## 1. Design Considerations

* need to know the users or actors of the app
* need to identify
  + data that can be accessed by users
  + data restrictions and revoke access from such sensitive data
  + users who should be allowed to customize the app and assign the necessary permissions

## 2. Managing your user’s experience

* Types of licenses
  + every user must have a user license
    - defines the fucntionality permitted for the user and determines the profile available to the user
    - can have only type of user license, but may have many feature licenses
  + Two types of license
    - Salesforce
      * full access to CRM, force.com appexchange apps, standard or custom apps
      * full access to CRM, force.com appexchange apps, standard or custom apps
    - Salesforce Platform
      * custom apps, force.com appexchange apps, no access to CRM functionality, can still use accounts, leads, contacts, reports, dashboards and documents
  + can have add-on features, such as apex mobile user, sf crm content user, marketing user
  + can have more than one type of feature license
* Profiles
  + define user permissions to perform different functions
  + each profile is associated with a license type
  + profile work with sharing models or role hierarchy
* profiles control permissions, access to data and the UI
  + Permissions
    - define all actions that a user in a profile can perform
  + access to data
    - controlled by field level security settings which allow to define object and field level permissions
  + user interface
    - page layouts, tabs, applications available for each user, record types for each profile determine what the users will see when they login
  + two types of profiles
    - standard
      * can’t be created or deleted and permissions cannot be customized
    - custom
      * create or clone the standard profile and modify the settings
  + types of permissions
    - Administrative
      * can grant some administrative permissions to custom profile
    - General User
      * control the ability that standard user can do like editing tasks
    - Standard Object
      * control read, create, edit and delete action on standard objects
    - Custom Object
      * control read, create, edit and delete actions on custom objects
    - View All Data allows administrators to view all records regardless of all other security settings
    - Modify all Data – allows administratos to modify all records
    - Customize Application permits administrators to administer the application
    - API only user cannot login to sf.com. such users can only use the application through API calls
    - Password Never Expires prevents password expiring
  + Permission Sets
    - collection of settings and permissions
    - represent a concept like job title
    - handle the system requirements that previously existed on the profiles
  + user can have only one profile, but can have multiple permission sets
  + while assigning permissiosn to users, use profile to assign most restrictive settings and assign additional permissions using permission sets
  + an org can have upto 1000 permission sets
  + permission sets can be used assign additional permissions, but not to deny permissions
  + permissions can be revoked by
    - removing permission from permission set
    - changing a user’s profile
    - disabling a permission set
  + use permission sets to grant permissions for
    - applications, objects, fields, tabs, apex classes, service provider, visual force pages
  + permissions that are not available in permission sets must be set through profiles (login hours, ip access, etc)
  + revoking delete permission for a child object in master child relationship will not prevent deleting the child record if parent record is deleted
* Field-Level Security
  + restricts access to fields
    - list views, reports, force.com, conenct offline, custom links, mail merge, related lists
  + overrides less restrictive page layout settings
  + set at profile level
  + each profile can have different level of access to object
  + doesn’t allow conditional security of records
  + all users can edit any accessible fields
* Customizing UI and Profiles
  + Record types
    - define the manner in which data is displayed according to business needs
    - determine the page layout and limit the picklist options based on the profile
    - not security tools – do not subclass or partition the data, they work at the UI level and not at the data level
    - users can change the record type of an existing record and it does not affect the values in the record

## 3. Controlling access to records

* Record ownership
  + has a owner, sharing based on owner of a record, can be transferred to any user who atleast has read access
  + child records in master detail relationship do not have owner, they inherit from parent record
* Types of owners
  + users – full access if a user is owner. if read permission revoked, then they can’t see their own record
  + queues – allows multiple ownership, assigned manually or thru assignment rules
* Record Access
  + read only access, read-write access, full access
* Ways to obtain record access
  + Full access: owner, above the owner in role hierarchy, contains modify all data permission in profile
  + Read/Write or Read only: owd, role hierarchy, sharing rules, manual sharing, apex sharing, view all data
* Profiles vs Sharing Models
  + profiles
    - control access to objects and fields
    - whether user can view positions
    - which fields the user can view
  + Sharing models
    - control access to records
    - control the positions to view
* OWD
  + security settings that define the base line level of access to records that the user doesn’t own
  + only way to restricts access to data in sharing model
  + 3 level of settings
    - public read-write
    - public read only
    - private
  + Determining OWD
    - identify the most restricted user of this object
    - will there be an instance of object that this user is not allowed to view
    - if yes, then owd is private
    - else,
      * will there be an instance of object that this user is not allowed to edit
      * if yes, then owd is public read only
      * else, then owd is public read-write
  + setting owd for child records in master-detail relationship – child inherits  owd from parents
  + child records in lookup will not inherit owd
  + it is possible to change owd any time, but it may have consequences
  + owd can be set for both standard & custom objects
* Roles
  + control the level of visibility to org data
  + every user associated to role
  + assuming no sharing rules created, users in the same role cannot access each other’s records
* Role Hierarchy
  + defines data access rights granted to users at higher roles
  + users access to all records they own and their sub-ordinates
* record access rolls up with role hierarchy with all standard objects
* with custom objects, a setting named ‘Grant Access using Role Hierarchy’, this can be prevented
* Public Groups
  + Roles are two dimensional structures. Public groups are way of grouping users together to grant them record access.
  + Groups are good way to extend access across the nodes in hierarchy tree
  + ‘All Internal users’ is a default public group. Public groups can be made up of any combination of users, roles and subordinates and other public groups.
  + Can use public groups in a sharing rule to reduce the number of sharing rules
  + Public groups can also be used for folder access.
* Sharing rules
  + are created to grant access to records between users when access does not roll up
  + Using sharing rules, read only and read/write access can be granted to users
  + Sharing rules cannot be more restrictive than owd settings.
* Manual Sharing
  + used to grant access to records on a one-off basis when randon users require record access.
  + Access rights can be granted by the owner of a record, anyone above the owner in the role hierarchy and by the system adminsitrator
  + It is granted at the record level and is not used to grant access at the organization level
* Apex sharing reasons
  + allow developers to define the reason why a user or group of users have access to record
  + apex sharing reasons exist only for custom objects and they are defined for individual objects.
  + each object can have up to 10 apex sharing reasons
  + sharing rule has to be created manually using new manual sharing rules
  + deleting apex sharing reasons will delete all manual sharing rules associated with it
  + Users with ‘Modify all data’ permission can change sharing using apex sharing reasons
  + apex sharing reasons should be used programatically and not through the application

## 4. Designing Data Access Security

* Establishing Data Access
  + When you want to determine data access for a object,
    - consider the OWD default
    - the owner of the records
    - uses who need access
    - rules governing data access
  + When determining access to sensitive data, you need to analyze the access requirements and restrictions for each profile