**Table of contents**

1.0 Introduction...................................................................................................... 3

1.1 Purpose................................................................................................. 3

1.2 Scope.................................................................................................... 3

1.3 Definitions, acronyms and abbreviations............................................ 5

1.3.1 Definitions.............................................................................. 5

1.3.2 Acronyms............................................................................... 6

1.3.3 Abbreviations......................................................................... 6

1.4 References............................................................................................. 6

1.5 Overview............................................................................................... 7

2.0 General Description........................................................................................... 8

2.1 Product Perspective............................................................................... 8

2.2 Product Functions.................................................................................. 21

2.3 User Classes Characteristics.................................................................. 30

2.4 Constraints............................................................................................. 30

2.5 Assumptions and dependencies............................................................. 31

3.0 Specific Requirements....................................................................................... 32

3.1 External Interface................................................................................... 32

3.2 Product Functions.................................................................................. 37

3.3 Performance Requirements.................................................................... 46

3.4 Logic Requirements of Database........................................................... 47

3.5 Design restrictions.................................................................................. 49

3.6 Software System Requirements.............................................................. 49

# List of Figures

|  |  |  |
| --- | --- | --- |
| Figure | Description | Location (Section) |
| 2.1.1.0 | System Interface | 2.1.1 |
| 2.1.2.1 | User Login Interface | 2.1.2 |
| 2.1.2.2 | Home Page welcome message | 2.1.2 |
| 2.1.2.3 | Enroll course display | 2.1.2 |
| 2.1.2.4 | Enroll course section display | 2.1.2 |
| 2.1.2.5 | Enroll course display unselectable example | 2.1.2 |
| 2.1.2.6 | Enroll course display drop course example | 2.1.2 |
| 2.1.4.0 | Software Interface | 2.1.4 |
| 2.1.5.0 | Communication Interface | 2.1.5 |
| 2.1.5.1 | Different layers of TCP/IP | 2.1.5 |

# List of Tables

|  |  |  |
| --- | --- | --- |
| Table | Description | Location (Section) |
|  |  |  |

# List of diagrams

|  |  |  |
| --- | --- | --- |
| Diagram | Description | Location (Section) |
| 2.2.0 | User case for Accessing Astra-Enroller. | 2.2 |
| 2.2.1 | User case for Login. | 2.2 |
| 2.2.2 | User case for Displaying Enrolment List. | 2.2 |
| 2.2.3 | User case for Create Enrolment List. | 2.2 |
| 2.2.4 | User case for Select Course. | 2.2 |
| 2.2.5 | User case for View Section(s). | 2.2 |
| 2.2.6 | User case for View Curriculum. | 2.2 |
| 2.2.7 | User case for View Transcript. | 2.2 |
|  |  |  |

# 1.0. Introduction

## 1.1. Purpose

The purpose of the Software Requirements Specifications (SRS) is to present in an organized and clear way the requirements of the system (AE). It will explain the purpose of the system called Astra-Enroller, its interface and how it works in conjunction with other systems. Included in this document are the features and a explanation of what it is going to do and not, like for example our system won’t be able to be access for any students if they don’t have Internet. The SRS is directed to clients, any person with an interest in the software; users, a student from the PUPR who will use this software; and the developers of the system, does responsible in created, designing maintenance and making updates to the system, Astrapi.

## 1.2. Scope of Project

The name of our enrollment system is Astra-Enroller. This system will be capable of the enroll students with far more ease, speed and precision than the current system on the Polytechnic University of Puerto Rico, known as MyPoly. Astra Enroller will offer great benefits versus the current system, for its various target users (students of the Polytechnic University of Puerto Rico) such as the following:

* Faster and more organized enrollment options for the students
* Showing them the courses they may enroll into
* Showing them the taken courses with low grades, from C to F including withdraws.
* Showing them the course description
* Showing them the special topic courses.

These previous benefits would reduce the need of the students to settle their course enrollment process on the campus. Also our software will facilitate the enrollment process so the enrolling students will have all the information needed to enroll on our software. Simply put, our main goal is to simplify and accelerate the enrollment process for the students, that being said we have future goals that could be added to Astra Enroller such as:

* Grant professors access to Astra Enroller this includes:
  + The ability to add final grades online.
  + Generate a role book for each course and section.
* Financial Services Improvement:
  + The option to chose parking.
  + Facilitate the payment options.
* Validate Health Care plan online.

