Animo.sys Enrollment System

A Field Report

for the course on

Introduction to Databases

(INTRODB)

Submitted by:

Aquino, Kurt Neil

Choy, Matthew Seaver

Hade, Alden Luc

Ms. Ethel Ong

Teacher

July 7, 2014

# **Introduction**

In answer to the ongoing clamor for enrollment and grade-viewing blues, the school administration introduced a new system in 2012. Called Animo.sys, this is an online enrollment system for students of De La Salle University that replaced the older My La Salle system. This can be accessed via animo.sys.dlsu.edu.ph.

The system was created mainly for convenience in enlisting for subjects for the next upcoming term. Animo.sys aimed to add new features to address the shortcomings of the older system, such as better and more efficient servers. In addition to the replaced computer system is the introduction of course codes.

Students use the system to enroll by first looking at a list of courses that they need to enroll in, and their corresponding course codes. These are seven-character strings used to uniquely identify courses (such as CCSALGE for Algebra in CCS, or COMALGE for Algebra in ). Beside the course codes are a list of room numbers with time slots that the user can freely choose from, and the class ID that uniquely identifies that room. If a room is not yet full, he/she can enlist in that course in that class.

The student does this for all the subjects he/she wishes to enroll in for the next term. As long as there are no overlapping time slots and full rooms, the user is guaranteed the class. This is the source of much of the problem in the system, because the number of students enrolled in a room can exceed the maximum due to an error in the application.

Initial reception of the system was negative. Many perceived it to be an unnecessary replacement for My La Salle, whose only alleged shortcoming was not being able to take heavy traffic to the servers at one time. Despite this, animo.sys continued to be used more and more in enlistment and enrollment. As mentioned earlier, the existence of the system provides incredible convenience for students because they no longer have to line up to enlist for classes. This is even more helpful for students who live far from Taft Avenue.

As more people enter in La Salle, the servers become constantly upgraded to keep up with the heavy traffic of students enrolling at the same time. It is gradually improving, and has proven to be more and more useful and necessary as time passes. The system is still used in the same way, requiring the user to identify the class ID, but now the class IDs are easier to be found since they are posted in multiple ways, either via My La Salle, or PDFs posted in Facebook.

## **Data Requirements**

* 1. **Software Features**

Animo.sys has three main menus: *Self Service, PeopleTools*, and *My Personalizations*.

*PeopleTools* is a set of tools that enhance, deploy, and extend PeopleSoft applications. Together with My Personalizations, both are menus that do not allow for much interaction with the student, and are mostly never used.

*Self Service* is the main feature of Animo.sys. This is where students can do everything they want to do pertaining to the upcoming term. They can view enrollment dates, see their current schedule and their upcoming schedule. This is also where the user can add, drop, or swap classes.

Viewing enrollment dates when not near the end of the term will result in a message that says, “You do not have access to enrollment at this time.” (See figure below.) Else, this screen will display the dates and times for enrollment for the different colleges, year levels, and dean’s listers.



Figure 1. Denial of access

Other screens include the *My Class Schedule*, which displays the class schedule for the upcoming term. Same with Enrollment Dates, this screen will show nothing useful if viewed at the start or in the middle of the term.

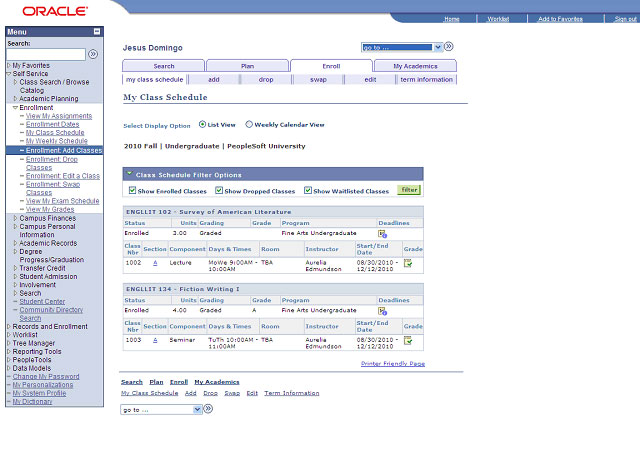


Figure 2. Class schedule

*My Weekly Schedule* shows the user’s current class schedule. Criticisms for this screen include a very cluttered display of courses, bordering on redundancy because some courses can have their name written multiple times. See figure below for clarification.

