**2.0. Overall Description**

***2.1. Product Perspective***

AE is a web based enrollment software being designed for the PUPR whose main purpose is to facilitate the online enrollment process for the students of the PUPR. Our software in comparison to the current system which is also another web based enrollment software known as “MyPoly” is much more reliable, organized and ergonomic for our target users, the students of the PUPR making their enrollment process a quicker, easier, secure and more satisfying one . MyPoly has been known for several issues and complaints based on feedback coming from several students of the PUPR . These issues and complaints are generally based on how MyPoly’s system performs it’s enrollment process for the students. These problems can be summarized in the following also showing how AE targets these problems:

* No tolerance on user input for the system’s search engine or direct course enrollment:

The process of course searching in order for the students to enroll can take about 30 seconds to 1 minute to fill out the search parameters and to make matters worse it can easily fail which will prompt you to fill out again the search parameters from the beginning which makes this process alone a tedious and repetitive one . As for a direct entry of a course enrollment the system requires the student to enter the course code and the desired section in order to skip the course search process but this method tolerance towards the data input given by the student is very low, as the entry is case and space sensitive. This forces the student to enter the course code and section exactly as the system expects the student to enter the information regarding case sensitivity and space between text making this method highly error prone for the student to use.

AE targets the described problem by querying the PUPR ‘s DBMS and automatically showing the students the course code and name the students are able to enroll for the following trimester based on the retrieved information from the DBMS. Eliminating the need of a manual search engine and showing the students the necessary information for them to begin their enrollment process.

* Course and course section selection process is complicated and error prone.

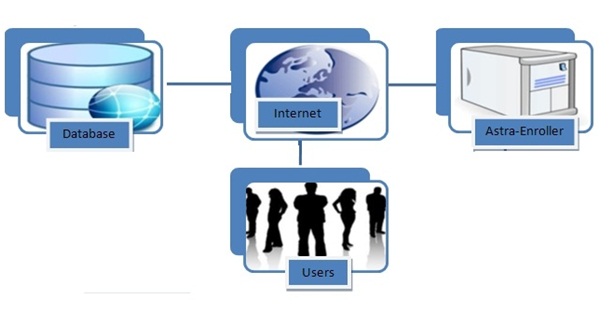
The course and course section selection process is unorganized as it shows the entire results of the search done by the students on MyPoly, most of the time students need to search per department in order to get a result due to the search engine issues aforementioned. Students are forced to view several pages that include the courses and their respective sections per course also while the student is forced to choose directly which course section to enroll without displaying essential information such as sections the student can’t enroll due to time conflicts such as two different courses being given at the same time slot. And unless the student is completely aware of his current enrollment status which is only shown on the main page, they can accidentally choose a course section that may have a time slot conflict, the system will deny the request and the student has to start the whole selection process from the beginning.

AE in this perspective of course and course section selection process, is much more simple for the students. All the students have to do is click on the course they wish to enroll and AE shows them a window with the course sections available as well as highlighting sections that have a time slot conflict with already enrolled courses. This keeps the enrollment process for the students clean and simple.

As it can be seen, AE helps the students by making their course enrollment process much more simpler, faster and secure than the current system. Staying with the software’s purpose the following sections show how the software operates inside various constraints.

***2.1.1 System interface***

AE being a web based software, it implies that it is part of a much larger system in which the software works. The main system in which AE takes part of would be composed of *The Internet, The PUPR DBMS, Astra-Enroller* and *The Users.* The main component is the Internet which will act as the means where the other components will be able to communicate with one another. AE will communicate with its target users the students, which will give AE the necessary inputs so that the system may begin performing its tasks requested by the user. AE via Internet will establish communication with the PUPR database using the student’s credentials and the input provided by the student to retrieve information from the DBMS for AE to process.



***2.1.2 User Interface***

AE will provide the students with several web based interface views which the students will interact with in order to enroll his or her courses online. It is expected that students should be able to learn to use the system without any previous experience with at most 20 minutes using the AE software. The following will show the required user interfaces that are necessary for the students to be able to complete enrolling their courses online using AE.