## 1.3. Definitions, acronyms and abbreviations

*1.3.1 Definitions*

|  |  |
| --- | --- |
| Term | Definition |
| Database | Collection of all the information monitored by this system. |
| Student | A student of the Polytechnic University of Puerto Rico. |
| Software Requirements Specification | A document that completely describes all of the functions of a proposed system and the constraints under which it must operate. For example, this document. |
| Client | Any person with an interest in the project who is not a developer. |
| User | A student of the Polytechnic University of Puerto Rico or a Admin. |
| Astrapi | Greek for the combination of lightning and pi |
| Astra | Lightning in Greek. |
| pi | Represent the Greek letter and also known as a mathematical constant whose value is the ratio of any Euclidean plane circle's circumference to its diameter. |
| Astra-Enroller | The name of the software to be developed. |
| Web-Page | A document on the World Wide Web or simply put on the Internet. |
| Our | This will be directed to Astrapi Technology team working on the Astra-Enroller Software. |
| Query | A form of questioning, in this case language a DBMS can understand of which we can ask for certain data and it will return the answer/results. |
| Administrator | In this case a user who is allowed access to delicate information. Usually is a user who can give maintenance to a system and even update/modify it. Most likely to be a member of Astrapi Technology. |
| We | This will be directed to Astrapi Technology team working on the Astra-Enroller Software. |

## 1.3.2 Acronyms

|  |  |
| --- | --- |
| Term | Acronym |
| Polytechnic University of Puerto Rico. | PUPR |
| Software Requirements Specification. | SRS |
| Database Management System. | DBMS |
| Astra-Enroller. | AE |
| Astra-Enroller Server. | AES |
| Uniform Resource Locator. | URL |
| Application Programming Interface. | API |
| Entity-Relationship Diagram. | ERD |
| Operating System. | OS |
| Hypertext Transfer Protocol Secure. | HTTPS |
| Transport Layer Security. | TLS |
| Secure Sockets Layer. | SSL |
| Transmission Control Protocol. | TCP |
| Internet Protocol. | IP |
| Administrator. | Admin |
| Hypertext Preprocessor. | PHP |

1.3.3 Abbreviations

|  |  |
| --- | --- |
| Term | Abbreviation |
| “And so on” or “et cetera” | etc. |
|  |  |

## 1.4. References

* IEEE. *IEEE Std 830-1998 IEEE Recommended Practice for Software Requirements Specifications.* IEEE Computer Society, 1998.
* [christopher.jones](http://wiki.oracle.com/account/christopher.jones). “*PHP Oracle FAQ”.* Oracle Wiki.April 17, 2011. <http://wiki.oracle.com/page/PHP+Oracle+FAQ>.
* No General Author. “PHP”. Wikipedia. April 17, 2011.

<http://en.wikipedia.org/wiki/PHP>.

***1.5. Overview of Document***

The following chapter of this document is the Overall Description. This section describes what the software can do or cannot do. The Requirements Specification section describes the software’s requirements in an informal and simple matter, with the product perspective, functions, user characteristics etc. This section is then used as the guide to implement technical and more complex requirements at the requirements specification section. The intended audience for the Overall Description of the SRS document is our client, Polytechnic University of Puerto Rico. The next chapter talks about the requirements specification section. It contains the description of the software’s requirements and functionality in a much more technical and robustly detailed manner intended for the developers or experts on the field.

**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 system will be part of the PUPR system, since it depends on the info from the database. Our software in compare with MyPoly is much more reliable, organized and ergonomic. The problems with MyPoly are:

* The Search engine is not very reliable for searching courses.
* The Search engine is caps sensitive.
* The students need to have his curriculum at hand when selecting courses.
* Courses that the student have pass are not validated and he can’t enroll on another course needs it as a pre-requisite.

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.

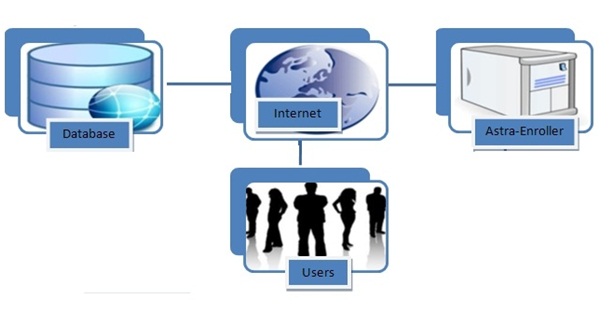


Figure 2.1.1.1

***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. With at least 20 minutes the students should be able to learn how to use the system. 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.

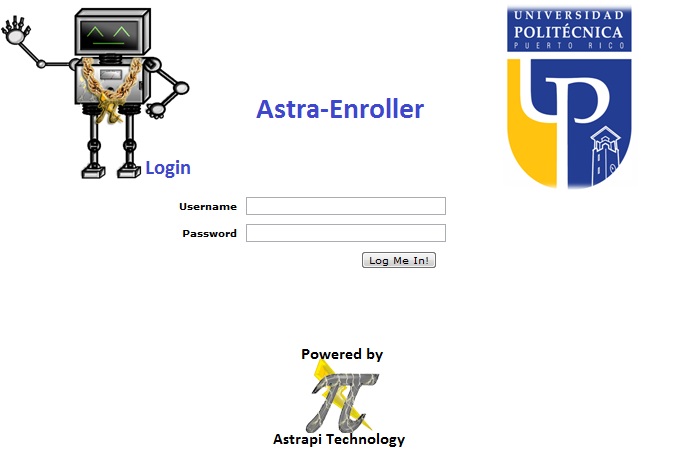


Figure 2.1.2.1 Login interface



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:

