**https://github.com/winonawinona/group2WAT**

**THINGS TO REVISE:**

**Part 1 redundant**

**Add database to the system (mySQL)**

**JavaScript version 1.8.5 is the programming language of HTML and the Web.**

**It is used to program the behavior of the web.**

**HTML5 (Hyper Text Markup Language)**

**HTML is the language to create websites**

**HTML is used to define the content of the website.**

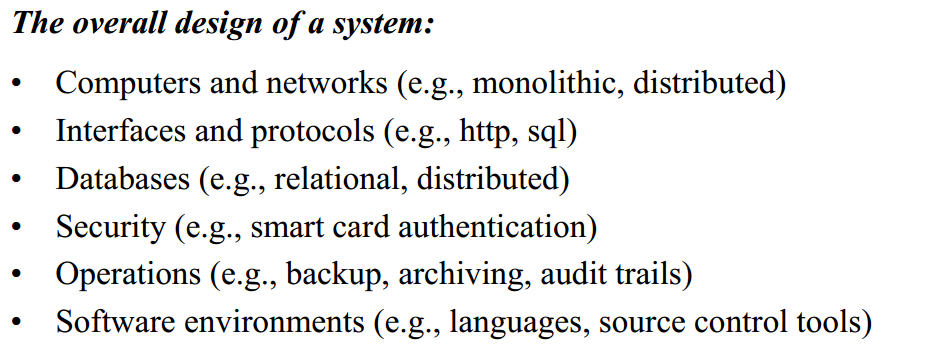
**CSS (Cas*cading* Style Sheet)**

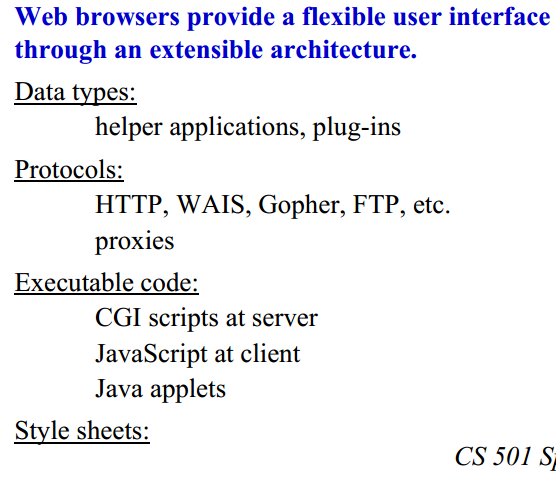
**CSS is to specify the layout of the web pages.**

**CSS is a stylesheet language that describes the presentation of an HTML (or XML) document.  
  
CSS describes how elements must be rendered on screen, on paper, or in other media.**

**Specs? Javascript ba ung ginamit? Additional plugins?**

**www.MBU.com/home**





**ERD- AJ :)**

**-boxes? Client should understand? What is PK?**

**PK is a primary key, in layman's term it is a “unique identifier”.**

**-missing email & password**

**-FINAL GRADE, GRADE, type (SW/HW)**

**-PAYMENT status? full/inc?**

**-FACULTY: status(employed/resigned)**

**- TRANSACTION database, timer of inactivity**

**GITHUB**

**-screenshots of the repository, uploaded documents(can be only 1) or step by step process(to be precise). - wat**

**NAKS WIN!**

**TRI:**

**Case change arrow into tomato**

**AJ:**

**Non functional req**

**WIN: develop using \_\_\_? Website?**

**DUMMY**

**E/R diagram? Flowchart?**

**Error handling for UI? (resigned person,not enrolled student, incorrect credentials)**

**Skilled users offered shortcuts? REVERSE? UNDO?**

**Doi we need to have administrator of faculty? Or for IT? Rights? WHEN RESIGNED CAN have inactivate account History of edits like google apps? Or logbook?**