* User Login Interface

This is a necessary user interface as it is the process where the students enter their credentials given by the PUPR into the system in order to verify that the student belongs to the PUPR and it is entitled to enroll courses. This interface will have two textboxes one for the username , another for their password which are credentials a student must have in order to access AE and these are provided by the PUPR services on the CTE building. The students after entering their credentials AE will proceed to verity and when confirmed will show a welcome message to the student in question. This is a login user interface example and also the welcome message upon success of login into AE system which will proceed the user to the main page.

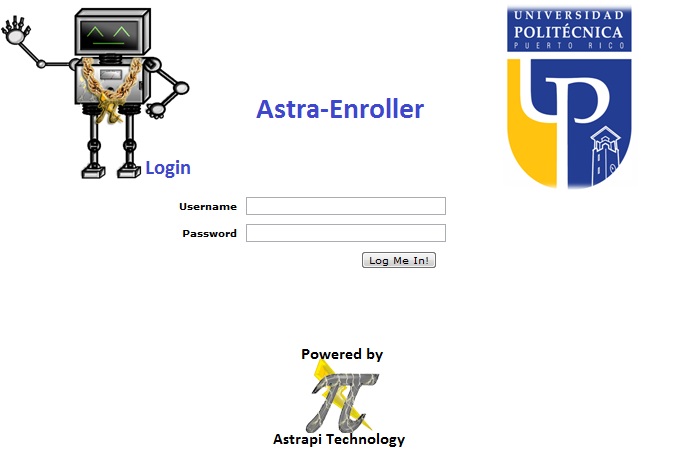


Figure 2.1.2.1 Login interface



Figure 2.1.2.2 home page welcome message

* Main Page User Interface:

The main Page interface will be shown after the user logs in successfully to AE and will consist of several options buttons on the above part of the page display which will begin performing different tasks for the students.

* Home button:

Has no main functionality other than greet the student to the system while the software awaits the task the student wishes to perform.

* Enroll button:

This button will be selected by the student when they wish to enroll courses. Upon the user clicking on this button AE will prompt to show them which courses the students are able to enroll into. The student will select the course he wishes to enroll by clicking and another display showing the section courses with radio select buttons where the student will click to tell the AE the section of the course they wish to enroll. Also this display will show the user in red, sections of a course the student is unable to enroll into due to reasons such as time slot conflicts with another course, or the student having a active debt to the PUPR during the time frame of course enrollment for future trimesters. To confirm the course enrollment the user must click on the enroll button shown below the course section display. Also if the course selected for enrollment is a course already enrolled by the student the system will show the selected section and a drop button option if the student decides to drop the selected course section. The following are examples of how the enroll interface will be displayed to the user:

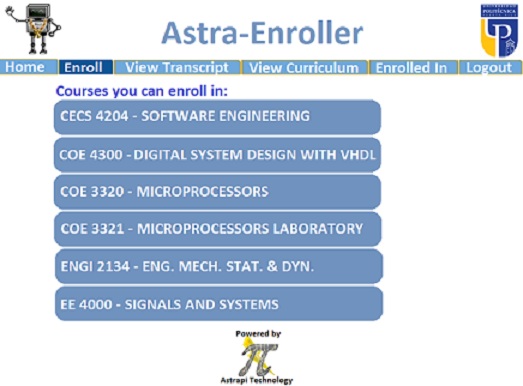


Figure 2.1.2.3 Enroll Courses display

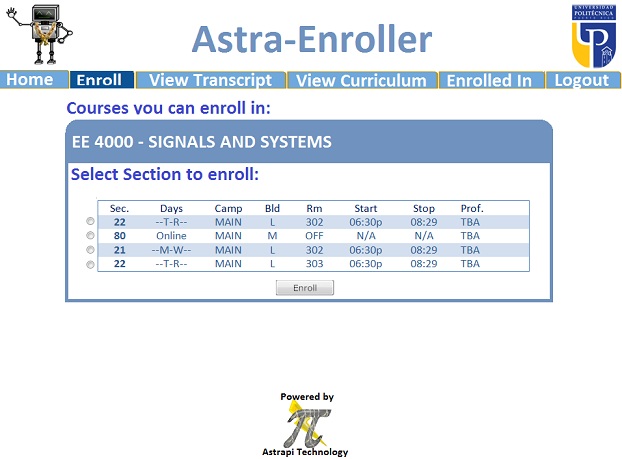


Figure 2.1.2.4 Enroll course section display

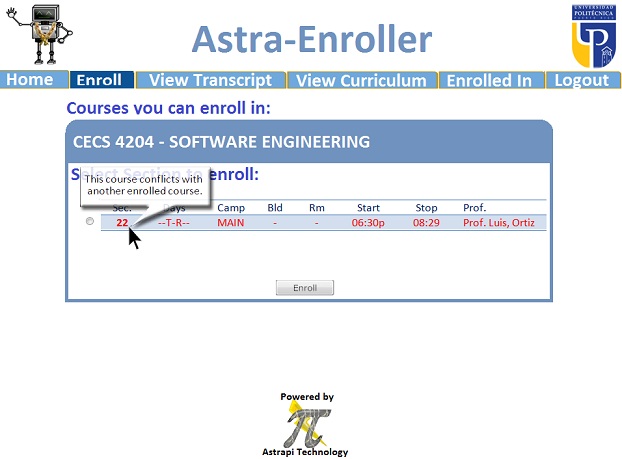


Figure 2.1.2.5 Enroll course section unselectable example

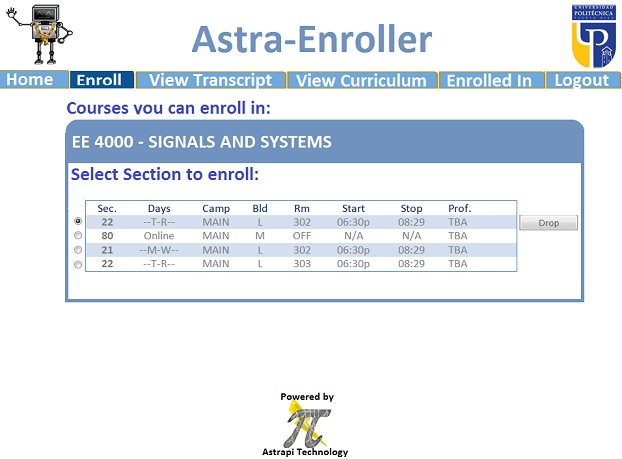


Figure 2.1.2.6 Enroll course section drop course example

***2.1.3 Hardware interface***

The hardware components necessary in order for AE to function or be used can be listed as the following:

* Server computer:

This is necessary as the server will be the key component where AE and the webpage for AE will be implemented and be kept online . The server must have the capacity to process information incoming from 10000 students maximum Using the AE software to enroll their courses online. The server is also required to be operational at all times except when the server is receiving maintenance or a software update.

* Computer /Laptop

AE requires the students to have access to a computer or laptop with a modem capable of connecting to the Internet in order for the student to be able to access and use AE. Computers are required to be connected to the Internet at a speed of 5.6 kbps also as minimum hardware specification requirements in order for computers to use AE would be the following, a processor that runs at a speed of 233 MHz. This is necessary in order to achieve the desired performance for the software.

***2.1.4 Software Interfaces***

Additional software required for students in order to use AE are the following:

* OS:

Any OS that is compatible with the supported web browsers that are AE compatible can be used in order to access AE.