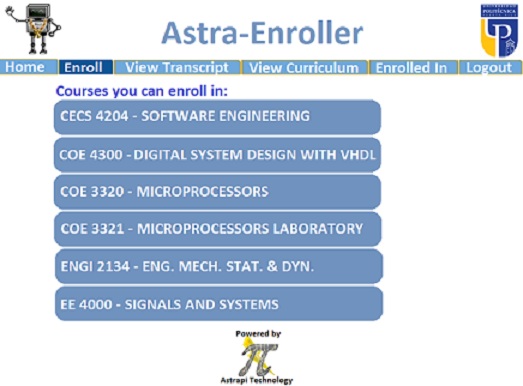


Figure 2.1.2.3 Enroll Courses display

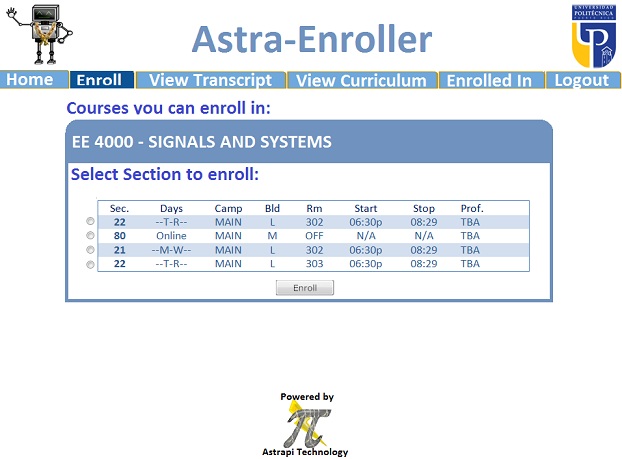


Figure 2.1.2.4 Enroll course section display

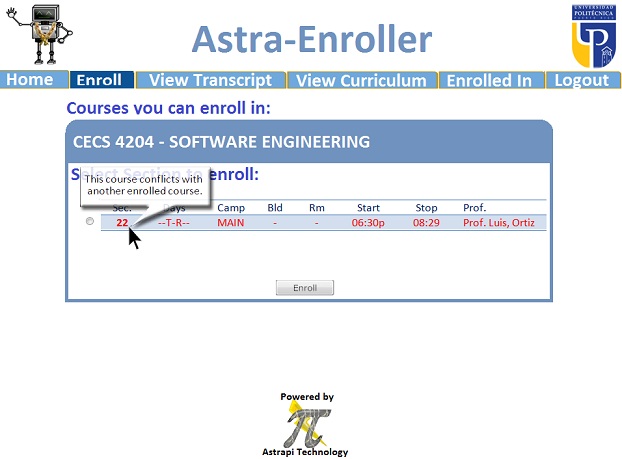


Figure 2.1.2.5 Enroll course section unselectable example

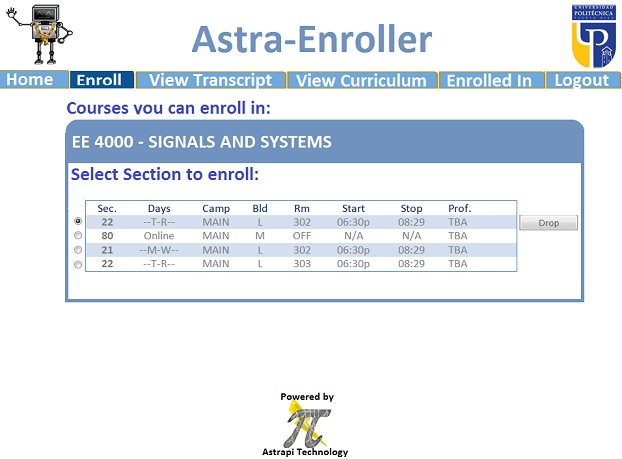


Figure 2.1.2.6 Enroll course section drop course example

***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:

An OS (Windows XP, Mac OS X) 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):

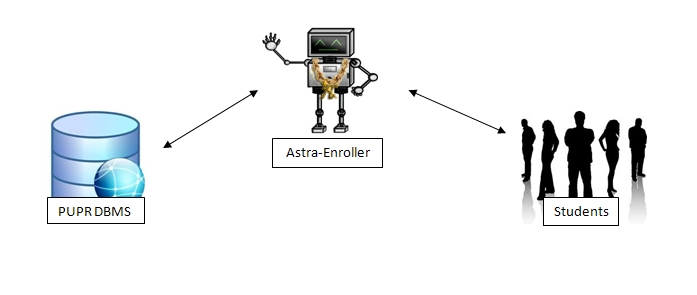
Students in order to access AE must use the aforementioned web browsers which are supported for AE. Otherwise the software may not run or perform tasks correctly or as intended risking the integrity of the information and the processes that will be performed by the software. Future compatibility support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

