**Astra-Enroller**

**Software Design Document**

<Ver. 2.0>

*Developers:*

Gabriel E. Nieves Rodríguez #52475 \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

Emanuel Rivera Castro #53502 \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

Joaquin Pockels Balaguer #54012 \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

Francisco O. Ramos Bravo #65601 \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

Luis Ayala Silva #54034 \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

Course: Software Engineering

Due Date: May 26, 2011

Prof. Luis Ortiz

**Table of Contents**

1. **Introduction**

**Revision history**

|  |  |  |  |
| --- | --- | --- | --- |
| Date | Version | Description | Author |
| 04/19/2011 | **1.0** | **Section 1** | **Emanuel Rivera** |
|  | **1.0** | **Design Draft** | **Joaquin Pockels** |
|  | **1.0** | **Section 2** | **Joaquin Pockels** |
|  | **1.0** | **Section 3** | **Joaquin Pockels** |
|  | **1.0** | **Section 4** | **Joaquin Pockels** |
|  | **1.0** | **Section 5-6** | **Joaquin Pockels** |
|  | **1.5** | **Updated Section 2-3** | **Joaquin Pockels** |
|  | **1.5** | **Updated Section 4** | **Joaquin Pockels** |
|  | **2.0** | **Updated Section 5-6** | **Joaquin Pockels** |
| 05/26/2011 | **2.0** | **Final Release** | **Joaquin Pockels** |

**1. Introduction**

**1.1 Purpose**

The purpose of the Software Design Description (SDD) is to represent or model the Astra-Enroller system (AE) with a precise information needed for planning, analysis, and implementation. This includes the architectural features of the system down through details of what operations each code module will perform and the database layout. It also shows how the use cases promised in the SRS will be implemented in the system using this design.The SDD partitioned the system into design entities and describe the important properties and relationships among the entities.

The design description model used to represent the software system will be expressed as a collection of designs entities, each possessing properties and relationships. To simplify the model, the properties and relationships of each design entity are described by the standard set of attributes. A design description is complete when the attributes have been specified for all the entities.

**1.2 Scope**

This Software Design Document (SDD) provides a complete description of how the requirements of the Astra-Enroller (AE), mentioned in the Software Requirement Specification (SRS) document will be implemented . The design methodology consists in a general-purpose scripting language, in our case a Hypertext Pre-processor (PHP) language and also a relational database management system (RDBMS) MySQL design using a visual interface to a database management system. AE will create a query to the university database for the user’s transcript and curriculum. With this information AE will be able to display the courses the user can take. When the user selects a course AE will create a query for all the sections available for that course. If the user selects a section that specific time and date will become unavailable unless the user drop it. AE will update the transcript on the database.

**1.3 Definitions and acronyms**

Definitions

|  |  |
| --- | --- |
| Term | Definition |
| Database | Collection of all the information monitored by this system. |
| Student | A student of the Polytechnic University of Puerto Rico. |
| User | A student of the Polytechnic University of Puerto Rico. |
| Query | A form of questioning, in this case language a DBMS can understand of which we can ask for certain data and it will |
| Software Design Description | Is a representation or model of the software system to be created. Provide the precise design information needed for planning, analysis and implementation, |
|  |  |

Acronyms

|  |  |
| --- | --- |
| Term | Acronym |
| Polytechnic University of Puerto Rico. | PUPR |
| Software Design Description | SDD |
| Software Requirement Specification | SRS |
| Database Management System. | DBMS |
| Astra-Enroller. | AE |
| Entity-Relationship Diagram | ERD |
| Operating System. | OS |
| PHP | Hypertext Preprocessor |
| relational database management system | RDBMS |
| “And so on” or “et cetera” | etc. |

**1.3. References**

1. IEEE, IEEE Std. 830-1998 IEEE Recommended Practice for Software

Requirements

1. UML 2.0 in a Nutshell (2005)By Dan Pilone, Neil Pitman

**2. System Overview**

**2.1 Introduction**

Astra-Enroller is a system which consists of a web server and various web pages, that together with the systems of the PUPR Database and the students fulfill the requirements established at the SRS. The role of the Astra-Enroller is to let the students of the Polytechnic University of Puerto Rico enroll their classes throughout an easy and user friendly way.

The web server sub-system of the Astra-Enroller system is configured to establish the web pages and algorithms in it, so that it can be accessed by the student’s computers and it is also configured so that it have access to the PUPR Database to get and/or modify the information needed to comply with the requirements.

**2.2 Deployment Diagram**

Basic hardware dependency or deployment diagram can be found at Figure 1. Detailed deployment diagram or hardware relation can be found at Figure 1.1.



Figure 1. Basic Deployment Diagram.

Figure 2. Detailed Deployment Diagram.

The deployment diagram represents the relation between all the hardware and systematic hardware that are required for the system to work as intended.

**3.0 Architectural Design**

**3.1 Decomposition Block Diagram**



Figure 2. Decomposition Block Diagram

The Decomposition Block Diagram in Figure 2 represents the general architecture of the Astra-Enroller system. Most of the classes or objects found at the Astra-Enroller are interfaces because each one of them are web pages created using HTML and their services using Java script and java.

The Database used is the Polytechnic University of Puerto Rico Database, from which the Astra-enroller system will access the information needed. The Astra-Enroller system will import information to the Data base so that the student can get updated.

**3.2 Main Activity Diagram**



Figure 2.1. Main Activity Diagram

This Diagram represents the activity from begging to end in a generalized matter. From this we develop all other processes.

**3.2.1 Login Sub-Activity**



Figure 2.1.1 Login

Login activity diagram works like most logins found now a day, it validates that the username and password is written within the standard parameters and then it utilizes the PUPR DB to validate the credentials submitted. After this, the student will have access to our system, and will be redirected to the home page.

**3.2.2 View Curriculum sub-activity**



Figure 2.1.2 View Curriculum

The function of view curriculum uploads to the student’s machine the curriculum of that particular student in PDF format.

The curriculum information is obtained through the Home Page which will get from the PUPR DB the student information including its determined curriculum.

**3.2.3 View Transcript sub-activity**



Figure 2.1.3 View Transcript

Similar to the View Curriculum, this function will upload to the student’s computer the transcript file in form of PDF.

**3.3 Enroll Sub-Activity Diagram**



Figure 2.2 Enroll

The Enroll sub-activity diagram represents the process with which our system will enroll the courses for each student.

The

**3.3.1 Display available courses Sub-Activity**

Figure 2.2.1 Display Courses

Available courses sub-activity explanation.

Display available course is the most important algorithm of the system. Basically it enlists each course that the students has not taken, has taken but graded bad or fails (“D” or “F”) and that are available at the current course schedule.

This process, at the end will display all the available courses to the student’s browser, and the student will decide to choose one of them. After this it will select the sections of the course and enroll it. This algorithm was developed so that it would be as easy as possible for the student to decide which one is the best possible option.

**4.0 Detail System Specifications**

**4.1 Class Diagram**

Figure 3 Class Diagram

**4.2 Classes used**

* Login
* HomePage
* Enroll
* ViewTranscript
* ViewCurriculum
* Logout
* EnrolledIn
* Student
* Transcript
* Curriculum
* CourseSchedule