* Web Browser (Mozzila Firefox or Google Chrome):

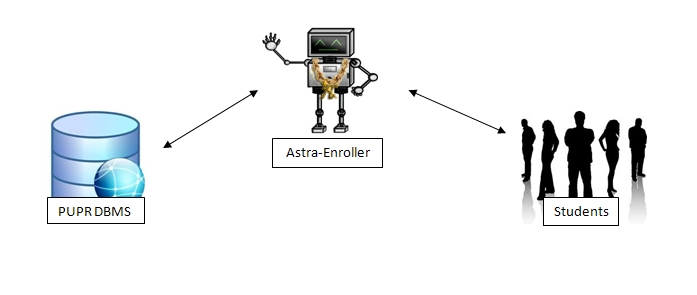
Students in order to access AE must use the aforementioned web browsers which are supported for AE. Otherwise the software may not run or perform tasks correctly or as intended risking the integrity of the information and the processes that will be performed by the software. Future compatibility support for other web browsers may be implemented on future software updates.

* DBMS

This is required, as AE in order to function must be able to communicate with a DBMS where the software will gather essential information in order to complete its designated tasks. AE will be able to establish communication with common high end DBMS such as ORACLE, MySQL and Microsoft Access. Other implemented DBMS could result in compromising the integrity of the tasks performed by AE.

***2.1.4 Software interfaces***

Our software receives the credentials by the user and validates it with the database. To achieve this interface is at least require that the user have a computer and a Web browser. The computer should have an OS that can support Mozilla Firefox and/or Google Chrome web browsers.



*Figure 2.1.4.0 Software Interface*

***2.1.5 Communications Interfaces:***

Astra-Enroller will be running on the PUPR server, this will make updating the database much easier and accurate. Since this is the case Astra-Enroller which will contain everything related to Astra-Enroller and their website. The students will communicate with the server using the internet. For managing communications we’ll use TCP/IP.

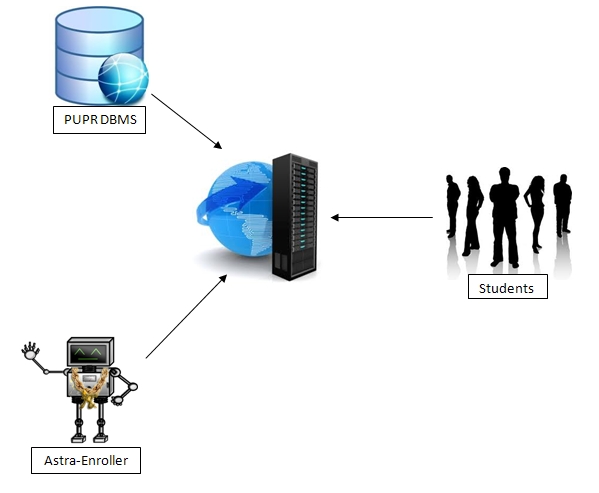


Figure 2.1.5.0 Communication Interface

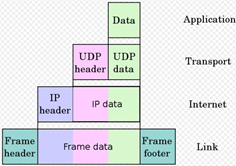


Figure 2.1.5.1 Different layers of TCP/IP

***2.1.6 Memory Constraints***

The minimum memory requirements for students to be able to use AE in their computers are, 64 MB of RAM, at least 250 MB of space on the computer’s Hard Disk storage device. These requirements aforementioned are essential to preserve the integrity of AE’s performance and having lower memory may result in severe loss on AE’s performance or incapability to complete its functions, tasks and interaction with its users. Vice versa computers with higher memory may see increased performance in terms of information processing or reduction of time needed to display the users the results of their instructions given to AE for example ( a student with a computer of 1 GB of RAM and 128 MB of video memory will see results in a approximate of 30 seconds faster than another user with the minimum requirements).

***2.1.7 Operations***

Astra Enroller has one general mode of operation which is for the users to enroll courses. These operations are defined by the following steps:

1. User logs in (enter credentials) on the system.
2. Validate the user and give permissions.
3. System validates if the user can enroll in that moment.
4. System validates what courses the student can enroll in, this includes if there are any requirements.
5. The system will display the name and course code of the courses that the user can enroll into at this moment or the user’s credit transcript depending on the user’s command.
6. The user will select one of the display courses.
7. The system will list the available sections for the selected course. Will validate and show the user any conflicts.
8. If the user can enroll the system will enroll the course. If not the system will reject the registration. This includes returning error messages.
9. If the user course was enrolled then user will be prompted back to step 5 and the user repeats these steps until the user concludes his course enrollment. Else if the system rejects the enrollment the user is still prompted back to step 5 as well with the error message indicating the process did not conclude. Also the system generates a statement of the currently enrolled courses for the user.
10. Once users are done with their course enrollment then they may log out or close the application on the web browser to terminate connectivity with the system.
11. The system will proceed to log them out and conclude services if the user logs out by request, terminates connection with the system or remains inactive for 10 minutes.

