**Description of the Architectural Drift and Erosion**

First Level Decomposition (Relation between Subsystems)



Figure : High level Subsystem Decomposition

Second Level Decomposition (Subsystems and Components)

**Client Side Subsystems**



Figure : User Interface subsystem

**Server Side Subsystems**



Figure : Security Management subsystem



Figure : Database Management subsystem



Figure : Social Media subsystem



Figure : Food Log subsystem



Figure : Login Management subsystem

Third Level Decomposition (Classes with Significant Drift and Erosion)

[insert all diagrams highlighting changes with respect to change in interacting with DB]

The most significant changes observed in the process of extracting the concrete architecture were in database management subsystem. This was so because initially we assumed that the Food Log, Social Media and Login Management subsystems would have their own database management classes. However, during implementation we used a common class named ‘DatabaseConnectionServiceImpl’ because all the other subsystems use the same interface (‘DBConnectionServiceAsync’ and ‘DBConnectionService’) to make RPC calls to the database. This is the rationale for this major difference in many of our class diagrams.