**Table of contents**

1.0 Introduction......................................................................................................... 2

1.1 Purpose................................................................................................................ 2

1.2 Scope................................................................................................................... 2

1.3 Definitions, acronyms and abbreviations............................................................. 4

1.3.1 Definitions........................................................................................................ 4

1.3.2 Acronyms.......................................................................................................... 5

1.3.3 Abbreviations.................................................................................................... 5

1.4 References........................................................................................................... 5

1.5 Overview............................................................................................................. 6

2.0 General Description............................................................................................. 7

2.1 Product Perspective............................................................................................. 7

2.2 Product Functions................................................................................................ 13

2.3 User Classes Characteristics................................................................................ 17

2.4 Constraints........................................................................................................... 17

2.5 Assumptions and dependencies........................................................................... 18

3.0 Specific Requirements......................................................................................... 19

3.1 External Interface................................................................................................ 19

3.2 Product Functions................................................................................................ 24

3.3 Performance Requirements.................................................................................. 33

3.4 Logic Requirements of Database......................................................................... 34

3.5 Design restrictions............................................................................................... 36

3.6 Software System Requirements........................................................................... 37

# 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. Also it includes the features and its explanation of what it is going to do and not, like for example our system will be able to make the process of enrollment totally online, and won’t be able to be access for any students if they don’t have Internet. The SRS is directed to clients, this is the people who pay for the product, users this are the students who will benefit from our product, and the developers of the system, people in charge of maintenance and making updates to the system.

## 1.2. Scope of Project

Astra Enroller will be name of our enrollment system. 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. |
| 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. |

## 1.3.2 Acronyms

|  |  |
| --- | --- |
| Term | Acronym |
| Polytechnic University of Puerto Rico | PUPR |
| Software Requirements Specification | SRS |
| Database Management System, also will be referred  to the PUPR's database. | 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 |

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.

***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 the client. 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***

Designing an online registration system is very beneficial because it saves the time and money. It is also in great demand as most college students have Internet service at home and prefer to use an online registration system instead of going in person to the university. But there are many problems with the current system of registration of the Polytechnic University of Puerto Rico. Among the most common are: problems of validation of courses and the current system contains a search function which is not sufficiently clear. It is for this reason that Astrapi Enterprise set out to create a system called Astra-Enroller, to improve the system that is currently in use (MyPoly) that allows online enrollment.

Many students have been in the obligation of having to go to the institution to complete the enrollment process. Because the current system do not validated courses already taken, pre-requisites for other courses, or they simply do not understand the system and did not know how to use it. That is why through our system, Astra-Enroller, we will offer the student, new and better features than the previous system. Our system will validate the courses depending on the student’s curriculum, this will make the registration process faster and more organized because the student will see exactly which courses can take for the next term. These new features will allow the user to the online registration process in a much more easily, quickly and efficiently.

To use our system the user must login to our web-page using the username and password given by the institution, our system in turn establishes a connection to the database of the Polytechnic University of Puerto Rico, which keeps and manages all the students’ information. The database of the university will send to our system all the information necessary for the registration process (transcript and curriculum).

*2.1.1 System interface*

Astra Enroller is a component of a larger system that consists of five principal and components.

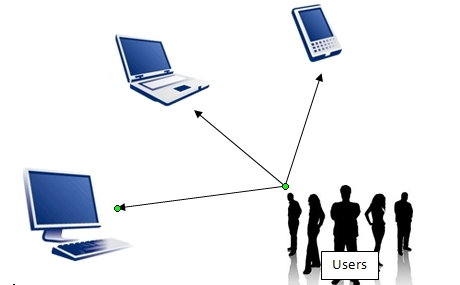
These components are *The Internet, The administrators, The Users, Astra Enroller and The PUPR DBMS.*  All of these components communicate via the Internet making it the principal component that is needed for the whole system to work. Administrators have the privilege to change and modify Astra Enroller mainly for maintenance or software update purposes. Users via Internet may access Astra Enroller and complete their course enrollments by using the many robust functions available which end in Astra Enroller establishing communication with the PUPR DBMS and officially enrolling courses.