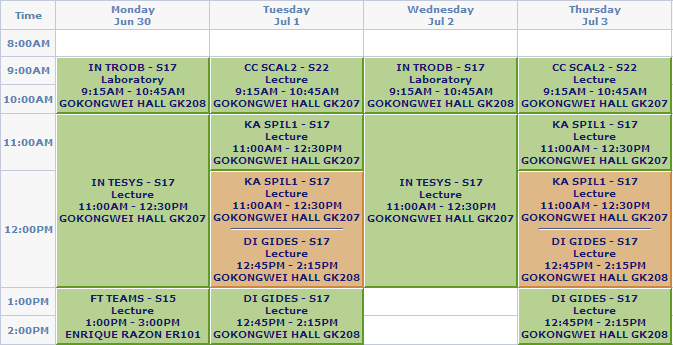


Figure 3. Weekly Schedule

In *Adding Classes*, the user can add classes if he knows their corresponding class numbers. If a list of classes is not available otherwise (usually PDFs containing class numbers are released before enrollment day), the user can search for a course right in Animo.sys.

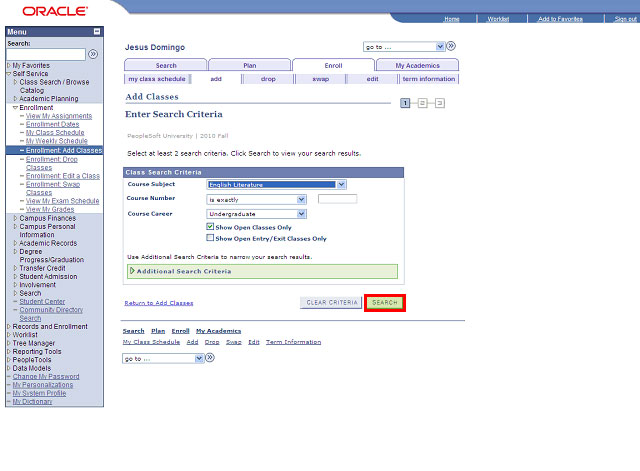


Figure 4. Add classes page

The same screens are used in *Drop Classes* and *Enroll Classes*.

* 1. **Database Design**

*Student Table* (table 1) is used to store the data of each enrolled student in the Amino.sys system. It contains the following fields:

* *student\_id* – 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 (dd/mm/yyyy)
* *status* – current status of the student (undergrad/graduate)

Table 1. Student Account Table

|  |  |  |  |
| --- | --- | --- | --- |
| **student\_id {PK}** | **last\_name** | **first\_name** | **birth\_date** |
| 11337974 | Aquino | Kurt Neil | 09/28/1996 |
| 11313803 | Choy | Matthew Seaver | 01/28/1996 |
| 11327278 | Hade | Alden Luc | 12/04/1996 |
| 11124345 | Connor | John | 02/28/1985 |
| 10725635 | Marley | Bob | 02/06/1945 |

|  |  |  |  |
| --- | --- | --- | --- |
| **email\_add** | **course** | **enroll\_date** | **status** |
| kurtaquino@yahoo.com | CS-ST | 05/20/2013 | Undergraduate |
| homenycablue@yahoo.com | CS-ST | 05/09/2013 | Undergraduate |
| aldstm@gmail.com | CS-ST | 04/15/2013 | Undergraduate |
| johnthebomb@gmail.com | CS-NE | 01/13/2011 | Undergraduate |
| bobsthename@hotmail.com | BS-IS | 02/29/2007 | Undergraduate |

*Faculty Table* (table 2) contains the information of each faculty member currently teaching in the specified school. It contains the following fields:

