>7.1</td>
<td>Is the block accessible with user login?</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

8. Is the system useful for real time learning environment.

<table>
<thead>
<tr>
<th>#</th>
<th>Review Checklist</th>
<th>Yes</th>
<th>No</th>
<th>N/A</th>
<th>Comments</th>
</tr>
</thead>
<tbody>
<tr>
<td>8</td>
<td>Usability</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>8.1</td>
<td>Will you use the product in your activity to change course and assignment start and end dates?</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

---

Your final comment if any

9. I was missing was the possibility to change all dates at the same time.
Appendix D: Future development suggestion layout

When 'Adjust Dates' on the left is clicked, it should show the tabular layout which is shown in Right.