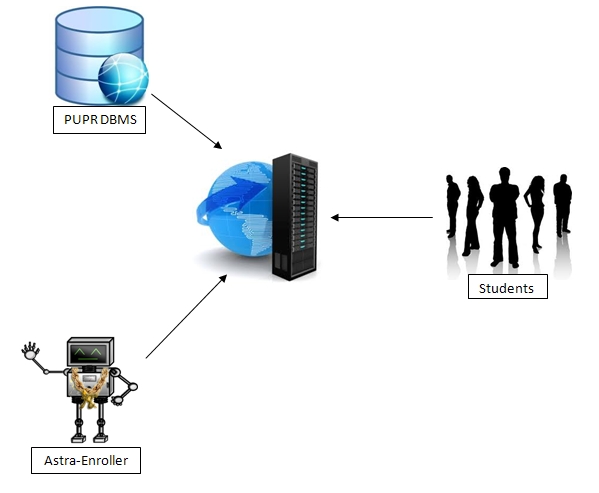


Figure 2.1.5.0 Communication Interface

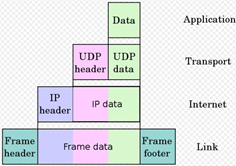


Figure 2.1.5.1 Different layers of TCP/IP

***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

******

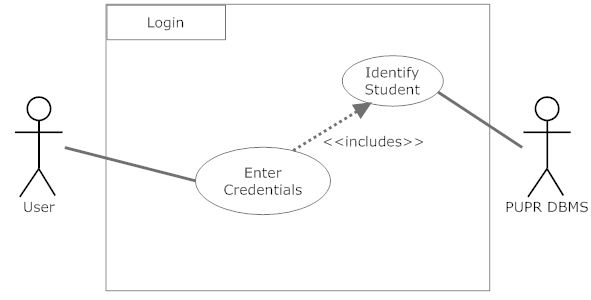
Diagram 2.2.0

***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 their credentials. Once the user has entered correct credentials 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*
    - Minimum Processor speed: 233 MHz
    - Minimum Memory: 64 MB RAM
    - *OS: 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.

**3.0. Specific Requirements**

***3.1 External Interfaces***

Our system, Astra-Enroller, will be able to interact with the user and the PUPR DBMS via inputs provided by the user . They will be organized in the following format as input/output, purpose of input/output and source of input/outputs.

* Astra Enroller Inputs and Outputs

3.1.1

1) Type-Input

1. Name:User Log in credentials (user-name and password)
2. Purpose: The purpose of this input is to validate an user’s credentials so that he may use the system to enroll courses, view transcripts, etc.
3. Source: User
4. Accuracy and Tolerance: The credentials, being a string type the input needs to be exact and will receive no tolerance as user-name and passwords will be considered case sensitive in order to make sure the user logging to our system is a valid one.
5. Unit of Measure: none
6. Timing: none
7. Relationship to other inputs or output: these inputs will be used throughout the whole software processes since all operations need these credentials so that the system performs its tasks accordingly and these tasks are to be applied on the correct user.
8. Screen format/organization: Log in text box user-name[\_\_\_\_\_\_\_\_] password [\_\_\_\_\_\_\_\_\_\_] upper part of web page.
9. Windows format/organization: On the window you will commonly see a tittle below that you will see a presentation of the page and the log in text box.
10. Data Formats: char, string
11. Commands Formats: none
12. End Messages: Log in successful ! or error message if log in is unsuccessful

3.1.2

2) Type- Input

1. Name: Course Selection
2. Purpose: To tell the system which course the user desires to enroll into
3. Source: User
4. Accuracy and Tolerance: This will be invoked by clicking a link with the course name and code so as long as the mouse is hovered above the link and then performing a left click will execute the input and subsequent script functions to do after the input is received by the system.
5. Unit of Measure: none
6. Timing: none
7. Relationship to other inputs or output: this input will produce an output so that the student may select the course section .
8. Screen format/organization: Course name and code displayed accordingly to the student’s curriculum
9. Windows format/organization: Multiple links sorted by an ascending alphanumerical order.
10. Data Formats: hypertext link
11. Commands Formats: none
12. End Messages: none

3.1.3

3) Type- Output

1. Name: Course section viewer.
2. Purpose: To show the target students the sections available to enroll in the course they selected and for them to see which sections have conflicts with already enrolled courses.
3. Source: Astra Enroller
4. Accuracy and Tolerance: doesn't apply here
5. Unit of Measure: none
6. Timing: none
7. Relationship to other inputs or output: invoked by Astra Enroller recieving a course selection input. and after this output the system invokes
8. Screen format/organization: lists in hyperlink the course section ,time,available space and professor and will mark if an specific section has a conflict.
9. Windows format/organization: small window box
10. Data Formats: hypertext link
11. Commands Formats: none
12. End Messages: Enrollment successful!

3.1.4

4) Type-Input

1. Name: Course Section selection
2. Purpose: To select the course section the student wishes to enroll.
3. Source: Student
4. Accuracy and Tolerance: The system will show which course sections have conflict with other courses upon selection of those sections, the system will prompt the student if he wishes to drop the conflicting course for his new selection, otherwise proceeds to enroll the selected course input.
5. Unit of Measure: none
6. Timing: none
7. Relationship to other inputs or output:
8. Screen format/organization:
9. Windows format/organization:
10. Data Formats: hypertext link
11. Commands Formats: none
12. End Messages: Enrollment successful!

