E-Moodles

Moodle Plug-in

Software Design Document

Names: Shani Shalel, Shani Hayik, Shirel Israelov

Lab Section:

Workstation:

Date: 24.12.2020

TABLE OF CONTENTS

1. INTRODUCTION
   1.1 Purpose
   1.2 Scope
   1.3 Overview
2. SYSTEM OVERVIEW
3. SYSTEM ARCHITECTURE
   3.1 Architectural Design
   3.2 Decomposition Description
   3.3 Design Rationale
4. DATA DESIGN
   4.1 Data Description
   4.2 Data Dictionary
5. COMPONENT DESIGN
6. HUMAN INTERFACE DESIGN
   6.1 Overview of User Interface
   6.2 Screen Images
   6.3 Screen Objects and Actions
7. DIVISIN OF RESPONSIBILTY

1. INTRODUCTION

1.1 Purpose

This software design document describes the architecture and system design of E- Moodle plug-in , the intended audience is students and lecturers.

1.2 Scope

software product name: E-moodles The software product will allow to send mail directly from personal email to the Moodle system. The plugin E-moodles will allow the user to :

- Sending massages to the Moodle system directly from his mail.
- Response to messages directly from Email.
- See all conversations from Email/Moodle.
- Filter the relevant messages for him.

1.3 Overview

This document will provide an explanation on the system on its sub-system and how to will build and provide a screen shot of how our E-Moodles user's perspective will see.

2. SYSTEM OVERVIEW

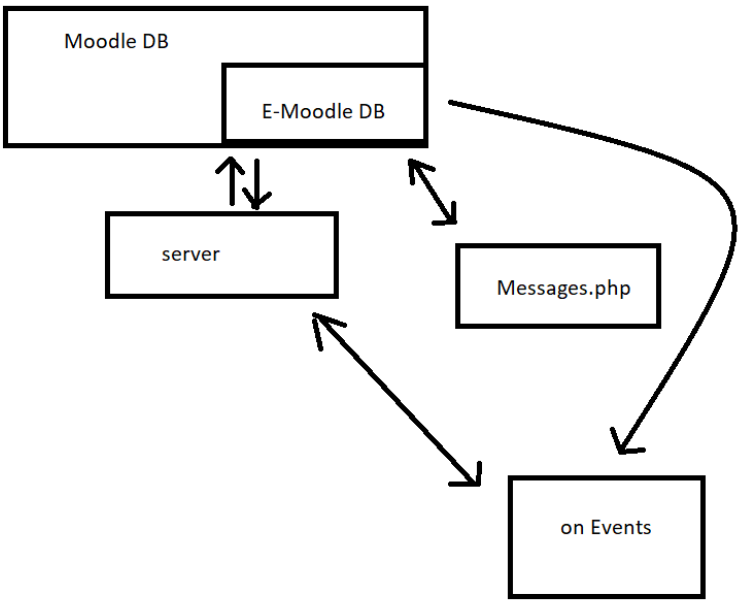
The users will send a message to email Moodle.number of the course@google.com to demonstrate the plugin that we will create will need to filter the emails that connected to the specific course and to provide a nice view of the message to Moodle site.
We will use SQL DB and php,html language.

3. SYSTEM ARCHITECTURE

3.1 Architectural Design

- Server that gets and sends the mails from the users and save them to DB.
- Php file that gets the messages from the Moodle web and sends them via emails and save them to DB.
- Php file that present all the messages in nice way.

3.2 Decomposition Description



That decomposition describes the relationship between the objects as follow:
The E-Moodle DB is getting update by the Messages.php file on specific Events we will send an email throw the server and the server is getting emails and updating the DB E-Moodle.

3.3 Design Rationale

We choose the sever that gets and send the emails so that the user will be able to reply the mail he gets and to send an email directory from his mail system.
We choose the php file that gets the messages and the one that present them because we see an example of quickmail and that how it works.

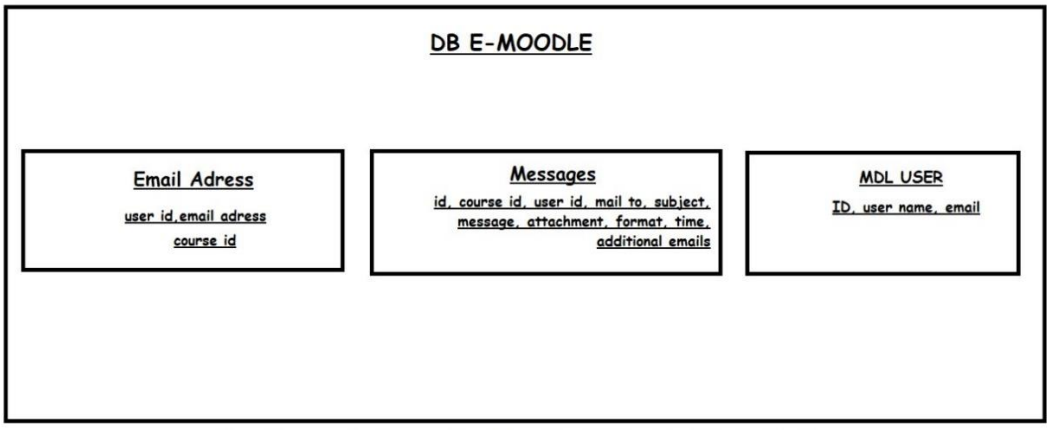
In additional we choose to add a plugin and not to replace the code because the code is use for specific events as the one that the teacher wants to send all the student and don't want to allow the students to reply.