* *faculty\_id* – 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.
* *dept\_num* – contains a 5 digit number in order to identify which department the faculty member belongs to.
* *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 (dd/mm/yyyy)
* *to\_date* – date the faculty member has been either moved to another department or has retired/fired (dd/mm/yyyy. 99/99/9999 means current, to avoid using null value)

Table 2. Faculty Account Table

|  |  |  |  |
| --- | --- | --- | --- |
| **faculty\_id {PK}** | **dept\_num {FK}** | **last\_name** | **first\_name** |
| 00001 | D001 | Ong | Ethel |
| 00002 | D002 | Caronongan | Arturo |
| 00003 | D002 | Ilao | Joel |
| 00004 | D004 | Reyes | Tristan |
| 00005 | D005 | Benjie | Mendrado |

|  |  |  |
| --- | --- | --- |
| **email\_add** | **hire\_date** | **to\_date** |
| ethel.ong@dlsu.edu.ph | 03/05/2009 | 99/99/9999 |
| arturo.caronongan@dlsu.edu.ph | 10/25/2012 | 99/99/9999 |
| joel.ilao@dlsu.edu.ph | 04/12/2012 | 99/99/9999 |
| tristan.reyes@dlsu.edu.ph | 03/31/2007 | 99/99/9999 |
| benjie.mendrado@dlsu.edu.ph | 05/22/2009 | 99/99/9999 |

*Course Table* (table 3) contains all of the details of each available course currently being provided in the term. It contains the following fields:

* *course\_num* – 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
* *emp\_num* – employee number of the faculty assigned to teach in this specific course (99999999 for null)
* *section* – section in which the course is available to
* *sched\_time* - starting and ending time of the course for the day (mmhh – mmhh)
* *sched\_day* – 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* – current number of enrolled students to this course

Table 3. Course Table

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **course\_num {PK}** | **course\_code** | **course\_name** | **faculty\_id {FK}** | **section** |
| 313 | INTRODB | Introduction to Databases | 00001 | S17 |
| 3510 | INTESYS | Introduction to Intelligent Systems | 00002 | S17 |
| 2069 | DIGIDES | Digital Design | 00003 | S18 |
| 2101 | CCSCAL2 | Calculus 2 | 00004 | S11 |
| 3732 | FTTEAMS | Team Sports | 00005 | S15 |

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **sched\_day** | **sched\_time** | **room\_num** | **capacity** | **enrolled** |
| MW | 0945 - 1030 | G306 | 45 | 44 |
| MW | 1100 - 1230 | G207 | 45 | 41 |
| TH | 1430 – 1600 | G208 | 45 | 43 |
| TH | 1245 - 1415 | G210 | 45 | 34 |
| M | 1300 - 1500 | ER101 | 45 | 43 |

*Transaction table* (table 4) stores the information of the transactions being made by the student in the enrollment system. It contains the following fields:

* *transaction\_num* – unique 12 digit key or every transaction being made for the system to use
* *date\_stamp* – date when the transaction is being made (dd/mm/yyyy)
* *time\_stamp* – time when the transaction is being done (mmhh)
* *student\_id* – id number of the student doing the transaction
* *course\_num* – course number of the subject being processed
* *transaction\_type* – determines which course of action is being done
* 1 - adding a course
* 2 - dropping a course
* 3 - swapping a course

Table 4. Transactions table

|  |  |  |
| --- | --- | --- |
| **transaction\_num {PK}** | **date\_stamp** | **time\_stamp** |
| 0123-4567-8910 | 03/20/2014 | 0945 |
| 0198-7654-3210 | 03/12/2014 | 1400 |
| 1234-7895-6540 | 03/10/2014 | 1200 |
| 1532-7815-0584 | 03/01/2014 | 1600 |
| 2354-4179-5124 | 03/11/2014 | 2300 |

|  |  |  |
| --- | --- | --- |
| **student\_id {FK}** | **course\_num {FK}** | **transaction\_type** |
| 11337974 | 313 | 1 |
| 11313803 | 3510 | 1 |
| 11327278 | 2069 | 3 |
| 11124345 | 2101 | 2 |
| 10725635 | 3732 | 2 |

