Explanation

The University Management System application program is implemented by Windows Builder, with the database embedded by JDBC. And the system has a quite user-friendly interface. There are many very meaningful queries and modifications implemented in our application, which can get interesting information from the database. We also include various kinds of complex SQL clauses to make the application more useful, such as subqueries, aggregates, set operations, etc. Our University Management System application makes it easier to get access to the database for people who even do not have a great knowledge of database systems. And it is show in such aspect as below.

The main menu and architecture

The main menu is divided into three parts:

1. Login and quit part, which include the buttons “Connect SQL” and “Close”;
2. Queries part, which include the buttons “Large Club”, “Good Stu.”, “Prof Names” and “Interests”;
3. Modification part, which include the “Update” button for executing different kinds of modifications to the databases, such insert, delete, update, etc.

Easy Connection and disconnection

We design a connect class which provides the user an easy and safe way to log in to the database. It can perform well for connecting and disconnecting to the database. For login, you just need to input your username, password and click confirm in the popped window after you click the “Connect SQL” in the main menu. It can also check whether the login is successful or not. If it connects to the database successfully, you will see such message showed in the textbox "Database connection established". Or you it will let you know that the login failed by showing "Cannot connect to database server". This information will make users to use the database easier.

For disconnection, you can just conveniently quit the University Management Systems by click “Close” button in the menu.

Interesting queries implemented

a) As social activities play a significant part in students’ daily life, we implemented a query to show which club is the most popular among students in the university. We quantify the popularity of a club by the capacity of a certain club. Therefore, we can get the most

popular club by select the club with the largest capacity. We implement this query as a "Large Club" button in the application, which can give us the most popular club's name and its capacity.

b) There are many students who work fairly hard to their majors. In order to find these students in each department. We design the "Good Stu." frame to find such students. Each department may have a core course which is very important for the study of that major. So when you click "Good Stu." button, you will be directed to a new interface showing different departments. Just by clicking one department, you can get the information about the student who gets the highest score of the core course of that department.

c) There are many professors in various departments in the university. Easily getting access to information about professors classified by department is quite meaningful, so we implemented "Prof Names", you can conveniently check professors in each department just by clicking "Prof Names" button and then the department you want to check.

d) Communications with colleagues will surly improve the efficiency of conducting research. Therefore, we implemented the "Interests" frame to get research interests in a certain department.

Flexible Update

As database is nearly always dynamic, operations to update the database is very significant, such as insertion, deletion, updating of certain tuples, etc. As the update part for SQL is fairly easy that you just need to write the operation you want and the corresponding values, our application give users a lot flexibility to perform the update. You can just by write the easy update SQL clause to perform changes you want to execute, and our application will send this message to the databases and execute the modification you want. And whether the modification is done successfully will be show in a textbox of our interface.