4. DATA DESIGN

4.1 Data Description

The DB will use MySQL server and will organized by number of courses .
Sql server is build on tables so we will organized the messages by user_id, email address, course_id, mail_to, subject, message, attachment, format, time, additional emails.

4.2 Data Dictionary

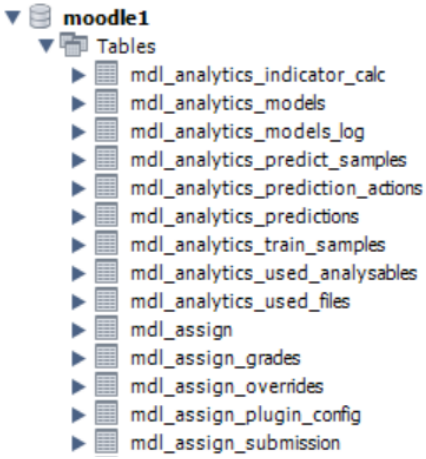


This is how our DB of E-Moodles will look like. We have 2 tables: Email Address, Messages and we will use the table mdl_user from Moodle DB. in each table we have information like user id, email, course id and so on. Those tables will help us to get information, data that we need easier.

The database of Moodle is used in MySQL . it builds from tables that every table have another subject. Example: the table "mdl_course" is table of the courses of the specific user with number of colums as we can see in the photo:

Result Grid | Filter Rows: | Edit: | Export/Import: | Wrap Cell Content:

id	category	sortorder	fullname	shortname	idnumber	summary	summaryformat	format	showgrades	newsitems	startdate	enddate
1	0	1	Email-moodle	E-moodle			0	site	1	3	0	0
2	1	10001	stam_to_check	stam			1	topics	1	5	1608847200	16403832
NULL	NULL	NULL	NULL	NULL	NULL	NULL	NULL	NULL	NULL	NULL	NULL	NULL



5. COMPONENT DESIGN

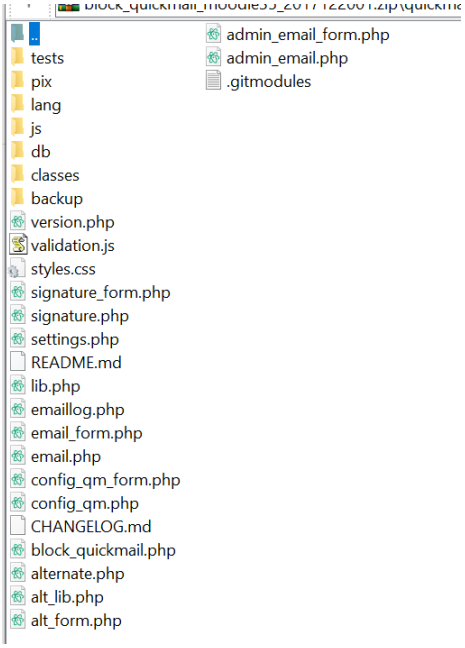
Our component will have the next directories:

- DB- will have 3 files : access, install, upgrade
- Tests
- Classes- (the objects of the plugin) Message, log (present all the messages) and other object that we will need.
- Events- update
- Server

We must have file that display the version- version.php

We must have file that start the E-Moodle DB -install.xml (in DB)

We will add an example of plugin “block_quickmail” that send noreply emails to all the students:



For example, the message class is the object that send the noreply message and he has function as:

- __construct
- Send – send all the emails.
- buildAdminReceipt- builds a receipt emailed to admin that displays details of the group message.
- sendAdminReceipt- sends the admin receipt.

6. HUMAN INTERFACE DESIGN

6.1 Overview of User Interface

The functionality of the system from the user side is :

- the able to send emails throw the email system by the relevant email address (according to the number of course).
- the able to see all the message in view in Moodle site .
- the able to filter the emails that he gets

6.2 Screen Images

6.3 Screen Objects and Actions

The user will be able to response the mails that any other user will send .

7. **DIVISIN OF RESPONSIBILTY**
    - Learn about plugin to Moodle-Shani Hayik
    - Learn about Email-server- Shirel
    - Learn about Moodle-DB – Shani Shalel
    - Learn the Moodle environment – Everyone