* 1. **Reports**

*List of Classes of a Student*

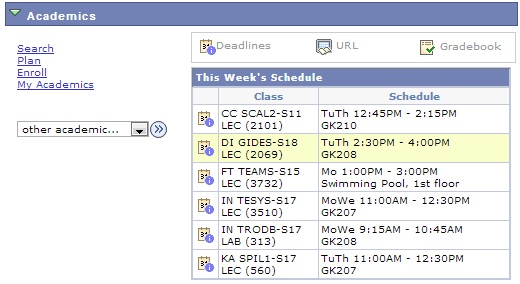
This report is used to generate the classes a student has in the current term. It can be used by the students and the university to check if the student is really enrolled in the class. The list is arranged alphabetically, as shown in Figure 5.

Figure 5. List of Classes of a Student

*Proposed List of Students in a Class*

This report will use Figure 5 as a guide to find the students who are enrolled in different classes for the entire term. This will help the faculty see the students who are originally enrolled in the class. Having the email address part of the table would not require the professors to ask anymore from the student. The professor inputs the class he is finding. The example of the table is shown email\_add in Figure 6.

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **Student\_id {PK}** | **Last\_name** | **first\_name** | **Email\_add** | **Course** | **Section** |
| 11337974 | Aquino | Kurt Neil | kurtaquino@yahoo.com | CCSCAL2 | S18 |
| 11313803 | Choy | Matthew Seaver | homenycablue@yahoo.com | CCSCAL2 | S18 |
| 11327278 | Hade | Alden Luc | aldstm@gmail.com | CCSCAL2 | S18 |

Figure 6. Propsed List of Students in a Class

select S.student\_id, S.Last\_name, S.First\_name, S.email\_add, C.course\_code, C.section from Student S, Transaction T, course C where S.student\_id = T.student\_id and C.course\_num = T.course\_num;

*List of Classes of a Course*

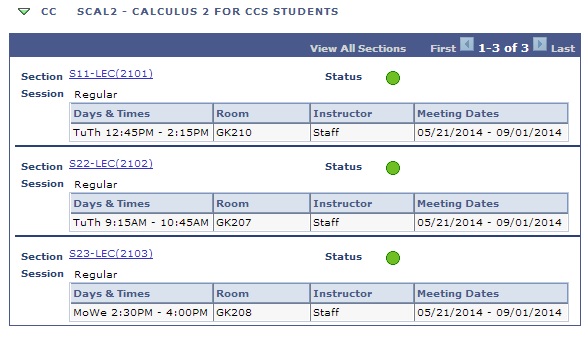
This report is used to generate the classes that can still be enrolled in a specific course. It can be used to guide the students on which classes to pick when enrolling. The list is arranged by section, as shown in Figure 7.

Figure 7. List of Classes of a Course.

*Proposed List of Classes of a Professor*

The report is used to generate the classes a professor would have in the term. This would enable the university to have an idea of professors who are having overload or underload work. Also, it gives professors an easier access to see which classes they will be handling, as shown in Figure 8.

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **Emp\_no {PK}** | **Last\_name** | **First\_name** | **Course\_num{FK}** | **Course\_code** | **Section** |
| 00001 | Ong | Ethel | 313 | INTRODB | S18 |
| 00001 | Ong | Ethel | 313 | INTRODB | S17 |
| 00001 | Ong | Ethel | 313 | INTRODB | S15 |

Figure 8. Proposed List of Classes of a Professor.

Select F.emp\_no, F.last\_name, F.first\_name, C.course\_num, C. course\_code, C. from Faculty F, Course C where C.emp\_no = F.emp\_no;

*List of Courses*

This report is used to generate the list of courses that are currently available in the enrollment system. It can be used to check what courses are available in the term currently, as well as, allowing students to choose which courses will be best for the next term. It is arranged alphabetically, as shown in Figure 9.

