Which one of the database systems is needed by your system? who will be responsible of handling the process of building a database (back end or front end)?Are they part of system? or they will be used on base-role contact? Who will determine whether the database system to be build in house or outsourcing? Which role is played by the database system administrator or developer during the system engineering , development cycle.

These are samples of an important discussion topics related to this week chapters as well as to the whole process of system development. I like to see you using your own words for expressing your understanding in these areas.

Ideally, I would like to use an ODBMS for my system. One of the most frustrating aspects of Object-Oriented programming is having to map it to a relational database. The control layer ends up consumed with porting relational data from a collection of normalized tables into objects that must be converted back to relational structures to perform persistence actions.

The DBA and his or her assistants would be responsible for building the database structures, both the table structures and the views and procedures for the application to interface with them. Everything must be approved by the DBA prior to going into production.

I must agree in part with Robert Franzese observation that, "there is such a thing as over-normalizing a database." I have experienced this first hand. We converted our partially-normalized database to fifth normal form, and the result was to turn a hundred tables into a thousand with accompanying rules and procedures to maintain them. As a result, database performance suffered dramatically, and we had to set up denormalized tables to pull data from to regain some of our efficiency, but at the expense of having to maintain redundant data.