**Animon.sys Enrollment System**

A Requirements Specification Document

for the course on

Introduction to Databases

(INTRODB)

Submitted by:

Aquino, Kurt Neil

Choy, Matthew Seaver

Hade, Alden Luc

Ms. Ethel Ong

Teacher

July 28, 2014

1. Introduction

In 2012, DLSU released Animo.sys, a new online enrollment system for the university. This is where students can enlist in available courses for the next term, and with the system being online, this means that students can enlist much faster than they normally would if it were manual enlistment, or if it were My La Salle, which it replaced due to server issues.

What we’re doing here is our own simplified version of Animo.sys, a version without some of the features on the website that people do not usually use. Features such as the extra buttons that appear at login, like PeopleTools, and My Personalizations. Options that are made for the network administrations are not necessarily needed to be viewed by casual users like enrollees.

We call it Animon.sys, named to honor the legacy of one of our previous professors.

The aim with this system is a more simplified UI than the current Animo.sys, such as going straight to a main menu rather than having to expand “Self-Service” before clicking “Add Classes” and having to load that page. We also aim to remove the stuff that students don’t use, such as the aforementioned PeopleTools, My Personalizations, and maybe even Swap Classes. Overall, our system’s goal is to add, drop, search and display courses desired by the enrollee, which is more straight to the point.

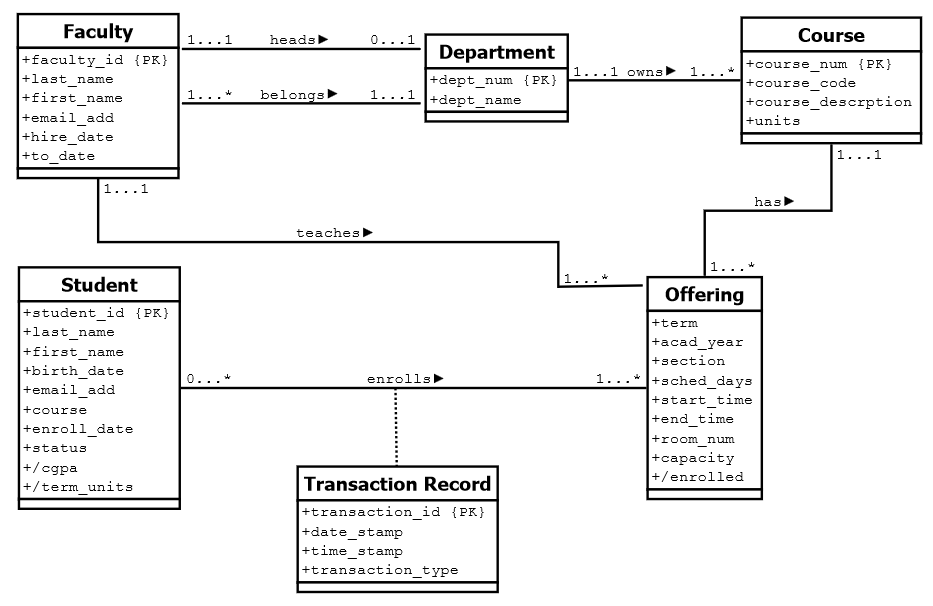
Students can use the system to add classes to their upcoming schedule, provided of course that those classes are not already full, or that the classes does not overlap each other. Courses are uniquely identified via course coeds, and specific classes of these courses are uniquely identified via course numbers. They can also drop classes here, for this term and the next, as long as the student is above the minimum units.

Students are of course our primary target with Animon.sys. Via the features listed above, the program essentially becomes a means by which students can conduct business with the school. Enlistment and removal of classes are done in this program, as well as viewing the current and upcoming class schedule. Viewing of the class schedules can be interpreted as similar to manually asking for an EAF for the current term. The enlistment and removal processes help support the first and most important phase of enrollment. Afterwards, the student must then manually pay for their tuition fees. Our program supports all of these these business processes except for the payments for obvious reasons.

The students, can access the data that they themselves need. They cannot, for example, see data pertaining to another student’s class schedule. They can only see the list of courses, the data associated with them (such as Professor or Room), their own schedules and already enlisted courses. The system accomplishes this by showing reports, which are programmed with extensive use of SQL scripts, combined with Java in order to have a good user interface.

