**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

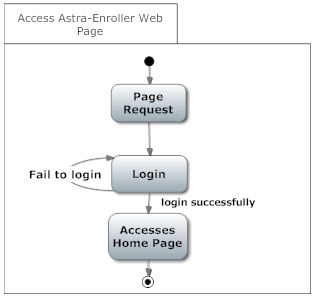
5) Type-Input

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***

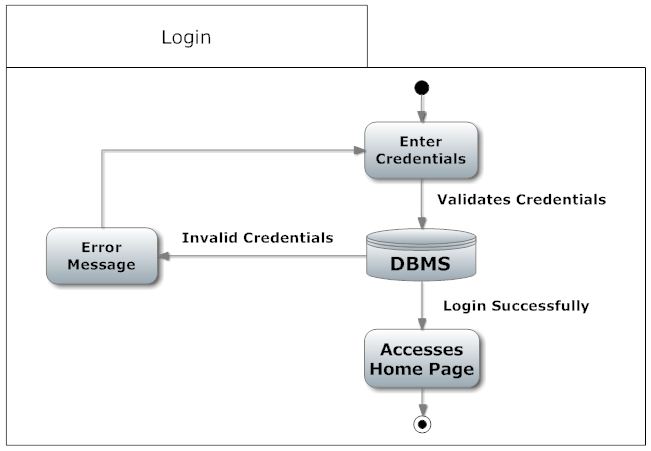
***3.2.1. Access Astra Enroller Web Page***

|  |  |
| --- | --- |
| **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** | If there is a connection failure the Astra-Enroller Server returns to the wait state |
| **Other** | N/A |



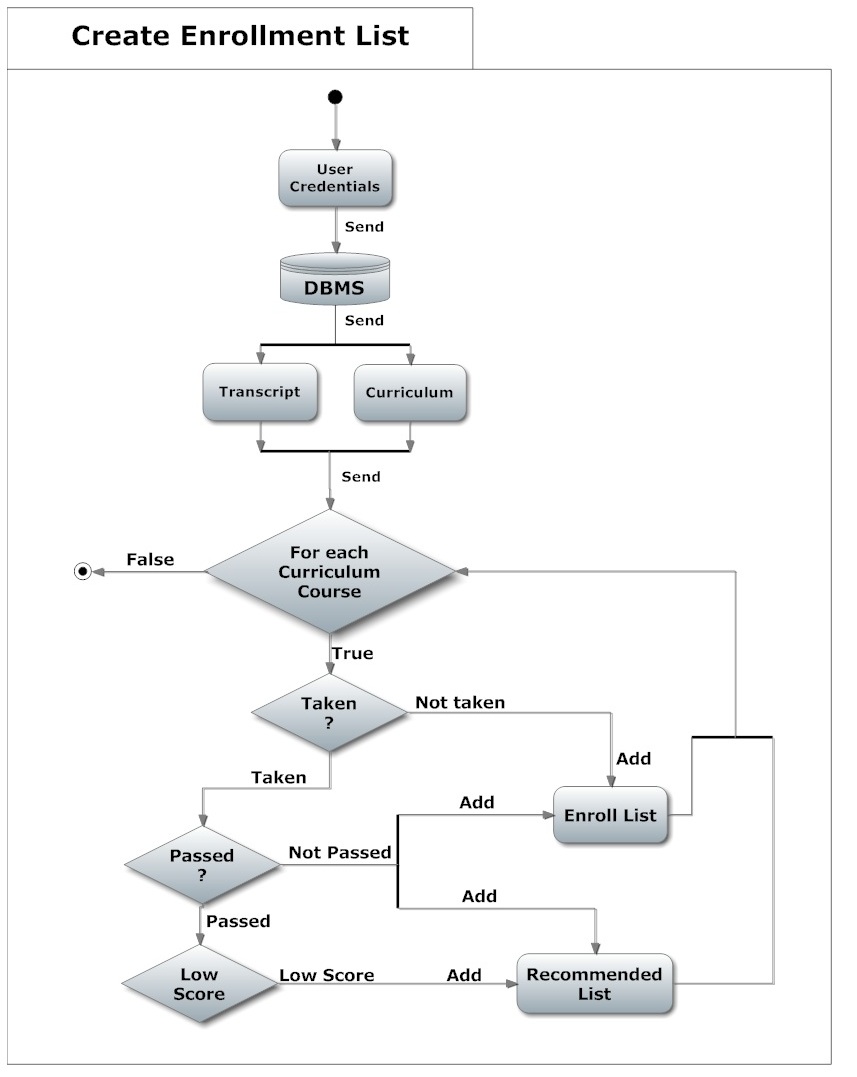
***3.2.2. Login***

|  |  |
| --- | --- |
| **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 |