***2.2. Product Functions***

The main purpose of the Astra Enroller system is to facilitate the tedious process of enrollment at the University, by having a better validation system and a much faster and user friendly way of enrolling each trimester courses in general. It provides functions that help the student to minimize time consumption and money.

***2.2.1 Student use case:***

* + Access Astra-Enroller Web Page

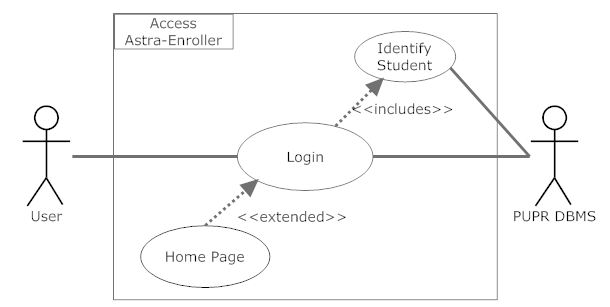
******

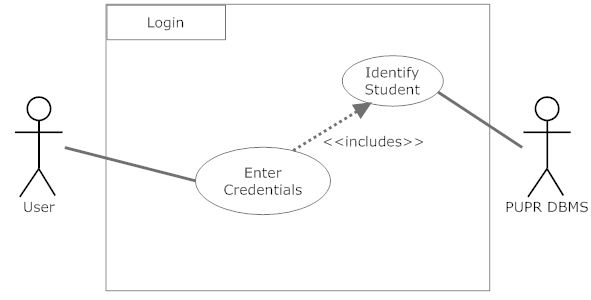
Diagram 2.2.0

***Description:***

The user access the Astra-Enroller Web-Page using a required browser. Once the user attempts to access Astra-Enroller, this user will be directed to the login page of which the user will enter there credentials. Once the user has entered there credentials and agreed to proceed (this could be with the use of a button title “Enter”), AE will request validation to the PUPR DBMS, if valid then the user can proceed to the AE Home Page.

***Initial Step-by-step description:***

* + - 1. The user connects to the Astra-Enroller Web Page using a browser and Internet.
      2. The Astra-Enroller directs the user to the login page and ask the user to enter credentials.
      3. Once the user agrees to proceed then AE requests PUPR DBMS for user validation.
      4. If user is valid then the user will be re-directed to AE Home page.
  + Login

Diagram 2.2.1

***Description***:

When this function is activate it will wait for the user to enter his or her required personal credentials. Once the user proceeds the enter information(credentials) will be validate by the PUPR DBMS. If valid then the user will be logged in.

***Initial Step-by-step description:***

* + - 1. The student logs in AE by filling the username and password personal credentials given by the university.
      2. The Astra-Enroller server creates a query to request the DBMS to validate the credentials and returns the results.
      3. If the credentials are valid then the user will be logged and can access AE.
  + Display Enrollment List

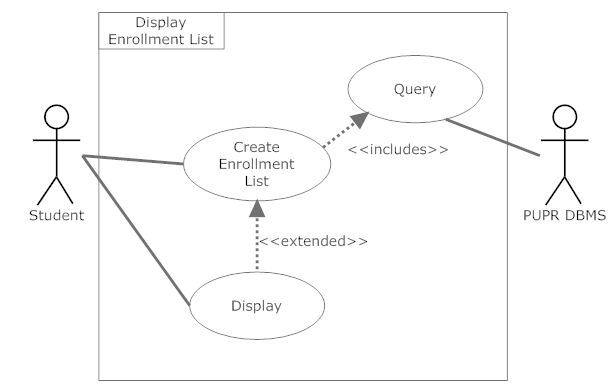


Diagram 2.2.2

***Description***:

When the user, in this case a student requests to view the courses to enroll, the display function will be activated. This means that AE will start the internal function Create Enrollment List using the students credentials. The Create Enrollment List will create and send a Query of which the PUPR DBMS can understand. Once the PUPR DBMS receives the Query it will return the results to Create Enrollment List which in return will start displaying each course.