*2.1.2 User Interface*

Every user has to use a web browser in order to communicate with Astra Enroller. After the user logs in their credentials and these are validated by the software, Astra Enroller will proceed to show the user their options such as to View their credit transcript, Show the courses the user can enroll in the corresponding trimester, where the user can click on the desired course in order to begin the enrollment process. Afterwards the system shows the user his enrollment status and when the user concludes his enrollment process they can simply leave by closing the web browser or logging out.

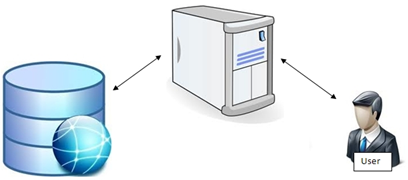
*2.1.3 Hardware interface*

In order to for users to access the software, the equipment needed is a computer with basic peripherals like a monitor, mouse , laptops, tablets and even smart phones that have web browsers with them. Also these hardware equipments should have peripherals that allow them to connect through the internet and a display screen to enable the desired software interface communication with Astra Enroller. Also these computers ,tablets, smart phones should have as a minimum requirement a processor with a clocking of 233 Mhz for desired performance.



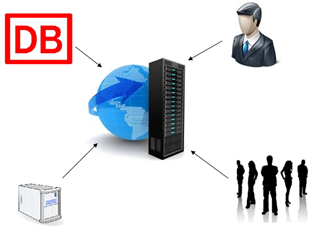
*2.1.4 Software interfaces*

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 a OS that can support Mozilla Firefox and/or Google Chrome web browsers.

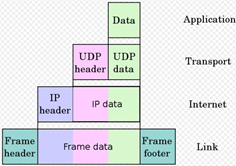


*2.1.5 Communications Interfaces*

We’ll have a server which will contain everything related to Astra-Enroller and their website. All the different types of users will communicate with the server. All equipment chosen by the user to access our server should be able to access Internet.



For managing communications we’ll use TCP/IP.



*Different layers of TCP/IP*

*2.1.6 Memory Constraints*

The recommended memory use is 64 MB of RAM. With that RAM all functions and features of our system will work at the response time desired. The speed or response of the system will vary depending if the user have a higher or lower ram.

*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.

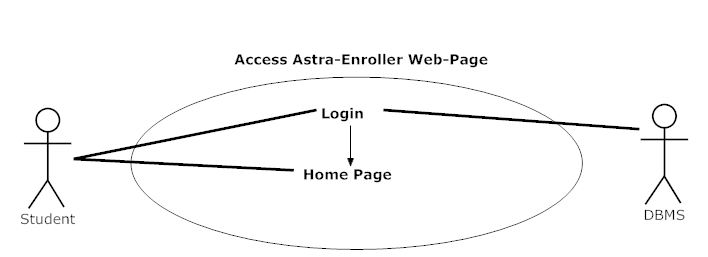
1. User logs in (enter credentials) on the system
2. Validate the user and give permissions
3. System validates if the user can enroll in that moment.
4. System validates what courses the student can enroll in, this includes if there are any requirements.
5. The system will display the name and course 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



***Description:*** The student access the Astra-Enroller Web-Page using a required browser

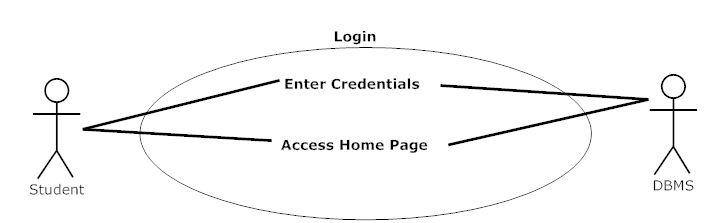
and logs in the Home Page by using its personal credentials.

