

# Mobile App Development - iOS Exercise 2

> So spannend kann Technik sein.



# Abstract

**Build an iOS Application that saves and displays hierarchical data (*artists* and associated *albums*), using the CoreData stack and UITableViews.**

## Modalities

Use a single view application template, with ARC & Storyboard.

Create a complete CoreData stack, either as a Singleton or inside the AppDelegate.

Create a DataModel containing the following models:

- *Artist* (name, label)
  - *Album* (name, year, format)
- (an artist has many albums)

Create two ViewController for displaying the list of saved artists and albums.

Create two ViewController to add/edit artists and albums.

# Album / Artist ViewController

Create two UIViewControllers to display the Artists and Albums inside a UITableView. The ViewControllers should contain:

- UITableView
- Button to add an artist/album (see “EditViewControllers”)
- Button to edit the tableView.
- a customized UITableViewCell (prototype cell)

Functionality:

- Add Button: Display a new ViewController to add Album/Artist data (see “EditViewControllers”)
- Edit Button: set the UITableView to editMode, allowing the user to delete rows or edit existing rows.

Navigation:

- Tap on a artist-row (no edit-mode) display tableView of albums associated with the selected artist.
- Tap on artist-row (in edit-mode) display EditController for that artist.
- Tap on album-row (no edit-mode, in edit-mode) displays EditController for that album.

# EditViewControllers

Create two ViewController for adding / editing the Albums/Artists.

ViewController should contain:

- Cancel Button (navigate back)
- Save Button (saves the entity and navigates back)
- TextField and Labels for every property of the Album/Artist model (hardcoded).

Fields should be pre-populated, when editing an existing record.

# Additional Information

To display only albums that are associated with a given artist, you can use the following NSPredicate (given that you have a reference to the selected artist):

```
NSPredicate *predicate = [NSPredicate predicateWithFormat:@"artist == %@", self.artist] ;
```

Use a NSFetchedResultsController to bind the UITableViews to the managed-objects.

To set a UITableView in edit-mode, you can use the `setEditing:animated` method of the UITableView.

# Mockups



# Grading

CoreData Stack	1.5 PT
TableViews (Album,Artist) + Delegate methods NSFetchedResultsController + delegate methods	3 PT
EditViewControllers (Album,Artist) Delegate Pattern	2 PT
Insert / Update ManagedObjects via EditViewControllers	1.5 PT
Delete ManagedObjects via TableViews (commitEditingStyle)	1 PT
custom UITableViewCell (prototype-cell)	1 PT