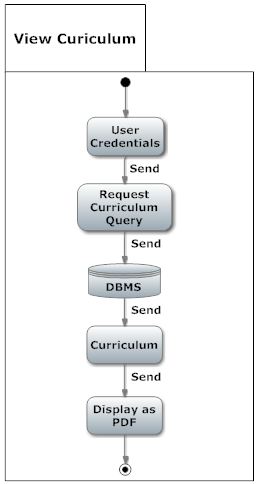
***3.2.3. Create Enroll list***

|  |  |
| --- | --- |
| **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 |



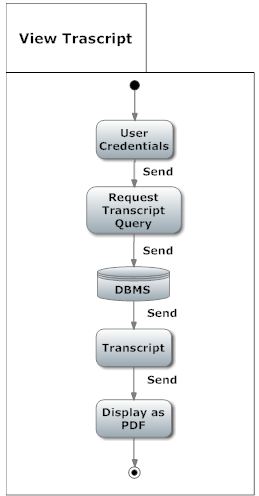
***3.2.4. View Curriculum***

|  |  |
| --- | --- |
| **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 |



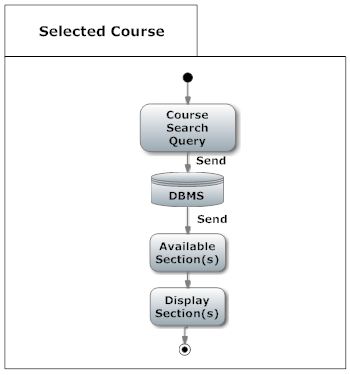
***3.2.5 View Transcript***

|  |  |
| --- | --- |
| **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 |



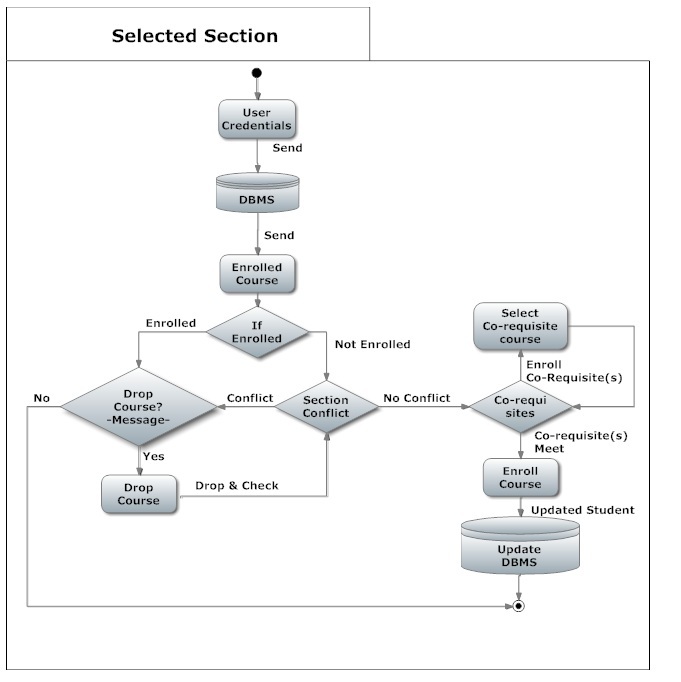
***3.2.6 Selected Course***

|  |  |
| --- | --- |
| **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. |

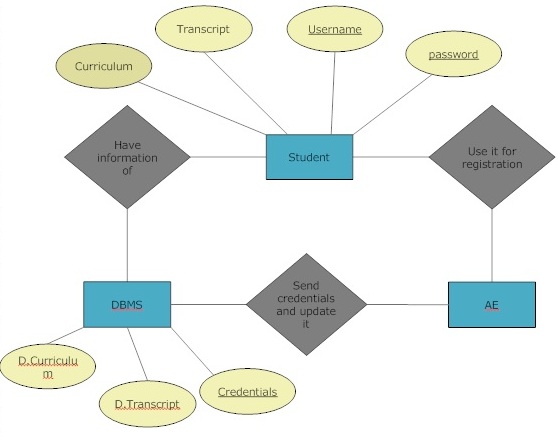
3.4.2 DBMS Entity

|  |  |  |
| --- | --- | --- |
| **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 .