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

1. The student connects to the Astra-Enroller Web Page using a browser and Internet.

2. The Astra-Enroller server validates the browser request and returns to the student the AE Login page

* + Login



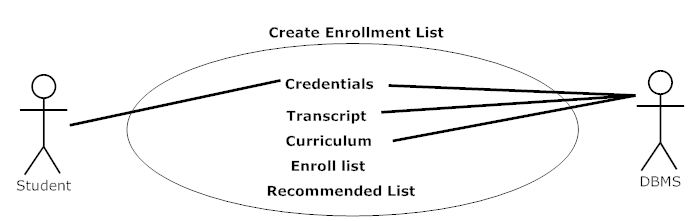
***Description***: The student logs in the Astra-Enroller Web-Page using a required personal credentials and is then being presented with the AE Home Page

***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 validates the credentials and returns to the student the AE Home Page

* + Create Enrollment List



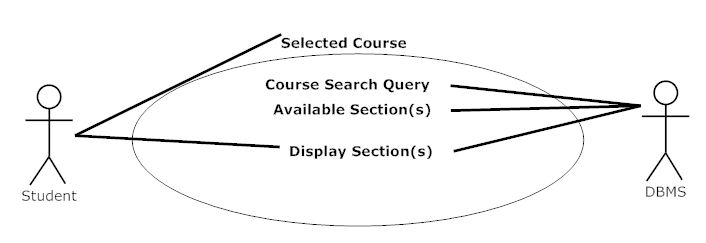
***Description***: After logging in, AE displays the Home Page presenting the student with many options in a menu but in the process by logging in, AE’s algorithm determine the courses that are available for that student and generates two lists, the enroll list and the Recommended List. The system does this by using the transcript and the curriculum of the student from the DBMS of the university.

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

1. The student logs in AE and the Home Page is successfully loaded with no errors.

2. AE algorithm will fill the enroll list.

* + Select Course



***Description***: After successfully loading the web page the user selects the course that he wants to enroll from the enroll list. This function will retrieve data about the sections of the course and other information from the DBMS and displays the user with the Section Selection Page.

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

1. User clicks on a course from the list.

2. AE retrieve information from the DBMS and displays the course Section.

* + Select Section
  + View Curriculum
  + View 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 great. 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 MySQL*

It’s necessary to take this assumption since AE will attempt to communicate with the Campus database interacting with instructions based on MySQL 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

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

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 recieved 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) Type- Output

1. Name: Course section viewer.
2. Purpose: To show the target students the sections avialable 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: doesnt 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!

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!

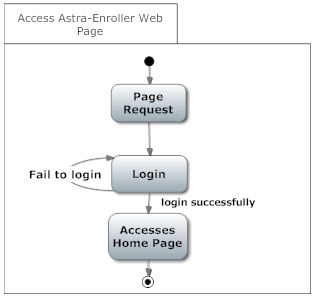
5) Type-Input

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***

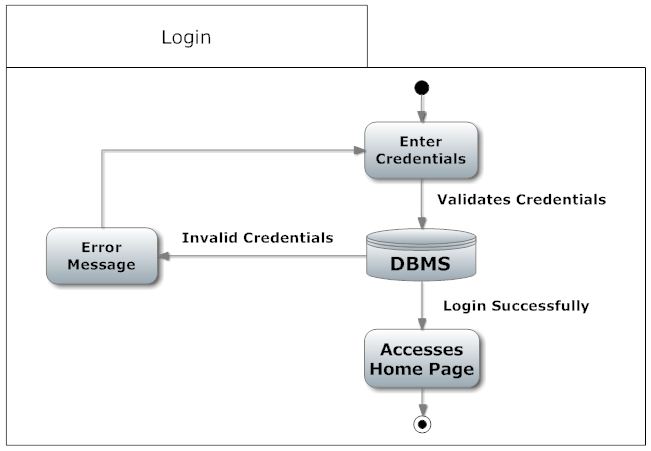
***3.2.1. Access Astra Enroller Web Page***

|  |  |
| --- | --- |
| **Use Case Name:** | Access Astra-Enroller Web Page |
| **Actors** | Primary: Student  Secondary: N/A |
| **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** | If there is a connection failure the Astra-Enroller Server returns to the wait state |
| **Other** | N/A |



