Criteria	Does Not Meet Specifications	Meets Specifications	Exceeds Specifications (Completely Udacious)
Basic Functionality The app contains two scenes of content: one for recording an audio file, and one for playing the audio with different effects. All UI elements (buttons and text) are appropriately formatted for iPhone portrait layouts.	The app does not have the basic structure specified. UI elements are not properly positioned on the screen.	The app contains the two pages of content and uses UINavigationController to navigate between these two scenes. UI elements are appropriately positioned on the screen.	*Not Applicable*
Actions and Outlets The app uses IBAction methods to record audio and playback sounds. Labels and buttons are shown or hidden as appropriate.	The app does not successfully connect buttons in the storyboard to actions in the code. The "Recording" label or the "Stop" button are visible even when recording is not in progress.	The app correctly connects each button to its own IBAction method. The "Recording" label and the "Stop" button are hidden by default, but appear when recording is in progress.	*Not Applicable*
AVAudioRecorder The first scene of the app uses AVAudioRecorder to record audio and keeps track of that data using a custom model class.	The app does not record audio successfully.	The app uses AVAudioRecorder to record audio and saves it using a custom Model class.	*Not Applicable*
Delegates and Segues The app uses the audioRecorderDidFinishRecording function to determine when the audio has finished recording. It also programmatically triggers a segue from the first scene to the second by using the performSegueWithIdentifier function.	The app does not use audioRecorderDidFinishRecording. The app uses a hardcoded Storyboard segue.	The app uses the delegate pattern and implements the audioRecorderDidFinishRecording function. The app invokes performSegueWithIdentifier in code.	*Not Applicable*
UINavigationController The app allows users to re-record audio after a recording is complete.	The app does not allow re-recording. Once the recording is complete the sound is fixed.	The app allows the user to re-record.	The app allows users to pause and resume recording.
Sound Effects The second scene of the app contains the following audio effects: Chipmunk, Darth Vader, Slow, and Fast.	Not all four buttons are present. The buttons are present, but the app doesn't play the sounds associated with the buttons.	The four buttons and the associated sounds are present and work.	The app showcases at least one additional audio effect, such as echo or reverb.
Code Improvements This Code Improvement document lists four tasks that you must complete on your own to improve the code written in this class. Your final code implements all four of these tasks.	One or more of these tasks is not attempted, or not implemented satisfactorily: • The model class uses an initializer, and this initializer is called in RecordSoundsViewController • The bug where sound effects overlap during playback is removed • Legacy, commented-out code is deleted • Meaningful information, such as a Tap to Record button, is provided to guide the end-user.	All four of these tasks are implemented satisfactorily: • The model class uses an initializer, and this initializer is called in RecordSoundsViewController • The bug where sound effects overlap during playback is removed • Legacy, commented-out code is deleted • Meaningful information, such as a Tap to Record button, is provided to guide the end-user.	*Not Applicable*
Code Quality Code is at a professional quality level, using control statements, methods, and comments appropriately and effectively. Code adheres to Swift naming and style conventions.	Control statements (for, while, if, else, switch) are not used appropriately. Code is unnecessarily repetitive and not effectively abstracted into reusable methods. Arguments and return values are not well aligned with their method's purpose. Code is not commented and/or does not adhere to Swift style conventions.	Control statements (for, while, if, else, switch) are used appropriately. Repeated blocks of code are contained in methods. Arguments and return values are appropriate. Code is well-commented. Code adheres to Swift style conventions.	*Not Applicable*