**HW: GITHUB :) I’m on it - tri**

**note: gawa na kayo ng account para ma try natin maraming users. medyo gets ko na ung github(basics).**

**Tomato is yummy - AJ**

**User has no Server, (What’s the plan ? ^\_^)**

**Idedeploy daw(WebApp) sa server sabi ni sir WAHEHEHEHE-win**

**MotherBoard University**

**Web application grading system**

**\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_**

**Version 1.0**

**03/09/2016**

***Software Requirements Specification***

**Adrian Jacob L. Severino**

**Triwynn Benedict E. Branzuela**

**Winona Janelli T. Dumpit**

**Requirements Documentation:**

1. **General**
2. **Purpose and scope of system**

By request of Mother Board University, a system shall be created to fill the need of the university for an effective yet simple way to connect teachers and students in terms of subject grading

The system is a web application accessed through web browsers, specifically Mozilla Firefox and Google Chrome, and will assist faculty members in registering students by section, calculating and recording different grading criterion and enabling students to view their grades, all while storing the data in a database.

The system will be available only to faculty members and students who are enrolled in the school and will involve the subjects that all teaching faculty members are handling. Activities done by the school will be made easier and more efficient through use of this system, specifically activities in assigning grades and viewing grades.

1. **Objectives and criteria for success**

The objective is to provide the client with a system that makes grading and viewing grades more efficient, quick, and linked to the internet but still simple enough to be operated. All functionalities of the system will be related to grading and will therefore prioritize in being the school’s digital assistant in recording grading data..

1. **List of terminology & organizations involved**

|  |  |
| --- | --- |
| Term | Definition |
| Faculty member | Employees of the school involved in subjects. These include faculty members, deans, and department heads. |
| IT | Information technology |
| Student | Individuals enrolled in MotherBoard University |
| User | Includes faculty members and students who have accounts in the system |
| MBU | MotherBoard University grading system |
| Database | Table used to store the grade data |

1. **Description of the current system**

The system created will be a web application that can be accommodated mainly by Mozilla Firefox and Google Chrome using languages in English and Cebuano. Its main purpose is for viewing of grades per subject both by the faculty members and students and involved in a certain subject. Faculty members, however, may also interact with the system as long as they are permitted.

Unless a faculty member user chooses to save new information, all information will be retained in the system and will be used for viewing purposes of the students.

Faculty members’ features include enlisting in a subject and section as well as assigning the corresponding score of per student. Each student can be scored either based on different criteria for grading set by the school (homework, seatwork, quiz, preliminary exam, midterm exam, final exam, etc.), where the system computes for the grade, or by inputting the final grade computed by a faculty member user.

Users from the school faculty can select “Save” or “Publish” when manipulating data. Save enables the faculty member to store new grades in the list of grades and exit the application. Meanwhile, publish entails that the student’s grade cannot be edited anymore and will be available for viewing by the student. A graphical representation of a student’s score per subject is available for the faculty member for further analysis.

Student features will include viewing grades depending on the number of subjects they are enrolled in, which will be enlisted by the school faculty.

Both users will be automatically logged out if the faculty member and student has an inactive session of 30 minutes and 15 minutes respectively.

1. **Requirements of proposed system**
2. **Functional requirements**
3. The system must be able to receive, store, delete, and deactivate names of subjects as well as their corresponding codes (e.g. “Basic Java Fundamentals; JAVA101”). These data will later be accessed by accounts and will act as a connection between faculty member users and student users.
4. It must be able to receive, store, delete, and deactivate data regarding a faculty member’s name and e-mail address, given by the faculty member user, the subject being handled and the section, which will be chosen from the subjects stored within the system. (refer to a.)
5. It must also be able to receive, store, delete, and deactivate data regarding a student’s name, e-mail address and ID number, given by the faculty member user, the subjects he is taking, chosen from subjects stored within the system, as well as whether he has fully paid his required tuition fee for the semester (refer to a.)
6. For all users (faculty member and student), an option to change the password is available in the case that a password seems undesirable to the user.
7. In the case that a user cannot recall his/her password, the system can be prompted to send the password to the user e-mail.
8. The data given to the system must be updateable in the case that:
   1. The faculty member user incurs a typographical error
   2. The faculty member user wishes to change the data because of a change in the actual information (e.g. grades were changed due to a late submission that was considered)
   3. The faculty member user wishes to correct discrepancies in the case that any of them are made
9. The system must be able to receive grades given by the faculty member user and store it, through a save button, into the grades of the student on whose account the faculty member selects using two methods:
   1. The faculty member user can input the total grade computed beforehand
   2. The faculty member user can input the grade by increments using criteria for grading (e.g. homework, quiz, midterms)
10. In a separate page, the faculty member must be able to view the subject sections handle by him/her, the students under each section, and the grades assigned to them.
11. In another page, the student must also be able to view the subjects he is enrolled in and the grades assigned by the designated faculty member. If the student has not yet settled his tuition fee account, the system will hide all the information contained within the page and, instead, notify that the student that he must settle his account first before viewing his grades.
12. If a user has been inactive for a specified span of time, the account will immediately log out from the web application. The specified is thirty (30) minutes time for faculty members and fifteen (15) minutes for students
13. **Non-functional requirements**