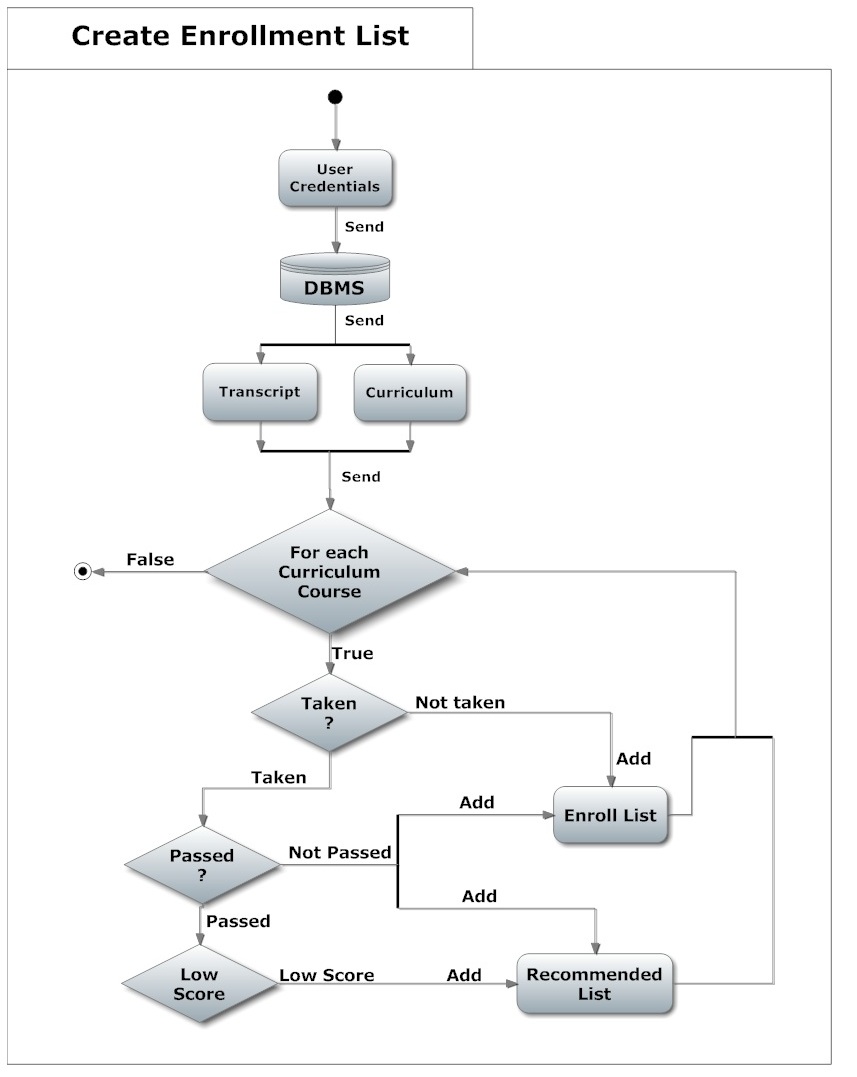
***3.2.2. Login***

|  |  |
| --- | --- |
| **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** | If there is a connection failure the Astra-Enroller Server returns to the Login page. |
| **Other** | N/A |



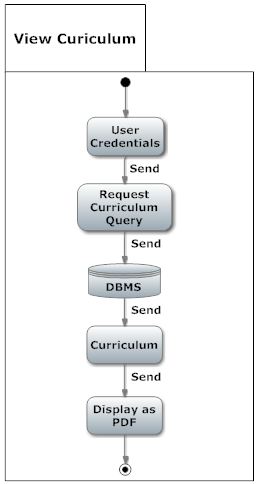
***3.2.3. Create Enroll list***

|  |  |
| --- | --- |
| **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. If there is a connection failure the Astra-Enroller shall return to the Home state if possible. The browser will handle the rest  2. In Step 2 If 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 |



