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, Compsoer, 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. | |
| **SQL Table(s): (C,R,U,D)** | | **Entities/DTOs/POCOs** |
| PlayList (R), PlaylistTracks(R) | | UserPlaylistTrack (P) |

|  |  |  |
| --- | --- | --- |
| **Controls** | | **Events** |
| ArtistDDL | | None / wired via ObjectDataSource |
| **BBL Class(es) and Method (s)** | | |
| ArtistController | List<SelectionList> List\_ArtistNames()  Retrieve a list of artist name 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 username * Validate * Send validate data to BLL * Refresh the playlist display |
| **BBL Class(es) and Method (s)** | | |
| PlaylistTracksController | void Add\_TrackToPlaylist(string playlistname, string username, int trackid)  TRX(create a playlist if needed, add a track to the playlist)  List< UserPlaylistTrack > PlaylistTracks\_GetByPlaylist(string playlistname, string username)  Retrieve the playlist for the user via playlist name. | |
| **SQL Table(s): (C,R,U,D)** | | **Entities/DTOs/POCOs** |
| Playlist(C opt,R), PlaylistTracks(C,R) | | UserPlaylistTrack(P) |

|  |  |  |
| --- | --- | --- |
| **Controls** | | **Events** |
| PlaylistFetch, PlayList, PlaylistName | | Code-behind   * Collect playlist name * Presence validation * Send to BLL * Refresh PlayList with results |
| **BBL Class(es) and Method (s)** | | |
| PlaylistTracksController | List< UserPlaylistTrack > PlaylistTracks\_GetByPlaylist(string playlistname, string username)  Retrieve the playlist for the user via playlist name. | |
| **SQL Table(s): (C,R,U,D)** | | **Entities/DTOs/POCOs** |
| PlaylistTracks (R), Tracks (R) | | UserPlaylistTrack (P) |

|  |  |  |
| --- | --- | --- |
| **Controls** | | **Events** |
| MoveUp, MoveDown, PlayList, PlaylistName | | Code-behind   * Verify tracks exist * Presence validation: playlistname * Validation only one track selected * Validation track must not be top/bottom track * Send to BLL * Refresh PlayList with results |
| **BBL Class(es) and Method (s)** | | |
| PlaylistTracksController | void MoveTrack(string username, string playlistname, int trackid, int tracknumber, string direction)   * Lookup the playlist id * Lookup the playlisttrack * Up   + Actions required * Down   + Actions required * Update to database * Save changes   List< UserPlaylistTrack > PlaylistTracks\_GetByPlaylist(string playlistname, string username)  Retrieve the playlist for the user via playlist name. | |
| **SQL Table(s): (C,R,U,D)** | | **Entities/DTOs/POCOs** |
| PlaylistTracks (U,R), Tracks (R) | | UserPlaylistTrack (P) |

|  |  |  |
| --- | --- | --- |
| **Controls** | | **Events** |
| DeleteTrack, PlayList, PlaylistName | | Code-behind   * Verify tracks exist * Presence validation: playlistname * Validation at least one track selected * Send to BLL * Refresh PlayList with results |
| **BBL Class(es) and Method (s)** | | |
| PlaylistTracksController | void DeleteTracks(string username, string playlistname, List<int> trackstodelete)  Delete listed tracks from PlaylistTracks.  List< UserPlaylistTrack > PlaylistTracks\_GetByPlaylist(string playlistname, string username)  Retrieve the playlist for the user via playlist name. | |
| **SQL Table(s): (C,R,U,D)** | | **Entities/DTOs/POCOs** |
| PlaylistTracks (U,R), Tracks (R) | | UserPlaylistTrack (P) |