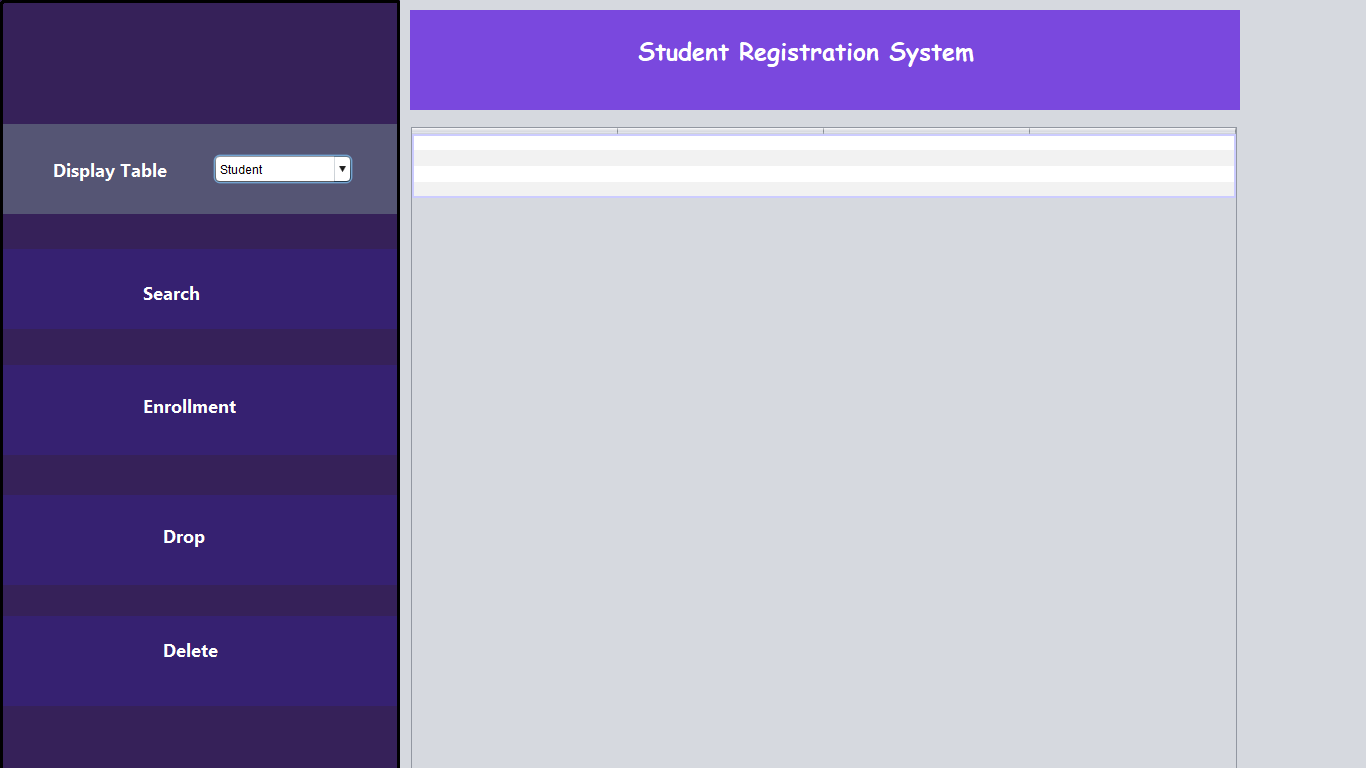
**TEAM REPORT**



By-

[Vinay Devarapalli - B00765886](mailto:hnikam1@binghamton.edu)

**INTRODUCTION**

Student Management System is software which is helpful for students as well as the school authorities. Proposed online student registration system will eliminate all the manual intervention and increase the speed of whole process.This project uses Oracle's PL/SQL and JDBC to create an application to support typical student registration tasks in a university.

**MEETING DESCRIPTION**

We decided to meet the following days since both of our schedules were same.

3 Nov – inserted the names in the table and completed the other prerequisites of the project

6 Nov – basically understood what the project requirement was

10 Nov – decided the outlook of GUI and the functionality

13 Nov – wrote the sequence and started with procedures. This took us a considerable amount of time.

15 Nov – started coding for the interface (the needed frames were developed)

17 Nov - GUI was fully developed and was connected to oracle database using JDBC

20 Nov - The stored procedures were called from our java code

23 Nov – Final step of writing Triggers was complete, checked them with help of log table.

24 Nov –Final part was testing the whole code as it passed through unit testing once modules were developed but whole system was never tested. So the exceptions which were needed to resolve were and solved.

25 Nov – Once the bugs were resolved we started with documentation

By 26 Nov we were ready with the student registration system and prepared for the demo along with reports.

**PROJECT COMPLETION SCHEDULE**

First, for this project, we started off with understanding what queries, procedures, triggers, sequences needed to be written. Then we did the Oracle Database setup on our personal computers. Once we were done with the setup, we decided that we shall at first draw an outline of the GUI and the functionalities the system would exhibit. Once the GUI was ready we would start coding for it and also side by side write the PL/SQL procedures. After the coding for GUI was done we connected it with the oracle database using JDBC. The functionalities were coded on connection and put into play. This included calling all the relevant procedures and triggers from the java code and displaying the result to the standard output. On completion of entire coding, we tested the functionalities of our system.

**FUNCTIONALITIES OF SRS:**

The functionalities our SRS implements are –

1. Display each of the seven tables on selection.
2. Search for TAs and output his B#, first name and last name. Messages are shown if a invalid class-id has been entered or if the class entered has no TA.
3. Search for prerequisites of entered course with dept code and course# as input from user. Similar to the one above, messages are displayed if any of the input parameter is invalid.
4. Enroll a student into a class with B# of the student and the class-id of the class provided as input parameters. Messages are displayed accordingly for any of the enrollment issues.
5. Drop a student from a class taking in the class-id and B#.
6. Delete a student from the Students table based on the given B# (as a parameter).

.

**CONTRIBUTIONS**

As planned, Hitesh started initially with the designing of GUI. The frames - homepage and designpage have been beautifully created by Hitesh and the others –search page, enrollmentPage were designed by Himani. The delete and drop functions were done together. While Hitesh was coding his part of GUI, Himani was working on the sequences and procedures. After the development of GUI we started with the coding for required functionalities. Each of us worked together on the functionalities. Once all the functionalities were successfully implemented in the code, we executed the written procedures in SQL initially and made changes if any necessary. This was the one part of the system that was time consuming. These were then added to the java code. We ensured that our interface utilizes every part of the PL/SQL code.

**THANK YOU**