General Requirements

# 1.0 Database Design Requirements

## 1.1 Database Schemas

1. Each Database Schema should manage a single area of concern and attend the smallest possible number of independent business entities for example the “Entity” schema manages the following business data objects:
   1. Organization – Agencies, Corporations, Institutions, or any other unit which conducts sort of business or operations.
   2. Officer – an Agent that works and/or provide services for an Organization.
   3. Person – details about a Human being.
   4. Contact – a data concept for a person or organization to contact.
2. Each Database Schema should be small, self-sustained and be isolated from other schemas.
3. A Schema could be removed from a Database and be placed in another Database without any modifications to any table, constraint or resource and be fully functional and accessible by just repointing the SQL instance.
4. Other data objects not in the Schema should be accessible only through a sort of a “Reference ID” and no FOREIGN KEY or any other dependency should be placed upon those. Referential integrity should be managed by an externa resource that may keep track of all data objects.
5. Duplicate Code Sets or some other duplication may be required to achieve schema isolation.
6. Data Objects within a Database Schema can fully reference and be dependent on each other as needed.

## 1.2 Unique Identifiers

1. All ID’s should be generated by a reliable redundant singleton service upon request and will guaranty that no ID is repeated.
2. Numeric Database generated ID’s cannot be used since a schema could be moved and/or merged with other identical schemas at any time.
3. While merging with other identical schemas there will be no conflict with any identifier since all should had been generated with the same singleton.
4. The singleton ID generation service could be substituted with any method that will guaranty the required uniqueness.
5. ID’s should be registered with additional metadata identifying its target reference object and identity.
6. Generated ID’s should be able to be used to search for related business objects.
7. The ID generator service should allow the request of a large number of instances upon a single call to be used in mass insertions.
8. An ID that had been used to identify an object or data entity should not be ever reused.

# 2.0 Functional Areas Requirements

## 2.1 User Session Management

1. Upon login, into the system and internally a corresponding time-limited "Session" is created.
2. Associated with the Session a "Session\_ID" is created along with other relevant session information (see Application.Session).
3. The session life is bounded by time or usage, a valid session is requested upon login, and it is closed and no longer valid after logout or the max session lifetime expires as configured.
4. The session ID should be recorded during all database transactions done to any database object during the lifetime of the session (see the “Session\_Updated\_ID” column).
5. It is possible that a (period-limited) Access-Token associated to the Officer is provided in the request and used to validate that the user is authorized to login (see Application.AccessToken).

## 2.2 Follow-Up List

1. Follow-Up or Checklists should be available and presented to Managers, Officers, or Users to work through pending requests, follow-up tasks or other related actions (see sample list in 4.0 (2)). These lists help the Users to keep track of pending tasks, priorities, or as a reminder of things to do.
2. Lists are configurable therefore, Managers or assigned Officer should be able to create, update or retire a list or items within a list.
3. Within a list, Managers should be able to assign tasks to Officers.
4. An Officer should be able to add or move items from a list to another as progress is made or as needed.