Examples of reports include the list of all classes in a course, list of courses offered, and a list of all remaining available class offerings in a course.

1. Data Requirements
   1. Conceptual Data Model



Entities and their Attributes:

*Student* is used to store the data of each enrolled student in the Aminon.sys system. It contains the following fields:

* *student\_id* (PK) – contains an 8 digit number that is unique for every student and is used by the system in order to identify whose data to access.
* *last\_name* – last name of the student
* *first\_name* – first name of the student
* *birth\_date* – birthdate of the student
* *email\_add* – contact email address of the student for the administration and their respective professors to use when needed.
* *course* – current course being taken by the student
* *enroll\_date* - date the student has enrolled in the system (yyyy-mm-dd)
* *status* – current status of the student (undergraduate/graduate)
* *cgpa* (derived) – cumulative grade point average of the student
* *term\_units* (derived) – total number of units enrolled for the next term to determine whether the student is underloaded or not

*Faculty* contains the information of each faculty member currently teaching in the specified school. It contains the following fields:

* *faculty\_id* (PK) – contains a 5 digit number that is unique for every faculty member and is used by the system in order to identify whose data to access.
* *last\_name* – last name of the faculty member
* *first\_name* – first name of the faculty member
* *email\_add* – contact email address of the faculty member for the administration or the students to use when needed.
* *hire\_ date* – date the faculty member has been hired (yyyy-mm-dd)
* *to\_date* – date the faculty member has been either moved to another department or has retired/fired (yyyy-mm-dd)

*Course* contains all of the details of each available course currently being provided in the term. It contains the following fields:

* *course\_num* (PK) – contains a number that is unique for every course and is used by the system to determine which course is being used
* *course\_code* – alphanumeric field which contains the abbreviated title of the course name for easier identification for the user
* *course\_name* – official name of the course
* *units* – number of units given by this course

*Offering* contains the details of each class a course offers. It contains the following fields:

* *term* – current term the course is currently being offered in
* *acad\_year* – current academic year the course is currently being offered in
* *section* – section in which the course is available to
* *start\_time* - starting time of the course
* *end\_time* - ending time of the course
* *sched\_days* – days when the course is scheduled (M,W,T,H,F)
* *room\_num* – an alphanumeric field which determines which room and building the course is being held
* *capacity* – maximum number of students who can enroll in this course
* *enrolled* (derived) – current number of enrolled students to this course

*Department* contains the details of the departments each faculty member belongs to. It contains the following fields:

* *dept\_num* (PK) - contains an alphanumeric code that is unique for every department
* *dept\_name* – name of the department

*Transaction* *Record* records the information of the transactions being made by the student in the enrollment system. It contains the following fields:

* *transaction\_id* (PK) – unique 12 digit key of every transaction being made for the system to use
* *date\_stamp* – date when the transaction is being made (yyyy-mm-dd)
* *time\_stamp* – time when the transaction is being done (hh:mm)
* *transaction\_type* – determines which course of action is being done
* adding a course
* dropping a course

Entity Relationships:

A ***Department***, containing a *unique department number and name*, has multiple *Faculty* members, but can only headed by only one. Each ***Faculty*** member has a *unique faculty id, last name, fist name, email address, hire date,* and *current date*.

Each ***Department*** offers many different ***Courses***. And each ***Course***, containing a *unique course number, code, and description* can be offered multiple times. An ***Offering*** contains the *term* and *academic year* it is currently being offered to,as well as the *section, scheduled days, start* and *end time, room number, capacity,* and the *enrolled number of students* to the offered course. Each ***Offering*** can only be headed by a single ***Faculty*** member, but a ***Faculty*** member can head multiple classes/offerings.

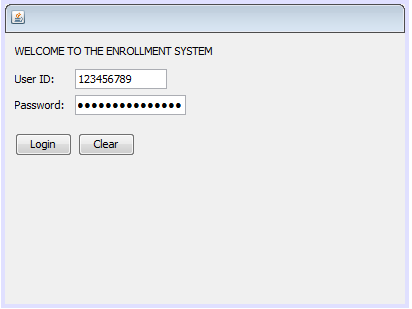
A ***Student***, *containing a unique student id, last name, first name, birth date, email address, course, enrolled date, status,* and *current GPA* can enroll many ***Offerings*** but is restricted according to the *total number of units* enrolled by the student for the next term (whether the student is under or overloaded) . Whenever a ***Student*** enrolls or drops an ***Offering***, a ***Transaction Record*** is made, with each one containing a *unique transaction id*, a *date* and *time stamp* when the transaction occurred,and the *transaction type* being done (i.e. added or dropped).