***Initial Step-by-step description:***

* + - 1. Student selects to view course he or she can enroll in, this activates Display Enrollment List.
      2. Once Display Enrollment List is active, it will start Create Enrollment List.
      3. Create Enrollment List will generate a query to requests the PUPR DBMS for results.
      4. Once the PUPR DBMS has received the query then it will return the results to Create Enrollment List.
      5. When Create Enrollment List has received the requested information it will start displaying it to the user.
  + Create Enrollment List

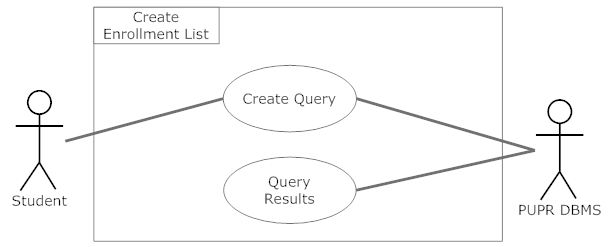


Diagram 2.2.3

***Description***:

The Create Enrollment List function will simply create a query depending on the user’s credentials, the reason for this is to ensure that the PUPR DBMS is comparing the transcript and curriculum of that particular student. The created Query will be sent to the PUPR DBMS. Once the DBMS has processed the received Query it will return it’s results.

***Initial Step-by-step description:***

* + - 1. Create a query using the students credentials.
      2. Send the created Query to the PUPR DBMS.
      3. PUPR DBMS will return the results.
  + Select Course

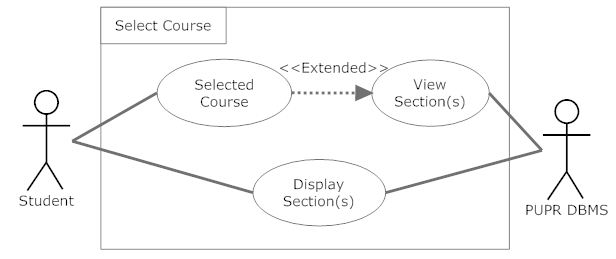


Diagram 2.2.4

***Description***:

After successfully loading the Enrollment page the Student will be able to select a course. When the Student selects the course that he or she wants to enroll in the View Section(s) function will be activate. View Section(s) will communicate with PUPR DBMS so the section(s) can be displayed.

***Initial Step-by-step description:***

* + - 1. User clicks on a course from the list.
      2. The users actions will activate the View Section(s) function.
      3. View Section(s) will communicate with the PUPR DBMS.
      4. Once the PUPR DBMS receive requests from View Section(s) it will return the result so that they can be displayed..
  + View Section(s)

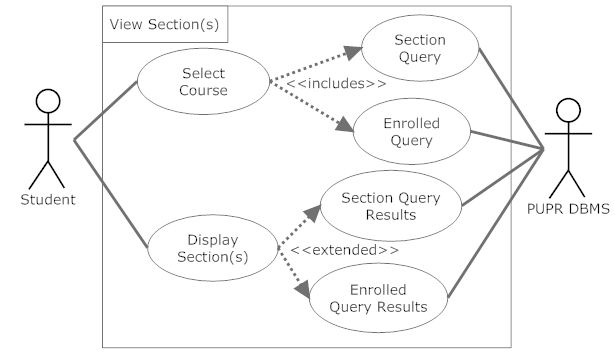


Diagram 2.2.5

***Description***:

Once the student has selected a course then this function will be active, which means there will be a list of sections that can be selected. When the user selects a section for the selected course, AE will create a two queries, one to get all sections available for the particular course (Section Query) and the other to get all course and sections the Student is currently enrolled in (Enrolled Query). These queries will be sent to PUPR DBMS. The PUPR DBMS will process each query and send the results. When the DBMS has sent the results then these results will be displayed for the user.

