SubSystem title

# ERD for UX process

Place your ERD diagram here.

# Possible UX Interface

Place your screen design(s) here

|  |  |  |
| --- | --- | --- |
| **Controls** | | **Events** |
| control name | | web page event or ODS message |
| **BBL Class(es) and Method (s)** | | |
| controller name | method header  statement about the method’s purpose. may include inputs and outputs | |
| **SQL Table(s): (C,R,U,D)** | | **Entities/DTOs/POCOs** |
| name of sql table involved and access | | name of any new DTO or POCO class |

# Event and Wiring Summations

# Samples

|  |  |  |
| --- | --- | --- |
| **Controls** | | **Events** |
| AlbumDDL | | None / wired via ObjectDataSource |
| **BBL Class(es) and Method (s)** | | |
| AlbumController | List<SelectionList> List\_AlbumTitles()  Retrieve a list of album titles and album id for DDL | |
| **SQL Table(s): (C,R,U,D)** | | **Entities/DTOs/POCOs** |
| Albums (R) | | SelectionList (P) |

|  |  |  |
| --- | --- | --- |
| **Controls** | | **Events** |
| Fetch (Artist, MediaType, Genre, Album),  Tracks, TracksBy, SearchArgID | | OnClick / wired via ObjectDataSource   * Move selection value to TracksBy, DDL id to SearchArgID; * Bind data to Tracks |
| **BBL Class(es) and Method (s)** | | |
| TracksController | List<TrackList> List\_TracksForPlaylistSelection(string tracksby, int argid)  Retrieve a list of tracks for supplied selection type and argument id. Return trackid, name, title, MediaName, genreName, composer, timelength, Size unitprice. Display in Tracks. | |
| **SQL Table(s): (C,R,U,D)** | | **Entities/DTOs/POCOs** |
| Tracks (R) | | TrackList (P) |

|  |  |  |
| --- | --- | --- |
| **Controls** | | **Events** |
| Fetch (PlayList), Playlist, PlayListName | | Fetch OnClick   * Validate data present; * call BLL; * Bind returned data to PlayList. |
| **BBL Class(es) and Method (s)** | | |
| PlaylistTracksController | List< UserPlaylistTrack > List\_TracksForPlaylist(string playlistname, string username)   * Retrieve a list of tracks for supplied playlist name and username. Return trackid, title, timelength, unitprice, tracknumber. Display in PlayList.   **\*ANOTHER WAY TO EXPLAIN THE METHOD\***   * Check for existing playlist * Optional : Add if playlist does not exist * Check for track on playlist * Error: If track exists, put out an error message * Add Track to playlist | |
| **SQL Table(s): (C,R,U,D)** | | **Entities/DTOs/POCOs** |
| PlayList (R), PlaylistTracks(R) | | UserPlaylistTrack (P) |

|  |  |  |
| --- | --- | --- |
| **Controls** | | **Events** |
| ArtistDDL | | None / wired via ObjectDataSource |
| **BLL Class(es) and Method (s)** | | |
| ArtistController | List<SelectionList> List\_ArtistNames()  Retrieve a list of artist names and artist id for DDL | |
| **SQL Table(s): (C,R,U,D)** | | **Entities/DTOs/POCOs** |
| Artists (R) | | SelectionList (P) |

|  |  |  |
| --- | --- | --- |
| **Controls** | | **Events** |
| TrackSelectionList, PlayList, PlaylistName | | Code-behind   * Collect track information – playlist name and user name * Validate * Send valid data to BLL * Refresh the playlist display |
| **BLL Class(es) and Method (s)** | | |
| PlaylistTracksController | 1. void Add\_TrackToPlaylist (string playlistname, string username, int trackid)  * TRX (Create a playlist if needed and add a track to the playlist)  1. List< UserPlaylistTrack > PlaylistTracks\_GetByPlaylist (string playlistname, string username)  * Retrieve the playlist for the user via playlistname | |
| **SQL Table(s): (C,R,U,D)** | | **Entities/DTOs/POCOs** |
| PlaylistTracks(C,R), Playlists(C – optional, R), | | UserPlaylistTrack(P) |