Business Rules:

* A student or faculty’s password should not be less than 8 characters long and should contain an uppercase letter and a special character (!, -, ?, @)
* A student or faculty must use a legitimate email address as it will be used to contact them.
* A student may enroll no less than 12 units per term
* A student may enroll no more than 21 units per term
* Enrolling of multiple classes with conflicting time schedules is not allowed
* A student may not view other students enrolled in a given course
* A student may only enroll a single offering of a given course
* A student may not enroll a course if he/she has not completed its prerequisite(s) (if there is/are any)
* Only a faculty member can view the enrolled classes of other students
* A faculty may only view his/her assigned courses and offerings as well as other faculty members’, but may not edit them
* A faculty may only view the students enrolled to his/her course offerings, but may not add or remove them
* Only the head of the department/administrator can view the transaction records of the students
  1. Software Features

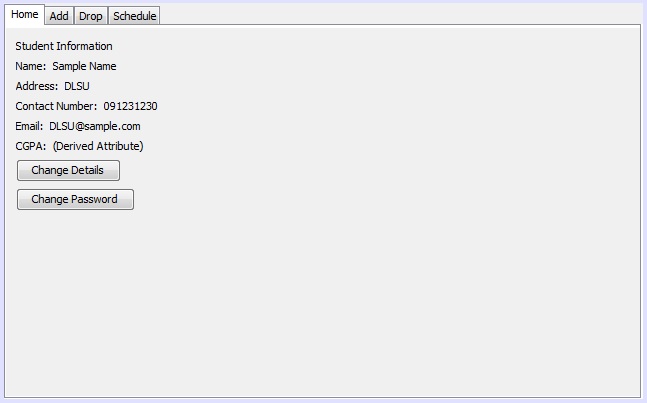
**For End Users/Customers:**

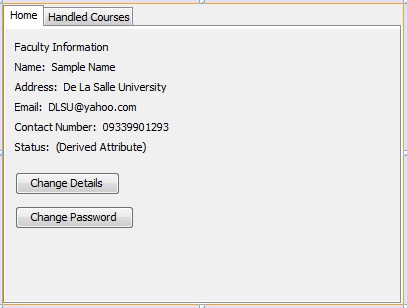
**LOGIN:**



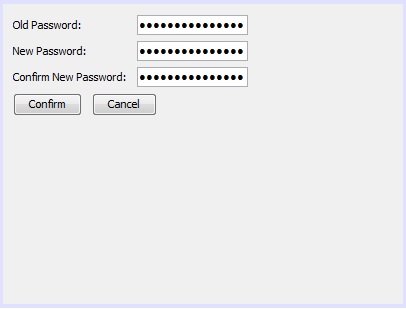
This would allow users to use their user ID and password to login and to keep their information secure and safe. Users cannot create a new account since the accounts are manually inputted in the database. The user ID would be searched through the entire database of faculty and students. Once found, the user will be redirected to their corresponding pages.

**Change Password/Change Details:**

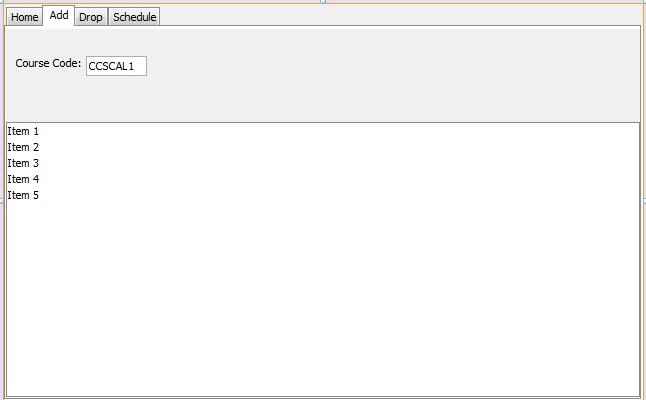




Both faculty and student are able to change the details that are wrong or changed. Of course, before being able to do this, they have to enter the current password to change it. At the same time, the changing of password requires the input of old password, new password, and a confirmation of the new password.

