# Document Attributes and Revision History

| Attribute | Value |
| --- | --- |
| Owner | Ankush Arora, Akshay Kumar Dani |
| Owner Contact Information | [aarora3@binghamton.edu](mailto:aarora3@binghamton.edu), adani1@binghamton.edu |

## Revision History

| Author | Date | Version # | Course | Revision Description |
| --- | --- | --- | --- | --- |
| Ankush Arora | 11/22/16 | 1.01 | Database Systems | Initial Version |
| Ankush Arora | 11/29/16 | 1.02 | Database Systems | Made changes to sync with the changes due to challenges faced during implementation. |

Table of Contents

[Document Attributes and Revision History 1](#_Toc261980931)

[Revision History 1](#_Toc261980932)

[Introduction 3](#_Toc261980933)

[Component List 3](#_Toc261980934)

[Detail Design 3](#_Toc261980935)

[Other Information and References 7](#_Toc261980936)

# Introduction

The Design Package describes the components of the solution that will be created or modified and how they will be created or modified in order to meet the requirements.

## Component List

This section lists each component that will be created, modified, or removed.

|  |  |
| --- | --- |
| **Component Type** | **Design Component** |
| Packages | Proj2\_student\_reg\_packageAD1 |
| Sequences | LOGID |
| Triggers | ENROLLMENTS\_INSERT\_LOGS\_T1  ENROLLMENTS\_DELETE\_LOGS\_T1  STUDENTS\_INSERT\_LOGS\_T1  STUDENTS\_DELETE\_LOGS\_T1  STUDENTS\_DELETE\_ENROLLMENTS\_T1  ENROLLMENTS\_INSERT\_CLASSES\_T1  ENROLLMENTS\_DELETE\_CLASSES\_T1 |
| Stored Procedures | show\_students  show\_courses  show\_prerequisites  show\_classes  show\_enrollments  show\_logs  add\_students  show\_students\_classes  show\_prereq  show\_classes\_student\_details  enroll\_student\_in\_class  drop\_student\_from\_class  delete\_student |
| Java Files | Driver  ShowStudents  ShowCourses  ShowPrerequisites  ShowClasses  ShowEnrollments  ShowLogs  AddStudent  ShowStudentClasses  ShowPrereqRecur  ShowClassesStudents  EnrollStudentInClass  DropStudentFromClass  DeleteStudent |

## Detail Design

This section describes the changes for each listed component sufficiently to meet the needs of the people identified to develop, test, deploy, and support the solution.