3.1.5

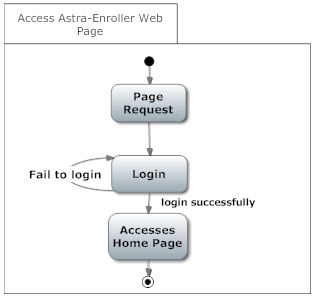
5) Type-Input

1. Name: Query Results
2. Purpose: This input’s purpose is to tell the system the students credentials where correct, what output to produce according to the student’s credentials and service requested by him and also to validate information given to the PUPR DBMS is correct and be able to proceed on functions.
3. Source: PUPR DBMS
4. Accuracy and Tolerance: will depend on the request made by Astra Enroller but its expected that the tolerance is none either its the query result that was expected or an incorrect one
5. Unit of Measure: none
6. Timing: none
7. Relationship to other inputs or output: Will affect the system’s outputs and function decision making
8. Screen format/organization: does not apply.
9. Windows format/organization: does not apply
10. Data Formats: varies from Boolean, char, date and integer data types.
11. Commands Formats: none
12. End Messages: none

***3.2. Functional Requirements***

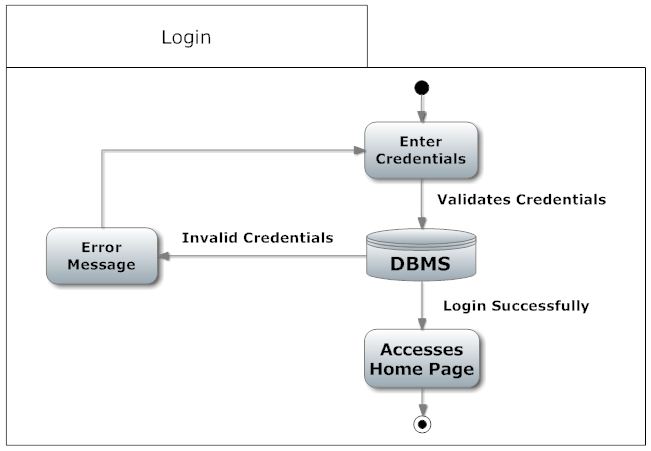
***3.2.1. Access Astra Enroller Web Page***

|  |  |
| --- | --- |
| **Use Case Name:** | Access Astra-Enroller Web Page |
| **Actors** | Primary: Student  Secondary: PUPR DBMS |
| **Precondition** | Student is connected to the Internet |
| **Basic Path** | 1. The Student shall request access to the web page  2. Astra-Enroller Server shall present the Student login Page |
| **Alternate Path** | N/A |
| **Post-condition** | The Student is in the login Page |
| **Exception Path** | 2. There is a connection failure therefore Astra-Enroller Server returns to the wait state |
| **Other** | N/A |