**Logical Structure of the Data**

**<insert ERD here>**

**<tables of data>**

**Performance Requirements:**

1. MBU can accommodate multiple users, up to 50 simultaneously per interface instance to interact with the web application.
2. The system can be accommodated by at least Mozilla Firefox AND Google Chrome browsers.
3. Only faculty members are allowed to manipulate data, and only what is allowed. Faculty members cannot manipulate data unless they are enlisted in the subject and section, and students are not allowed to manipulate any data at all.
4. All data must be stored within the system as long as they are not overwritten by the user.

**Operational Requirements:**

1. The client will be responsible for hiring an IT personnel for continuous maintenance of the system after project closure.

**Compatibility Requirements:**

1. The system is accessible through web browsers. By request, the system is compatible mainly with Mozilla 44.0.2 version browser and Google Chrome browser version 48.0.25 or better.

**Availability Measures:**

1. The system is through a web browser using a server provided along with the system.
2. Assuming the internet connection has a minimum of 1 MB per second. The system shall meet or exceed 99.99% uptime.
3. To ensure that the system shall be used by students and teachers of the school, the system will need an email address per user created by the university.

**Security Measures:**

1. To avoid discrepancy, faculty member accounts will require password protection.
2. Only faculty member accounts can make changes in the data of the system. Student accounts can only view grades to ensure that no student may manipulate data.
3. A separate account type shall be created for non teaching faculty members, (Deans, Presidents) to manipulate data that the teacher type cannot access (Subject names, Subject Codes)

**Usability Measures:**

1. Four out of five faculty member users shall be able to create a student record within 10 minutes after an hour of training.
2. Novice users shall perform creating of 1 subject with complete categories for scoring, correct formula in 10 minutes.
3. Experienced users shall perform creating of 1 subject with complete categories for scoring, correct formula in 3 minutes.
4. At least 85% of faculty members surveyed after 1 semester of usage shall rate their satisfaction with the system at 7 or higher on a scale of 1 to 10.
5. User training will be provided for the faculty members using an a desktop computer for approximately 2 hours.

**Maintainability Measures:**

1. Upgrading to a new version shall leave all database contents and personal settings unchanged.

**Testability Measures:**

1. The delivered system shall include unit tests that have at least 70% of covered scenarios.

**Robustness Measures:**

1. The estimated loss of data in case of a disk crash shall be less than 0.01%.
2. The system shall be able to handle at least 50 faculty members concurrently using and up to 1000 concurrent students upon logging in or browsing their grades.
3. **System Models**
4. **Scenarios**

Scenario: Faculty member

Individual: Winona Dumpit, Faculty member at Motherboard University, handles Basic Nihongo for Dummies (NGO101), location Vidal Tan Hall.

Equipment; Acer desktop computer with Windows 7 Operating System. Mozilla 44.0.2 version browser, Google Chrome browser version 48.0.25 and MBU Grading System.

Scenario:

1. Professor Winona opens up the computer and starts the web browser.
2. Professor Winona logs into the MBU.
3. Professor Winona is then redirected by the browser to the faculty member’s page.
4. The faculty member’s page now allows the faculty member to record, view, publish or print report cards.
5. Professor Winona chooses to record option.
6. Professor Winona then records the student’s grades individually.
7. She then views the grades of all students.
8. Professor Winona finally chooses to publish the complete set of grades of the students.
9. The faculty member can still edit the grades before they fully publish it.
10. Professor Winona prints report cards of those students that are already complied

with the requirements.

