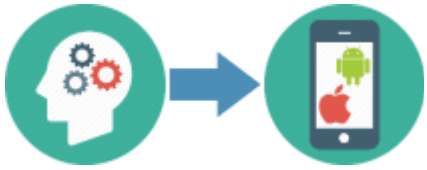
Mobile App Development

CS 4153



Weekly Assignment 10

Last updated: Oct 23, 2017 10:00 PM

Create an app that lets its user keep track of her/his music library.

---

1. Create a new, Swift 4 project in Xcode 9 production. Name the project
   W10_lastName_firstName
   where *lastName* is the part of your name that OSU considers to be your last name, and *firstName* is your first name.
2. The app only is required to run in portrait mode.
3. The app uses core data, so you probably want to specify this when creating the project.
4. The data model contains two tables (entities): *Album* and *Song*.
   - Album contains the attributes *title* (string), *artist* (string), and *year of release* (integer of some form), plus any other reasonable attributes you choose to include. It also contains a *To Many* relationship to Song, as an album can contain zero or more songs. This also has an inverse relationship, set up so that deleting an album deletes the associated songs.
   - Song contains the attributes *title* (string) and *song length* (in a numerical format), plus any other reasonable attributes you choose to include. This has a *To One* relationship back to its album, set up so that the pointer to the song is nullified in the album if the song is deleted.
   - Use Xcode to create custom NSManagedObject subclasses for Album and Song, and use these throughout your code to access the entities and their attributes rather than using key-value coding.
5. The app contains (at least) two view controllers: the *album list view controller*, and the *song list view controller*.
   - Each of these is a table view controller.
6. When the app begins, it displays the album list view controller. Each cell of the album list represents one album; the cell contains the album title as its first line, and the artist name (or "Various") and year of release as the second line, plus any other reasonable attribute values you choose.
   - The navigation bar includes a "+" (add) bar button item.
   - When the user taps on the add button, an alert controller with the ".alert" style is displayed; the alert controller permits the user to enter the title, artist, and year of release for a new album. It also provides buttons to permit the user to cancel or save.

- The "save" button is disabled until the user has provided valid, non-blank values for each of the title, artist, and year boxes.
- If the user taps "save" (after it is enabled), then trim any leading and trailing spaces from the values entered by the user, create an Album, populate its attributes, and then store it in the Album table of the database.
- The user can swipe-to-delete any album.

7. When the user taps on a cell in the album list view controller (or on an optional detail disclosure – it's your choice), segue to the song list view controller for that album. Each cell of the song list represents one song on the current album; the cell contains the song title as its first line, and the song length as the second line, plus any other reasonable attribute values you choose.
   - The navigation bar includes the associated album title as its title.
   - The navigation bar also includes a "+" (add) bar button item.
   - This view controller (but not the album view controller) has a tool bar at the bottom. This bar contains the total length of songs on this album, in a user-friendly format of *h:mm:ss*, where *h* is number of hours, and *mm* and *ss* are two-digit minutes and seconds, respectively.
   - When the user taps on the add button, an alert controller with the ".alert" style is displayed; the alert controller permits the user to enter the title and song length for a new song. It also provides buttons to permit the user to cancel or save.
   - The "save" button is disabled until the user has provided valid, non-blank values for each of the title and song length boxes.
   - If the user taps "save" (after it is enabled), then trim any leading and trailing spaces from the values entered by the user, create a Song, and then store it in the Song table of the database. The record also much be associated with the appropriate Album.
   - The user can swipe-to-delete any song.

General Notes

- Here is a short tutorial on creating swipe-to-delete table view cells.. You can get more information on this and other Core Data topics from other online sources and our iOS textbook.
- Feel free to customize your app to make it more attractive or user-friendly.

Submitting Your Solution

- Zip your project folder into a single file, go to the course BrightSpace site, navigate to the *Dropbox* page, and submit the zip file in the folder that corresponds to this assignment.