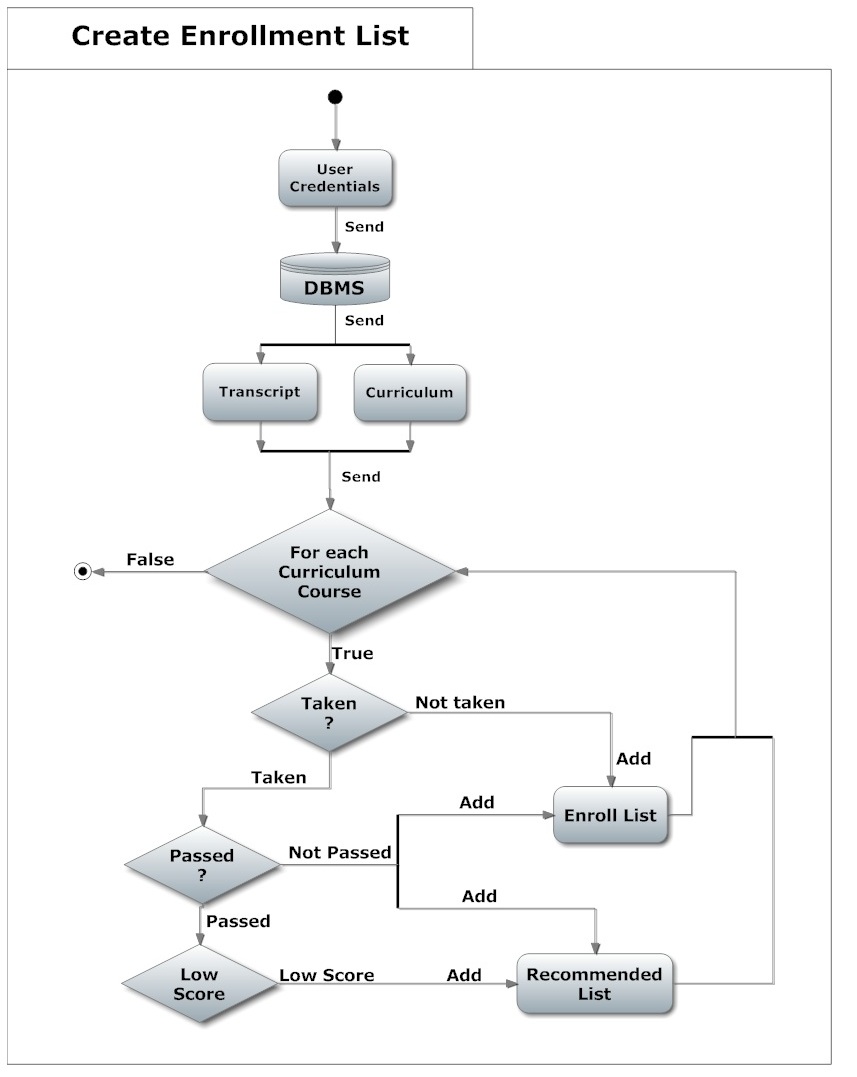
***3.2.2. Login***

|  |  |
| --- | --- |
| **Use Case Name:** | Login |
| **Actors** | Primary: Student  Secondary: Data Base Management System |
| **Pre-condition** | Student is connected to the Internet and have accessed Astra-Enroller Web Page |
| **Basic Path** | 1. The Login Page shall contain a field for a user name, a field for a password as a secret field (not displayed) and a button labeled login.  2. The Student shall fill the username and password field and press the login bottom.  3. The AES shall send a request to the DB to validate the login credentials.  4. The AES shall present the Home Page. |
| **Alternate Path** | N/A |
| **Post-condition** | The Student is in the Home Page |
| **Exception Path** | 4. There is a connection failure therefore Astra-Enroller Server returns to the Login page. |
| **Other** | N/A |



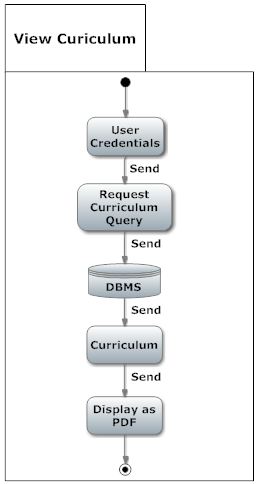
***3.2.3. Create Enroll list***

|  |  |
| --- | --- |
| **Use Case Name:** | Create Enrollment list |
| **Actors** | Primary: Student  Secondary: Data Base Management System |
| **Pre-condition** | Student is connected to the Internet and on the Home Page |
| **Basic Path** | 1. The AES shall query the Student Database using its credentials for the particular trimester and it returns to AE a list of courses from the Student curriculum and transcript.  2. Astra Enroller shall use this data to determine both the enrollment list and the recommended list by validating the each course taken and not taken by the student.  3. The Enroll list shall be filled with the validated course not taken. |
| **Alternate Path** | 3a. If the course being validated was already taken, AE shall check if the course was passed or drop.  1. If the course being validated was drop or not passed it shall be added to the enroll list and the recommended list.  2. If the course being validated was passed with low grades, it shall be added to the Enroll list and the recommended list. |
| **Post-condition** | The Student is in the Enrollment List Page |
| **Exception Path** | 1. When there is a connection failure therefore Astra-Enroller shall return to the Home state if possible. The browser will handle the rest  2. In Step 2 AES can’t determine the courses existence in the DBMS, or the grades of the courses from the transcript file, AE should display an error windows revealing that the data was not able to be retrieved. |
| **Other** | N/A |



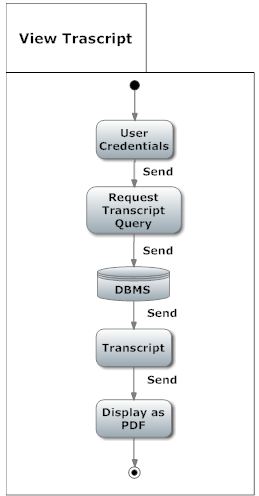
***3.2.4. View Curriculum***

|  |  |
| --- | --- |
| **Use Case Name:** | View Curriculum |
| **Actors** | Primary: Student  Secondary: N/A |
| **Pre-condition** | Student is connected to the Internet and on the Home Page |
| **Basic Path** | 1. The AES shall querie the Student’s Database using its credentials for the particular trimester and return the curriculum of the Student.  2. The AES shall return the curriculum in PDF format. |
| **Alternate Path** | N/A |
| **Post-condition** | The Student is in the Home Page |
| **Exception Path** | 1. There is a connection failure therefore Astra-Enroller shall not return the file and instead shall return to the Home Page and the browser will handle the rest. |
| **Other** | N/A |



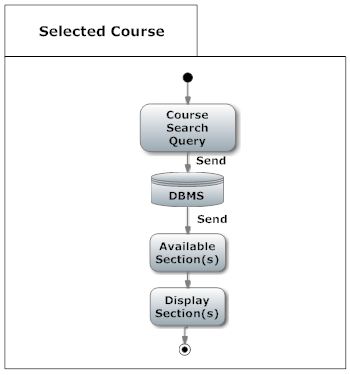
***3.2.5 View Transcript***

|  |  |
| --- | --- |
| **Use Case Name:** | View Transcript |
| **Actors** | Primary: Student  Secondary: N/A |
| **Precondition** | Student is connected to the Internet and on the Home Page |
| **Basic Path** | 1. The AES shall query the Student’s Database using its credentials for the particular trimester and return the transcript of the Student.  2. The AES shall return the Student’s transcript in PDF format. |
| **Alternate Path** | N/A |
| **Post-condition** | The Student is in the Home Page |
| **Exception Path** | 1. There is a connection failure therefore Astra-Enroller shall not return the file and instead shall return to the Home Page and the browser will handle the rest. |
| **Other** | N/A |