|  |  |  |  |
| --- | --- | --- | --- |
| **Req. ID** | **Design Component** | **Description** | **Tables used** |
| AD-1 | Proj2\_student\_reg\_packageAD1 | A **package** is a schema object that groups logically related PL/SQL types, items, and subprograms. Packages usually have two parts, a specification and a body. The **specification** (**spec** for short) is the interface to your applications; it declares the types, variables, constants, exceptions, cursors, and subprograms available for use. The **body** fully defines cursors and subprograms, and so implements the spec.  Define package Proj2\_student\_reg\_package and declare and implement all stored procedures inside it. |  |
| AD-2 | LOGID | Sequence to generate the values for logid automatically when new log records are inserted into the logs table. All logid values should have 4 digits.  Use starting value as 1000, max value as 9999 with increments of 1. |  |
| AD-3 | ENROLLMENTS\_INSERT\_LOGS\_T1 | This trigger is fired when a tuple is inserted into the enrollments table to insert a log entry into the logs table.  Use logid value as nextval of logid sequence generated in AD-1.  Use table name as enrollments and operation as insert.  Use key\_value as sid,classid of the inserted tuple in enrollments table. | Enrollments table, logs table |
| AD-4 | ENROLLMENTS\_DELETE\_LOGS\_T1 | This trigger is fired when a tuple is deleted from the enrollments table to insert a log entry into the logs table.  Use logid value as nextval of logid sequence generated in AD-1.  Use table name as enrollments and operation as delete.  Use key\_value as sid,classid of the deleted tuple in enrollments table. | Enrollments table, logs table |
| AD-5 | STUDENTS\_INSERT\_LOGS\_T1 | This trigger is fired when a tuple is inserted into the students table to insert a log entry into the logs table.  Use logid value as nextval of logid sequence generated in AD-1.  Use table name as students and operation as insert.  Use key\_value as sid of the inserted tuple in students table. | Students table, logs table |
| AD-6 | STUDENTS\_DELETE\_LOGS\_T1 | This trigger is fired when a tuple is deleted from the students table to insert a log entry into the logs table.  Use logid value as nextval of logid sequence generated in AD-1.  Use table name as students and operation as delete.  Use key\_value as sid of the deleted tuple in students table. | Students table, logs table |
| AD-7 | STUDENTS\_DELETE\_ENROLLMENTS\_T1 | This trigger is fired when a tuple is deleted from the students table to delete the tuples from enrollments table where the sid is of the student deleted. | Students table, enrollments table |
| AD-8 | ENROLLMENTS\_INSERT\_CLASSES\_T1 | This trigger is fired when a tuple is inseted in the enrollments table to increment the class size of the class which is taken by student. | Enrollments table, classes table |
| AD-9 | ENROLLMENTS\_DELETE\_CLASSES\_T1 | This trigger is fired when a tuple is deleted from the enrollments table to decrement the class size of the class which is dropped by student. | Enrollments table, classes table |
| AD-10 | show\_students | This stored procedure is used to get all student tuples from the student table using cursor. | Students table |
| AD-11 | show\_courses | This stored procedure is used to get all courses tuples from the courses table using cursor. | Courses table |
| AD-12 | show\_prerequisites | This stored procedure is used to get all prerequisites tuples from the prerequisites table using cursor. | Prerequisites table |
| AD-13 | show\_classes | This stored procedure is used to get all classes tuples from the classes table using cursor. | Classes table |
| AD-14 | show\_enrollments | This stored procedure is used to get all enrollments tuples from the enrollments table using cursor. | Enrollments table |
| AD-15 | show\_logs | This stored procedure is used to get all logs tuples from the logs table using cursor. | Logs table |
| AD-16 | add\_students | This stored procedure takes all attributes of student information like sid, firstname, lastname, status, gpa, email as user input and adds a tuple to the students table.  Trigger STUDENTS\_INSERT\_LOGS\_T1 is fired automatically if exists when tuple is successfully added to students table. | Students table, logs table |
| AD-17 | show\_students\_classes | This stored procedure takes students sid as user input and shows student sid, lastname and status along with classid, dept\_code,course\_no, title, year, semester information from the enrollments table for the classes enrolled by this student. | Students table, enrollments table |
| AD-18 | show\_prereq | This stored procedure takes dept\_code and course\_no as input from user and extracts all prerequisite courses from the courses table recursively.  Additional check in beginning to check if the course entered exists, as entered course may not have any prerequisites (thus no entry in prerequisites table) but still a valid course. | Prerequisites table, courses table |
| AD-19 | show\_classes\_student\_details | This stored procedure takes classid as user input and extracts details like classid, title, semester from a join of classes and courses table and also sid and lastname of the students enrolled in the class.  2 separate cursors used getting class details and students enrolled details respectively. | Classes tables, courses table, enrollments table, students table |