1. Professor Winona then logs out of the MBU and closes the browser.

Scenario: Fully-paid Student

Individual: AJ Severino, a student at Motherboard University, major in Nihongo.

Equipment; Acer Desktop computer with Windows 7 Operating System. Mozilla 44.0.2 version browser, Google Chrome browser version 48.0.25 and MBU Grading System.

Scenario:

1. AJ opens up the computer and starts the web browser.
2. AJ logs into the MBU.
3. The browser then redirects the student to the student’s page.
4. AJ can now view his grades.
5. AJ then logs out of the MBU and closes the browser.

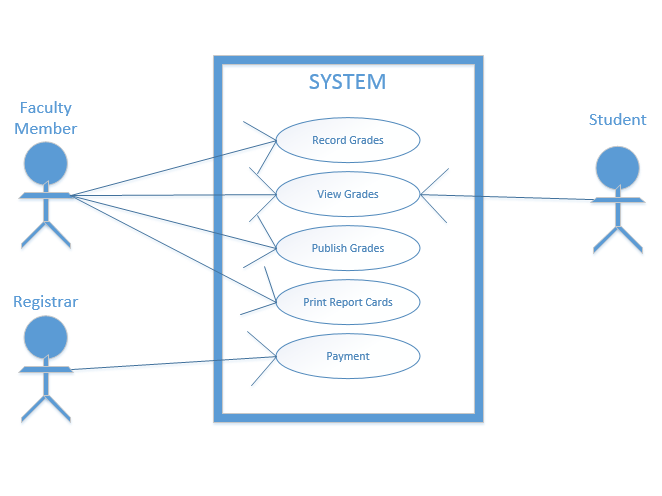
Scenario: Unpaid Student

Individual: Kenny McCormick, a student at Motherboard University, major in Nihongo.

Equipment; Acer Desktop computer with Windows 7 Operating System. Mozilla 44.0.2 version browser, Google Chrome browser version 48.0.25 and MBU Grading System.

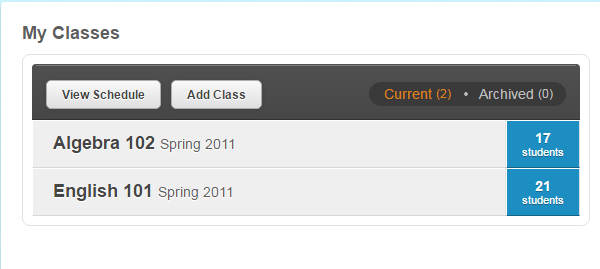
Scenario:

1. Kenny opens up the computer and starts the web browser.
2. Kenny logs in to the MBU.
3. Kenny The browser then redirects the student to the student’s page.
4. Kenny cannot view his grades because he hasn’t paid his tuition fee yet.
5. Kenny then logs out of the MBU and closes the browser.