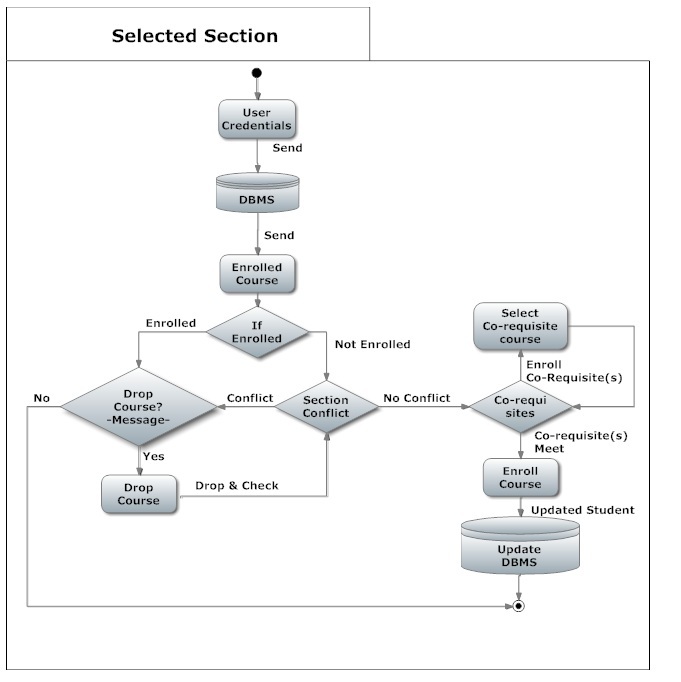
***3.2.6 Selected Course***

| **Use Case Name:** | Selected Course |
| --- | --- |
| **Actors** | Primary: Student  Secondary: Data Base Management System |
| **Precondition** | The Student is in the Enrollment List Page and selected the course to be enrolled. |
| **Basic Path** | 1. The student shall select a course of it’s preference by clicking one of the presented courses of the enrollment list generated.  2. With the course selected by the student, AES shall request the DBMS the list of the sections available referred to the selected course.  3. AE shall display the list of sections to the student. |
| **Alternate Path** | N/A |
| **Post-condition** | The Student is in the Home Page |
| **Exception Path** | 1. There is a connection failure therefore Astra-Enroller shall not return the Section list and instead shall return to the Enroll list Page and the browser will handle the rest. |
| **Other** | N/A |



***3.2.7 Selected Section***

| **Use Case Name:** | Selected Section |
| --- | --- |
| **Actors** | Primary: Student  Secondary: Data Base Management System |
| **Precondition** | The Student is in the Section Selection list Page and selected the section of its preference. |
| **Basic Path** | 1. AE shall enroll the selected course, using the selected section by the student.  2. The AE system shall compare if the course is already enrolled or not.  3. If the course is not enrolled then AE will check if there is a Section conflict.  4. If there is no conflict then AE shall check if the selected course has any co-requisites.  5. If the selected course doesn't have any co-requisites or the co-requisites are met then the course shall get enrolled for that student and both the copy of the transcript and the original transcript from the DBMS shall be updated. |
| **Alternate Path** | 3a. If the course is enrolled then AE shall ask, throughout a windows, if the student wants to drop the course.  4a. If there is a conflict then AE shall ask, throughout a windows, if the student wants to cancel and drop the course.  5.a  1. If there is a co-requisite and the student have not enrolled this course then AE shall ask throughout a windows if the student wants to enroll the co-requisite.  2. If the student enrolled the co-requisite, then AE shall confirm that the course does not have any other co-requisite and validate it. |
| **Post-condition** | The Student is in the Home Page |
| **Exception Path** | 2. There is a connection failure therefore Astra-Enroller Server shall not return the file and instead shall return to the Home Page |
| **Other** | N/A |



***3.3 Performance Requirements***

***Response time***

The minimum time the system will take to display the course that user can take for the next term is 2 seconds. Selecting a course for a term will have minimum time of 1 second. The system will have a minimum time of 3 seconds updating the database.

Note -- all the times mentioned were set using Internet connection of 1 Kpbs, a processor of 1GHz and Google Chrome. If the user has a different Internet connection, processor and browser the times mentioned above will vary.

***Capacity***

The maximum number of courses a user can enroll in a term is 7 for an under-graduate user. The maximum number of users the system will handle at any given time should not be higher than 10000 to prevent any server crashes. At this point the web-page will show a message exhorting the user to wait while the server refreshes.