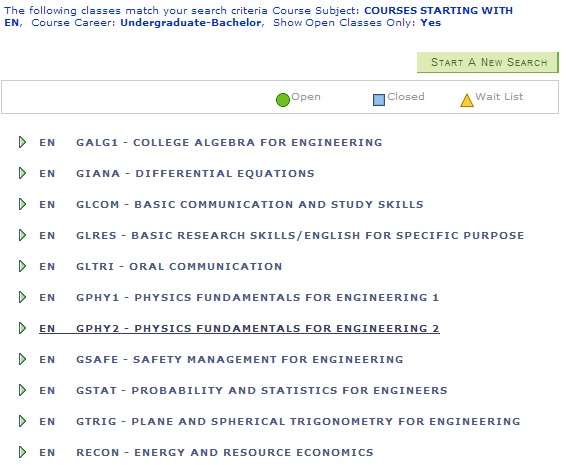


Figure 9. List of Courses.

*Proposed List of Available Classes in a Course*

This report is to be used by students to help them to find the courses that are still available. This would reduce the time students need to find the appropriate classes for them. The list is shown in Figure10.

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **Course\_code** | **Section** | **Slots left** | **room\_num** | **sched\_time** | **sched\_day** |
| INTRODB | S18 | 2 | G206 | 0915-1045 | MW |
| INTRODB | S18 | 10 | G206 | 1100-1230 | TH |

Figure 10. List of Available Classes in a Course.

Select Course\_code, section, (capacity – enrolled) as 'Slots left', room\_num, sched\_time, sched\_day from course where capacity-enrolled > 0;

1. **Other Submittals**

**3.1 End-user Survey**

Here are the tallies of the end-user survey given to students

1. **Which of the features of Animo.sys do you use?(Can choose all)**

* Add classes - 15
* Drop classes - 8
* Class Schedule - 10
* Swap classes - 4
* Term Information - 3

1. **Do you find Animo.sys' user interface easy to use?**

* Yes - 9
* No - 6

1. **ID Number(first 3 digits)**

* 113 - 11
* 112 - 3
* 111 - 1

1. **Any recommendations for the improvement of Animo.sys?**

* BIGGER SERVER BECAUSE IT IS LASALLE, STUDENTS (WE) PAY FOR IT. WE DESERVE BETTER. as always, ph sucks (113)
* Improve user interface and transaction process to attain better user experience in such a level as the old system (111)
* Less on Animo.sys but more on the enrollment times. I find it to be better to divide the enrollment times more so as to prevent more crashes and system failures. Having the entire 113 CCS student body can and will crash the system. Dividing that by having another criteria such as maybe by blocks would greatly help in lessening the burden on the servers. (113)
* more stable and faster. (113)
* Make it more like MLS(112)

1. **Have you encountered any problems with Animo.sys?**

* Yes - 6
* No - 9

1. **If yes, what problem(s) have you encountered?**

* The first time I enrolled using animosys, I was not allowed access even though it was my time to enroll. (113)
* Longer transactions (111)
* It wasn't as good as MLS (112)
* What sucked about it was it even bugged in the middle of enrollment when the graduate classes came in (113)
* wait list when class is full does nothing (113)
* the server is too weak/noob/sucks (113)

1. **In your opinion, how often do you think Animo.sys is being maintained? (choose one)**

* Yearly - 1
* Monthly - 3
* Weekly - 3
* Daily - 2
* Only when needed - 6

1. **Animo.sys schedules student enrollment times in order to limit data access and modification to avoid system failure and sever crashes, that being said, do you think Animo.sys will still be better if everyone can enroll at the same time?**

* Yes - 3
* No - 12

1. **How long do you think Animo.sys' server, system and database can manage to stay efficient and maintain proper usage? (choose one)**

* 1-3 years from now - 8
* 3-5 years from now - 2
* 5-10 years from now - 0
* As long as possible (with occasional maintenance) - 5

1. **Which system would you still prefer for enrollment?**

* Animo.sys - 7
* My.lasalle - 8

**3.2 Stakeholder’s Interview**

Here is a summary of the questions we asked Sir James Sy about the Animo.sys enrollment system and database

1. **What is the purpose of having a database in our enrollment system?**

* Can you imagine doing systems without a database? A database is a collection of data.