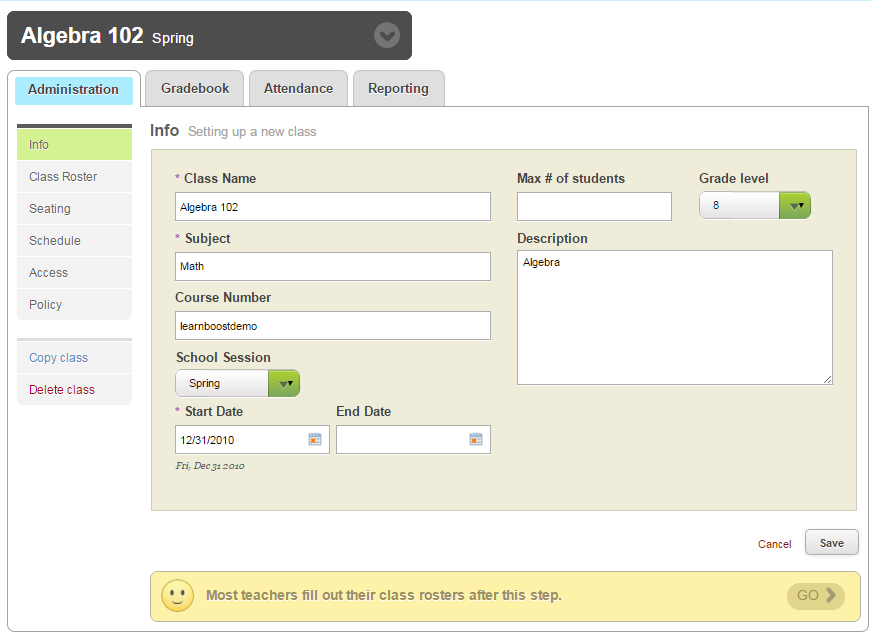
1. **Use cases** 
2. **UI**

**faculty member’s page**

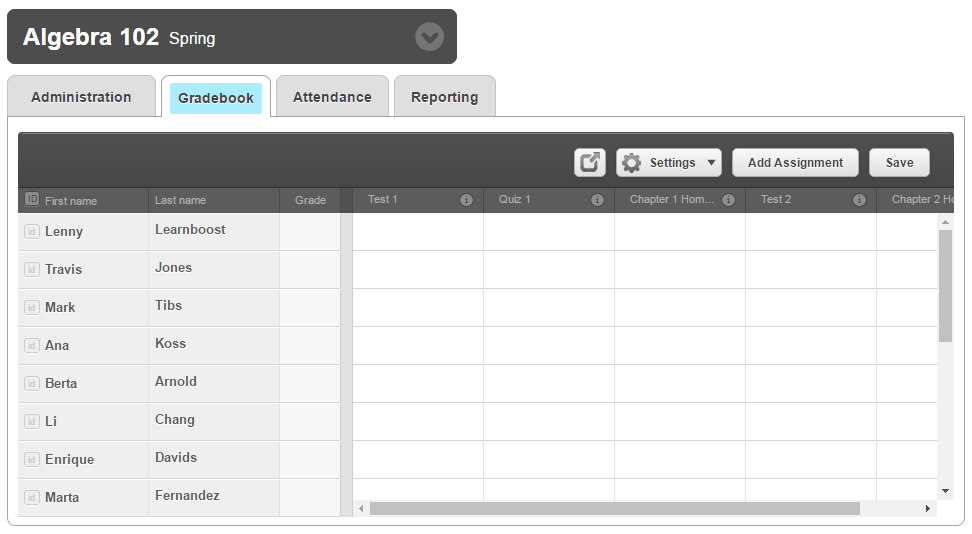
**View Classes**



**Administration Tab**

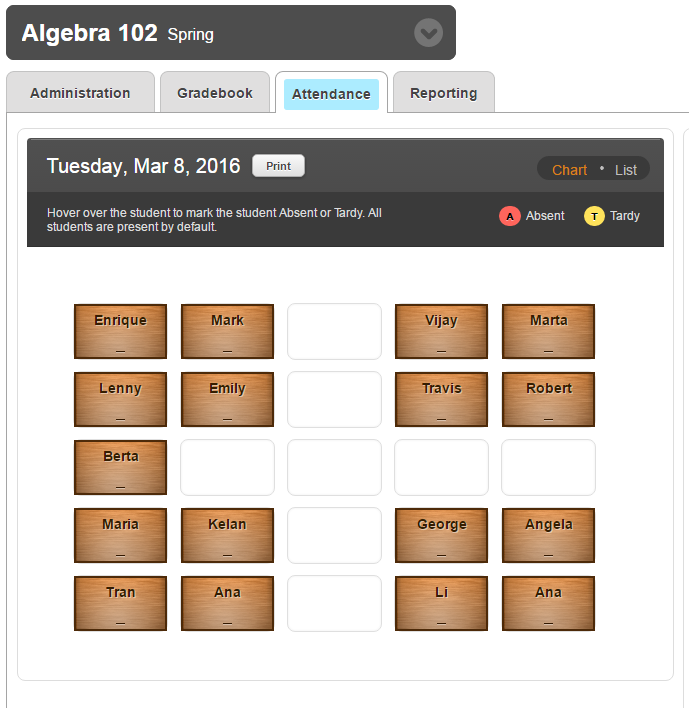


**Faculty member’s(Teacher) Gradebook**



**I**

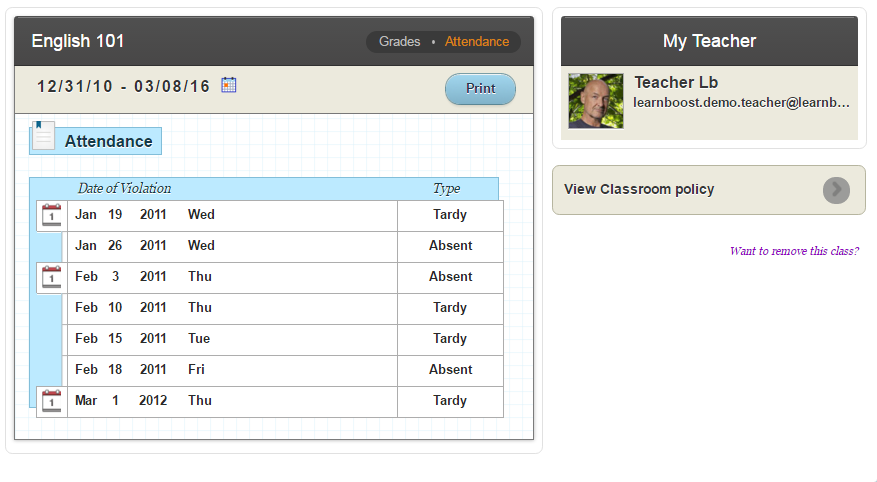
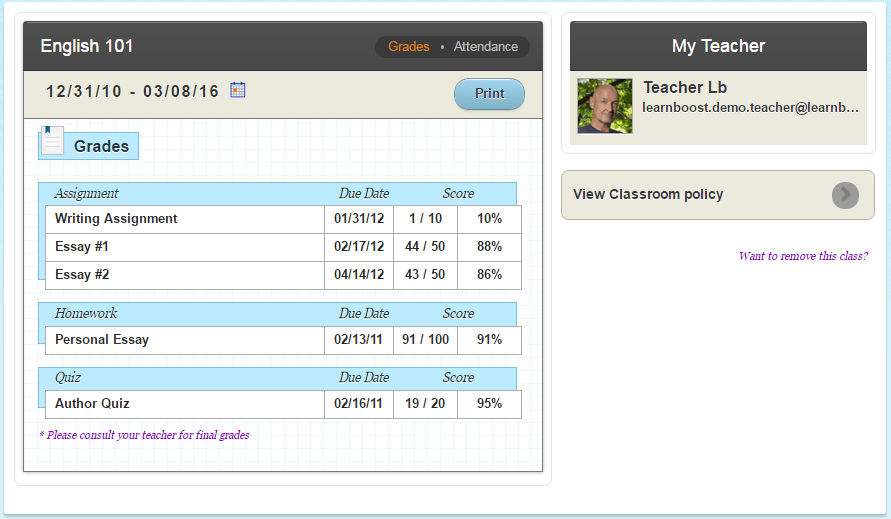
**Attendance Tab**



**Reporting Tab**



**Student’s Page**



**VI. Deployment Plan**

The MotherBoard University system will need to undergo testing to ensure that its performance will fit the needs and demands of the MotherBoard University. Both users from the development team as well as users from the client will be given an opportunity to test the system’s functionality and efficiency and, once both set of users are satisfied, the final version of the system shall be created and released for use of the MotherBoard University. During the testing period of the users from the client, they shall be assisted in familiarizing with operating the system for 14 days.

The server required for the system to operate shall be provided by the development team and shall be included in the cost for the final version of the system. Since the client wishes to hire maintenance personnel to maintain the system as needed, the development team shall give them ample time to do so as well as to train such personnel in the operation of the MotherBoard University system. A period of six(6) months after the deployment of the final product shall be provided for this, beginning August 29, 2016, with the cost also included with the final version of the system.

The cost incurred from providing the server used by the system will be a total of P 215,000; this includes all the hardware required as well as the fee for the set up. The maintenance and training provided by the development team which spans for six (6) months, shall incur a cost of P 15,000 per month.