• Existing Functionality

o All functionality and project requirements from the previous assignment were successfully implemented and functioned correctly.

• Search Enhancements for Existing Pages

o A search section was added to the existing pages for Artist, Album, and Track.

o All records were shown on the first viewing of the page.

o When the user entered a search term, the page only displayed records that contained the selected search term in the name or title.

o LINQ queries were implemented to perform the searches.

• Music Catalog

o A new Music Catalog page was added to the application.

o Links to the new page included an image and were accessible from the Pages menu and the button toolbar. Buttons had descriptive tooltips.

o The Catalog could either show all data or be blank when first viewed.

o All data were displayed in the GUI page using databinding.

o LINQ queries were implemented to retrieve and aggregate all data.

o The Catalog data was grouped by the first letter of the artist’s name.

o Each artist grouping displayed a count of how many artists that group contained.

o Artist groupings displayed in a collapsed form initially, allowing the user to expand and view the artist data for that group if desired.

o If a group was expanded, all artists in the selected group were displayed, with expandable sections to display all that artist’s albums.

o For each album, an expandable section displayed all the tracks on the selected album. • Customer Order Search

o A new Customer Orders page was added to the application.

o Links to the new page included an image and were accessible from the Pages menu and the button toolbar.

o The page could either show data for all customers or be blank when first viewed.

o All data were displayed in the GUI page using databinding.

o LINQ queries were implemented to retrieve all data displayed on the page.

o For any displayed customer, a minimum of the following fields was shown: Full name as a single string, in last-first order. City and State information as a single string. Note: Data display accounted for the fact that some customer records had no state value. Country Email address

o Invoice data for a selected customer displayed in a collapsed form to start, allowing the user to expand and view the invoice data if desired.

o If the invoice data was expanded, it showed: The date the invoice was created. The total purchase amount of the invoice. The number of tracks purchased on the invoice

o All displayed data were formatted appropriately.

• Program UI, Project Architecture & Code:

o Project architecture (folders, files, namespaces, etc.) was clean and organized, according to conventions learned in class.

o Styles were defined as resources whenever appropriate.

o UI Styling elements were defined as resources, at an appropriate level, whenever possible.

**Opening Scene: Logo and Project Title**

Fade in from black. Your project logo appears, followed by the text: "Chinook WPF with EF: Enhanced Search and Catalog Features Update".

**Narrator:** "Welcome back to our project update on Chinook, the dynamic WPF application powered by Entity Framework. Today, we're excited to showcase the latest enhancements that elevate your music browsing experience to the next level."

**Scene Transition: Overview**

Cut to a screen recording of the application's home page.

**Narrator:** "Building on the solid foundation of our previous version, we've introduced powerful search capabilities, a comprehensive Music Catalog, and an intuitive Customer Orders search functionality. Let's dive into the details."

**Section 1: Search Enhancements**

Cut to Artist page with the new search section visible.

**Narrator:** "First up, we've added a search section to our Artist, Album, and Track pages. Initially displaying all records, the pages now dynamically filter results based on your search terms, thanks to LINQ queries powering our searches behind the scenes."

Show typing in the search box and the page updating with filtered results.

**Section 2: Music Catalog**

Transition to the new Music Catalog page.

**Narrator:** "Next, we're thrilled to introduce the Music Catalog. Accessible from the Pages menu, this new feature groups artists by the first letter of their name, offering a count of artists within each grouping."

Show the collapsed groupings, then expand one to reveal artists, albums, and tracks.

**Narrator:** "Each artist grouping can be expanded to explore albums and track listings, making it easier than ever to browse the extensive music collection."

**Section 3: Customer Order Search**

Switch to the Customer Orders page.

**Narrator:** "Our new Customer Orders page streamlines how you access and view customer purchases. Displaying essential customer details, including invoices in a collapsible format, this page enhances your ability to manage and review orders."

Demonstrate expanding a customer's invoice to show detailed purchase information.

**Section 4: UI and Architecture Improvements**

Show various UI elements, highlighting the clean and organized layout.

**Narrator:** "We've also made significant strides in improving our application's UI and project architecture. With a focus on clean design and organized code, we've ensured that our app not only works well but looks great."

**Closing Scene: Call to Action**

Return to the application's home page.

**Narrator:** "These updates to the Chinook project represent our commitment to providing a robust and user-friendly application. We're excited for you to explore these new features and see how they enhance your interaction with the Chinook music database."

Fade to the project logo.

**Narrator:** "Thank you for joining us for this update. Stay tuned for more enhancements and happy browsing!"

Fade out to black.