1. **How does data relate to the end user's needs?**

* Based on the needs of the users. You will be taught later on in INTRODB on how to make data into physical view. Given a form, we need certain data to be part of the data. We have searching needs, etc. The data really depends on the users, and how you're going to structure your database.

1. **When the data is encoded in the system, is there a way for the student to change the data? Is there a schedule for it?**

* Yes, there is a schedule for changing the data. It is actually in the application level of the program not the database itself.

1. **What are the data involved in the enrollment system?**

* Let's talk about tables and what tables are needed. Tables are as follows enrollment, student, class, course, term, facility, faculty. Normalization is very important. It is used to decrease the amount of data needed and takes out redundant data in the database. This would allow a much more relative database, and a much more efficient database.

1. **Does animo.sys consider the pre-requisites, etc?**

* As of now, it has not yet been activated. They said that it will slow the system down, so we are still testing these features.

1. **Who was the creator of animo.sys?**

* Animo.sys was bought from PeopleSoft. They chose oracle instead of SQL because SQL was shaky in the earlier years.

1. **When was Animo.sys launched?**

* It first started May 2013 which failed, then on November 2013, it was relaunched.

1. **How much bytes are needed to handle the enormous amount of data?**

* Actually, since hard drives are cheaper, it would not matter too much. Since your ID number only needs a few bytes, we can actually know a lot about you with just 1000 bytes.

1. **How did you give access rights to the users of the database system?**

* We did it by assigning each person a role. In a sense, these roles gives them the right to access the data. This would make things easier since you do not have to change the data a person can access, instead only change the role of that person. This implementation allows an easier revising of the code. For example, if I have to change something about the student's role, I won't have to change all of your access, but instead change the access of the student.

1. **Ms. Ethel mentioned that sometimes memory of the database are put offline? Is that the case with Animo.sys? Why or why not?**

* Again, with the issue of hardware storage, since it is cheap, we do not need to have offline storage anymore. The only reason why there are offline storages before was because memory was expensive.

1. **Is there something you would like to modify the Animo.sys?**

* I wouldn't dare modify it. Maybe customizing it is a possibility. Aside from that, there are still tons of features that we have not been able to explore to make sure its stability.

1. **Are there derived data in the system?**

* As much as possible, database should not have derived data. The problem is, whenever the data that would be computed is changed, the derived data would not change. It would take a while before the derived data is changed which is dangerous. As much as possible, derived data should be done minimally.

1. **What errors are we supposed to watch out for in the making of this database?**

* Misconfigurations mostly.

1. **How would you suggest we create this database?**

* It actually depends on you. It depends on how big you want your database to contain. To be honest, three tables are already enough to create a simple kind of this database. If you want to take note of the faculty, put a faculty table. It goes as big as you want.

1. **References**

* Sir. James C. Sy, Director of the Information Technology Office
* Student of the College of Computer Studies
* Animo.sys official webste, <http://animo.sys.dlsu.edu.ph/ps/signon.html>
* My.Lasalle official website, <https://my.dlsu.edu.ph/>

Mano, M., & Ciletti, M. (n.d.). Digital Design, 5th Edition by M. Morris Mano and Michael Ciletti.pdf - Google Drive. Digital Design, 5th Edition by M. Morris Mano and Michael Ciletti.pdf - Google Drive. Retrieved July 6, 2014, from https://docs.google.com/file/d/0B9LJy8vattSMeWxOMDl1Sk43Sjg/edit?pli=1

Stephens, R., & Plew, R. (2001, January 1). Database Design. . Retrieved July 6, 2014, from http://www.dtic.co.cu/FTP/libros/database\_desing.pdf