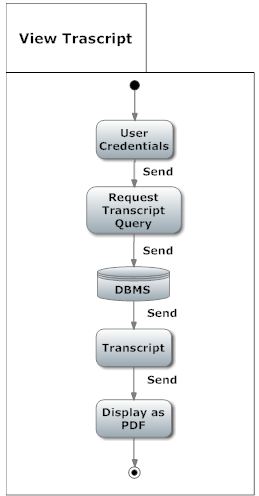
***3.2.4. View Curriculum***

|  |  |
| --- | --- |
| **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. If there is a connection failure the 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 |



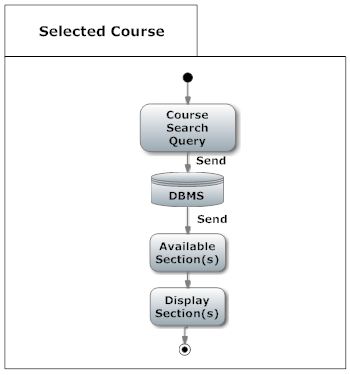
***3.2.5 View Transcript***

|  |  |
| --- | --- |
| **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. If there is a connection failure the 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 |



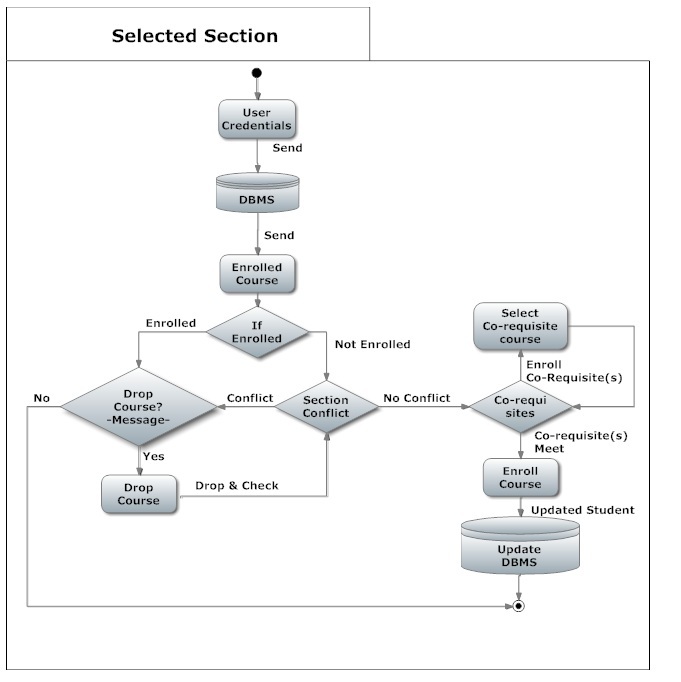
***3.2.6 Selected Course***

|  |  |
| --- | --- |
| **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. If there is a connection failure the 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** | If there is a connection failure the 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. |

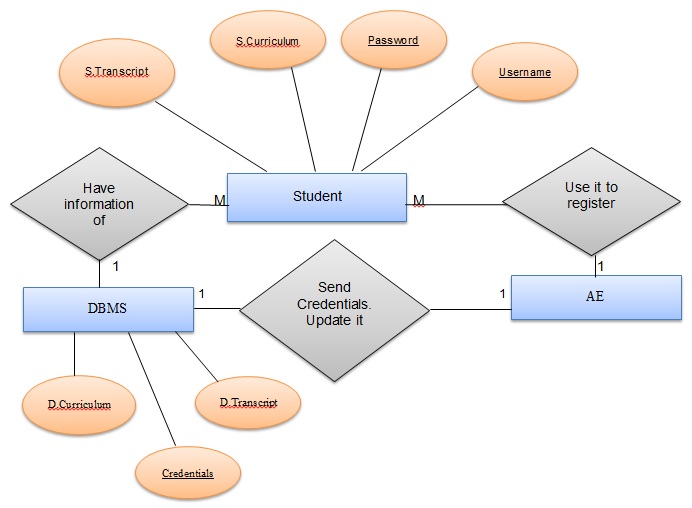
3.4.2 DBMS Entity

|  |  |  |
| --- | --- | --- |
| **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 Minimum: 233 MHz

· Memory Minimum: 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 Minimum: 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 .