| AD-20 | enroll\_student\_in\_class | This stored procedure takes a classid and sid as user input and enrolls student into class if it passes all below constraints in the sequence mentioned:   1. If the student is not in the students table, report “The sid is invalid.” 2. If the class is not in the classes table, report “The classid is invalid.” 3. If the enrollment of the student into a class would cause “class\_size > limit”, reject the enrollment and report “The class is closed.” 4. If the student is already in the class, report “The student is already in the class.” 5. If the student is already enrolled in three other classes in the same semester and the same year, report “Students cannot be enrolled in more than three classes in the same semester.” and reject the enrollment. 6. If the student has not completed the required prerequisite courses with minimum grade “D”, reject the enrollment and report “Prerequisite courses have not been completed.”   For all the other cases, the requested enrollment should be performed.  If after enrollment, the student is enrolled in 3 classes in the same semester and the same year, report “You are overloaded.”  Trigger ENROLLMENTS\_INSERT\_LOGS\_T1 is fired automatically when tuple is successfully added to enrollments table. | **Tables:**  Students, classes, courses, enrollments, prerequisites |
| AD-21 | drop\_student\_from\_class | This stored procedure takes a classid and sid as user input and drops student from class if it passes all below constraints in the sequence mentioned:   1. If the student is not in the students table, report “The sid is invalid.” 2. If the classid is not in the classes table, report “The classid is invalid.” 3. If the student is not enrolled in the class, report “The student is not enrolled in the class.” 4. If dropping the student from the class would cause a violation of the prerequisite requirement for another class, then reject the drop attempt and report “The drop is not permitted because another class uses it as a prerequisite.”   In all the other cases, the student should be dropped from the class.  If the class is the last class for the student, report “This student is not enrolled in any classes.”  If the student is the last student in the class, report “The class now has no students.”  Trigger ENROLLMENTS\_DELETE\_LOGS\_T1 is fired automatically when tuple is successfully deleted from enrollments table. | **Tables:**  Students, classes, enrollments,  prerequisites |
| AD-22 | delete\_student | This stored procedure takes sid as user input and deletes the tuple from students table.  Trigger STUDENTS\_DELETE\_ENROLLMENTS\_T1 is fired automatically when tuple is successfully deleted from students table to delete tuples from enrollments table where the student exists.  Trigger STUDENTS\_DELETE\_LOGS\_T1 is fired automatically if exists when tuple is successfully deleted from students table. | Students table, enrollments table |
|  | Driver | Java driver file from where the main execution starts (main function).  Driver displays the menu and gives user choice to select an action out of below:  1. Show Students  2. Show Classes  3. Show Enrollments  4. Show Courses  5. Show Prerequisites  6. Show Logs  7. Add a student  8. Show a student's details including his enrolled classes  9. Show all prerequisites of a course  10. Show a class detail including students enrolled in that class  11. Enroll a student  12. Drop student from class  13. Delete a student  14. Exit  And calls different functions in one of other java class files below to perform appropriate action. |  |
|  | ShowStudents | Establishes connection to oracle server and calls stored procedure show\_students |  |
|  | ShowCourses | Establishes connection to oracle server and calls stored procedure show\_courses |  |
|  | ShowPrerequisites | Establishes connection to oracle server and calls stored procedure show\_prerequisites |  |
|  | ShowClasses | Establishes connection to oracle server and calls stored procedure show\_classes |  |
|  | ShowEnrollments | Establishes connection to oracle server and calls stored procedure show\_enrollments |  |
|  | ShowLogs | Establishes connection to oracle server and calls stored procedure show\_logs |  |
|  | AddStudent | Establishes connection to oracle server and calls stored procedure add\_students with arguments sid, firstname, lastname, status, gpa, email taken from user input. |  |
|  | ShowStudentClasses | Establishes connection to oracle server and calls stored procedure show\_students\_classes with argument as sid taken from user input. |  |
|  | ShowPrereqRecur | Establishes connection to oracle server and calls stored procedure show\_prereq with argument as dept\_code and course\_no taken from user input. |  |
|  | ShowClassesStudents | Establishes connection to oracle server and calls stored procedure show\_classes\_student\_details with argument as classid taken from user input. |  |
|  | EnrollStudentInClass | Establishes connection to oracle server and calls stored procedure enroll\_student\_in\_class with argument as classid and sid taken from user input. |  |
|  | DropStudentFromClass | Establishes connection to oracle server and calls stored procedure drop\_student\_from\_class with argument as classid and sid taken from user input. |  |
|  | DeleteStudent | Establishes connection to oracle server and calls stored procedure delete\_student with argument as sid taken from user input. |  |

## 

## Other Information and References

## Assumptions:

### Table setups have been done using instructor provided sql file: proj2data2016.sql, i.e., the following tables exist with appropriate data:

Students(sid, firstname, lastname, status, gpa, email)

Courses(dept\_code, course\_no, title)

Prerequisites(dept\_code, course\_no, pre\_dept\_code, pre\_course\_no)

Classes(classid, dept\_code, course\_no, sect\_no, year, semester, limit, class\_size)

Enrollments(sid, classid, lgrade)

Logs(logid, who, time, table\_name, operation, key\_value)

**References:**

1. docs.oracle.com
2. [www.javatpoint.com](http://www.javatpoint.com)
3. www.tutorialspoint.com