MINIMUM APPLICATION LOGIC REQUIREMENTS

In each project, for at least one user (say, admin, officer, clerk, whoever), you must implement typical admin scenarios: add/delete/update and view all tables, make fixed and parametric queries, view useful reports. Please read the discussion about QUERIES and REPORTS at the END of this REPORT!

Additionally, for correct functionality, each project should satisfy some criteria. We give a minimum subset of such requirements that should be supported; but you can of course imagine others by considering your own application. Note that, your main switchboard (or form) may support the requirements of different user scenarios by providing items (buttons) like “admin operations”, “clerk operations”, “student operations”, etc. Of course, each of these items may also include several other submenus that make operations on corresponding database tables and achieve the required functionality.

**Car Service Management System**

- The receptionist should be able to add/delete/update new customers, cars, service orders, etc. The department employees should see cars that are waiting for repairment, and again add/del/update the service operations and spare parts. The “status” attribute of the car should change accordingly (“just arrived”, “in repair”, “finished”, etc). The receptionist should also see the “finished” cars and obtain bills for them.
- The system admin should be able to see reports and queries about how well the entire service doing (e.g., the number of cars repaired per month, the spare parts used, total revenue, etc). Of course, the admin has additional rights with respect to receptionist or department employees, (s)he can add/delete/update anything (i.e., departments, employees, stocks of spare parts, etc.).
- Let’s assume the customer can also login to system. He can query for his car to see its status and the (current) bill.

**SuperStore Data Management System**:

- When you buy a stock of a product, you will insert it to your stock table and you have to update the available quantity value of the product in the product table.
- Add “amount” field to “includes” relation between Product and Transaction. For a sale operation you should check the availability of the product and only complete the sale operation if the requested amount of products is available. (You need to check the available quantity value of the product). If the sale operation is done successfully then you have to update the available quantity values of the products again.
- Be careful about the campaigns. Since campaigns decrease the unit price of some products for a date interval, do the total amount computation in the sale operation by considering the possibility of a campaign for this product. If there is a campaign of this product on the sale date, then apply the campaign price of the product.
Museum Data Management System:

- The officer should be able to add/delete/update new artworks, artists, etc. In addition, he/she should be able to adjust the salons for exhibitions at particular seasons. The security should see which salons he/she is responsible for, and the events in these salons. The art experts should see the current restorations they are working on and be able to modify these.

- The system admin should be able to see reports and queries about how well the exhibitions and events for this museum is going (e.g., the number of exhibitions at a particular season, the number of exhibitions in a salon, total number of artworks exhibited at a salon/season, total number of events in a month and the artists that reserve these event salons, etc.). Of course, the admin has additional rights with respect to other museum staff, he/she can add/delete/update anything (i.e., officers, security, art experts, salons, seasons, etc.).

- The artist can also login to system. He/she can reserve an event salon for a special demonstration.

Online Restaurant Order System:

- The user should be able to login to the system. The user should see the restaurants that are open and serve to the district of this user. The user should be able to order several products from a restaurant. After giving an order, user should be able to display past orders and review/comment on them (within a week).

- Provide an admin panel for restaurants. This panel should display the products that can be ordered in a restaurant. Admin can add new products and remove existing ones. Admin can also modify the properties (price, description etc.) of the existing products. Admin should be able to view delivery staff working for that restaurant. Admin should be able to adjust their salary and fire/hire them. Admins should be able to view delivery staff who have received an extra payment. Finally, admins should be able to see detailed reports of for the restaurant (most preferred products, most busy days, number of products sold in a month etc.)

- Once an order has been placed by the user, system should automatically check the delivery staff working for the corresponding restaurant and assign this order to the delivery staff with the least amount of deliveries on that day.

Queries: While achieving the above functionalities, you will need several queries embedded to your application: That is, buttons in your GUI will execute selection queries (that could be fixed or parametric; using one or more tables) as well as modification queries (i.e., insert/del/update certain information in one or more tables). You may write a few lines of VB code (or macros) so that a particular button click will execute more than one queries (e.g., insert a tuple first to the person table and than the employee table). Also, you will need explicit queries to allow sophisticated
searches. You should have fixed queries, parametric queries with un-exact search conditions (e.g., show products with names similar to “AA”) and range conditions (e.g., show products sold between these dates, etc.) as well as typical parametric queries (e.g., show products with particular product id).

Reports: Reports are queries that are executed periodically and usually provide overview information in a neatly organized way (i.e., with nested headings, using charts, etc.). For instance, provide reports of most sold products monthly, top-10 employees of each month, percentages of product categories sold each month shown in a chart, etc.