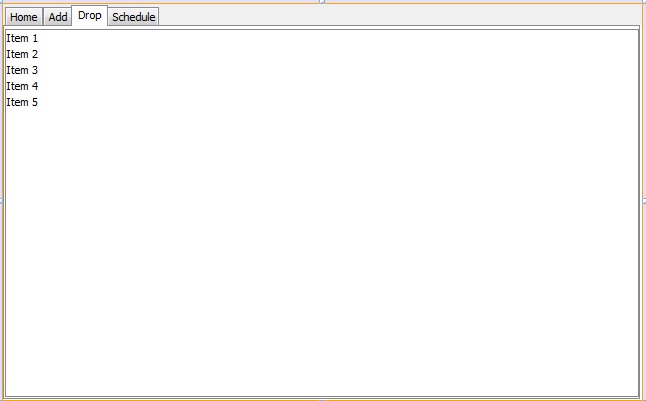


**Adding Classes for Students:**

****

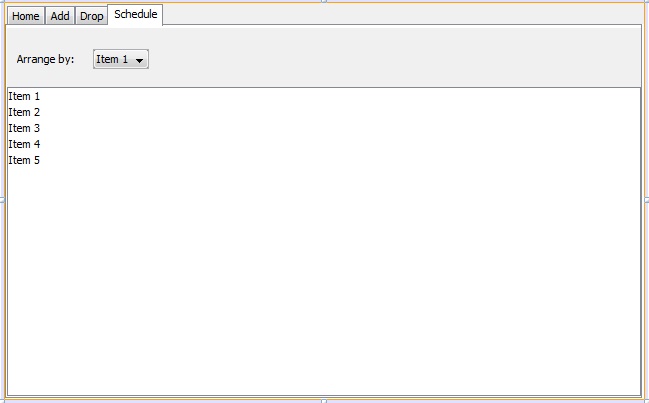
To add a class, the student must first search the course code then search results are placed in the list of items. Each item in the list would contain course code, section, and corresponding day and time, all separated by spaces. For example, “Item 1” will have values: “CCSCAL1 S17 MW 1100-1230.” After double- clicking the class, a pop up window, displaying the information of the course with a ‘yes’ or ‘no’ button and asking if he/she is sure, would come out and ask if he/she is sure of the class. After a ‘yes’ is given, the selected class will be added to his/her schedule.

**Dropping Classes for Students:**



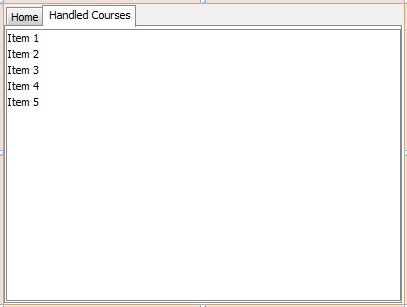
To drop a class, the student must select an item that he/she added in the add tab. Each item in the list would contain course code, section, and corresponding day and time, all separated by spaces. For example, “Item 1” will have values: “CCSCAL1 S17 MW 1100-1230.” If no item was picked in the add tab, nothing is in the list and nothing can be selected. After double-clicking the class that he/she selected, a pop up window, displaying the information of the course with a ‘yes’ or ‘no’ button and asking if he/she is sure, would come out. After a ‘yes’ is given, the selected class will be taken out of his/her schedule.

**Current Schedule of Students:**

****

A student can check his/her schedule by clicking the schedule tab. When clicked, there would be an available list of items that he/she picked in the add tab, and it is arranged, by default, alphabetically. Each item in the list would contain course code, section, and corresponding day and time, all separated by spaces. For example, “Item 1” will have values: “CCSCAL1 S17 MW 1100-1230.” It can be arranged by date, time, section, or course code. This would allow the student to find if he/she got all the courses he/she needs for that term. If an item is double- clicked in the list, the information of the course pops out.

**Current Classes of Faculty:**



A professor can check what classes are currently assigned to him. Each item contains course code, section, day and time. For example, “Item 1” would display “CCSCAL1 S17 MW 1100-1230.” If an item in the list is double- clicked, a pop- up window comes out with the information of the class, how many students are currently enrolled, and the day and time of the class.