***Other Requirements***

The user needs to have the username and password that the institution gave to them if they want to use our system. Selecting course and updating will only be available at registration time. The student can enter at any time on the web-page to see the courses he can take for the next term but it cannot select any course.

***3.4 Logical Database Requirements***

3.4.1 Student Entity

|  |  |  |
| --- | --- | --- |
| **Attribute** | **Type** | **Description** |
| Username | Varchar | Use for the log in function. |
| Password | Varchar | Use for the log in function. |
| Transcripts | Varchar | Courses taken by the student. |
| Curriculum | Varchar | All the courses, major and pre-requisites. |

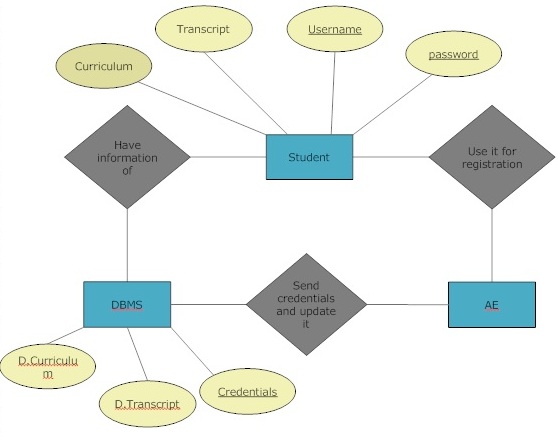
3.4.2 DBMS Entity

|  |  |  |
| --- | --- | --- |
| **Attribute** | **Type** | **Description** |
|  |  |  |
| Transcripts | Varchar | The DBMS send it to AE to validate the course taken. |
| Curriculum | Varchar | The DBMS send it to AE to validate all the courses, major and pre-requisites. |
| Credentials | Varchar | The DBMS validates if the username and password exists. |

3.4.3 Relationship table

|  |  |  |  |
| --- | --- | --- | --- |
|  | DBMS | AE | Student |
| DBMS |  | Sends transcript and curriculum to. | Have information of. |
| AE | Send credentials of student to validate. Upgrade it. |  |  |
| Student |  | Use it to register. |  |

**AE it’s an entity but does not have any attributes because it is not a database, it serves as a view of the DBMS,**



**Figure 3.4 – ERD between DBMS, AE and Student**

***3.5 Design Constraints***

***Hardware restrictions:***

***User system requirements:***

· Processor : 233 MHz

· Memory : 64 MB RAM

Server requirements:

either Windows Server or Linux Server

processor : 8 core processor (3.6 GHz)

memory minimum: 30 GB of RAM

***Software Restrictions:***

· *OS: None*

· *Web browser: Mozilla Firefox or Google Chrome*

***Programming languages:***

· PHP (ver 4.3 or higher)

· MySQL

***3.6 Software System Attributes***

* ***Reliability***

Factors required to ensure that Astra-Enroller functions properly are Internet connection with at least a speed of 1 kbps can be DSL or Broadband or even Dial-up connection to make sure communication of Astra-Enroller with it’s user and other systems such as the DBMS servers, complies with the established response time. Astra-enroller is capable of handling an amount of ten thousand (10,000) users logged in while performing tasks and transactions at the same time, without having problems such as task incompletion or slowdowns (reduction in the speed at which Astra-Enroller communicates with the DBMS and it’s users) . The Servers with the DBMS whom Astra- Enroller will communicate to perform it’s determined tasks, must be operational otherwise Astra-Enroller would be unable to complete it’s given tasks.

* ***Availability***

Astra-Enroller will only be available for its main use at a determinate date established by the administration of the PUPR for enrollment during trimesters. Other cases where the system Availability may be compromised is when both Astra-Enroller’s or the DBMS Servers are under maintenance (common server maintenance is around 2 to 4 hours of duration) or shut down due to black outs. As for the User’s end, as long as they have an Internet connection they can access and use Astra-Enroller.

* ***Security***

Astra-Enroller requires and will implement “https” for user authentication purposes. Using the authentication certificate and user credentials implemented on the PUPR database, ensuring that the target users belong to the PUPR system. While at the same time protecting the system’s communication of sensitive information belonging to the user with the PUPR Database using the encrypted SSL and TSL connection features. Thus preventing eavesdropping attacks on our system that would compromise our user’s credentials and sensitive information. Other security measures are authentication certificates will expire if the user is inactive or doesn’t perform an action in X minutes forcing validation of user credentials in order to continue using our system.

* ***Maintainability***

Maintenance to the Astra-Enroller’s server can be done at any needed time, and developers and administrators may access it remotely since it’s a web based application. Internet is required to perform maintenance to the system and this method allows for system optimization ( increasing performance speed for example ) and updates such as implementation of new features that are between the scope of goals of our system. Also since the system’s main functions are kept with as much independent functionality from one another, this allows for more precise maintenance to the system without compromising its internal structure.

* ***Portability***

Astra-Enroller is a web based application using https with, php which will allow accessibility to the users regardless of web browser and most general OS platform they are using, so as long as the users have an Internet connection, then they can access and use Astra-Enroller from anywhere from any equipment such as computers, smart-phones and tablets that have a web browser .