***Initial Step-by-step description:***

* + - 1. User selects course.
      2. AE will create a two queries and request the PUPR DBMS for results of each query.
      3. The PUPR DBMS will receive the request and process it.
      4. Once the PUPR DBMS has returned the results then these results (sections) will be displayed.
  + View Curriculum

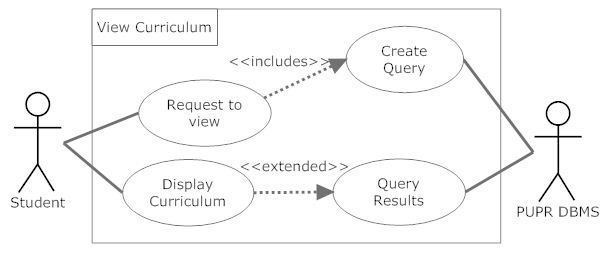


Diagram 2.2.6

***Description***: The view curriculum function will display all the courses the student needs to take for his or her concentration. A query is created depending on the user’s credentials and sent to the PUPR DBMS. The PUPR DBMS will process and return the results. Once the PUPR DBMS has returned the results then the user can see his or her curriculum.

* + View Transcript

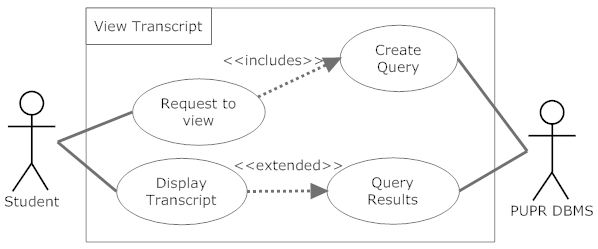


Diagram 2.2.7

***Description***:

The view transcription function will display all the courses the student has taken from the PUPR. A query is created depending on the user’s credentials and sent to the PUPR DBMS. The PUPR DBMS will process and return the results. Once the PUPR DBMS has returned the results then the user can see his or her transcript.

***2.3 User Characteristics***

Since the main features of AE are in the interfaces on the web page the user characteristics or skills are not too high. Most of our users, if not all, are students and they already have the necessary skills to use our system. To mention the “skill” that the user needs is to have experience in navigating the web. But in case one of our user haven’t use a web-page before our interface would be detailed enough to guide the user through all the steps.

***2.4 Constraints***

* + *Hardware and Software limitations*
    - Processor Minimum: 233 MHz
    - Memory Minimum: 64 MB RAM
    - *OS Minimum: None (user can use any OS that can support Mozilla Firefox and/or Google Chrome)*
    - *Web browser: Mozilla Firefox or Google Chrome*
    - Internet connection with at least a speed of 1 kbps

* + *Languages needed for implementation*
    - PHP
    - MySQL

* + *Security*
    - HTTPS with Encrypted SSL and TSL connection features.

* + *Others*
    - Depends on the database that the institution uses.

***2.5 Assumptions and Dependencies***

* Users Belong to the PUPR and have an account already created on the CTE in the PUPR campus, otherwise they will not be able to access AE.

It is assumed that the corresponding users have created this account on the campus. This is because these credentials [user name and password] will also be used as the log in credentials for the user in order for them to access AE . Otherwise users will not be able to access AE and it’s services.

* The PUPR DBMS being used is one of the following: MySQL, Oracle and Microsoft Access:

It’s necessary to take this assumption since AE will attempt to communicate with the Campus database interacting with instructions based on one of the aformention DBMS SQL queries. If the campus decides to change the DBMS then AE has to be changed accordingly to be able to communicate on a common ground between the campus’s DBMS. Otherwise AE will not function properly.