

Mobile Application Development
Aileen Pierce

TABLE VIEWS

Table Views

- A table view presents data in a scrolling single-column list of multiple rows
- Used to navigate through a hierarchically structured data

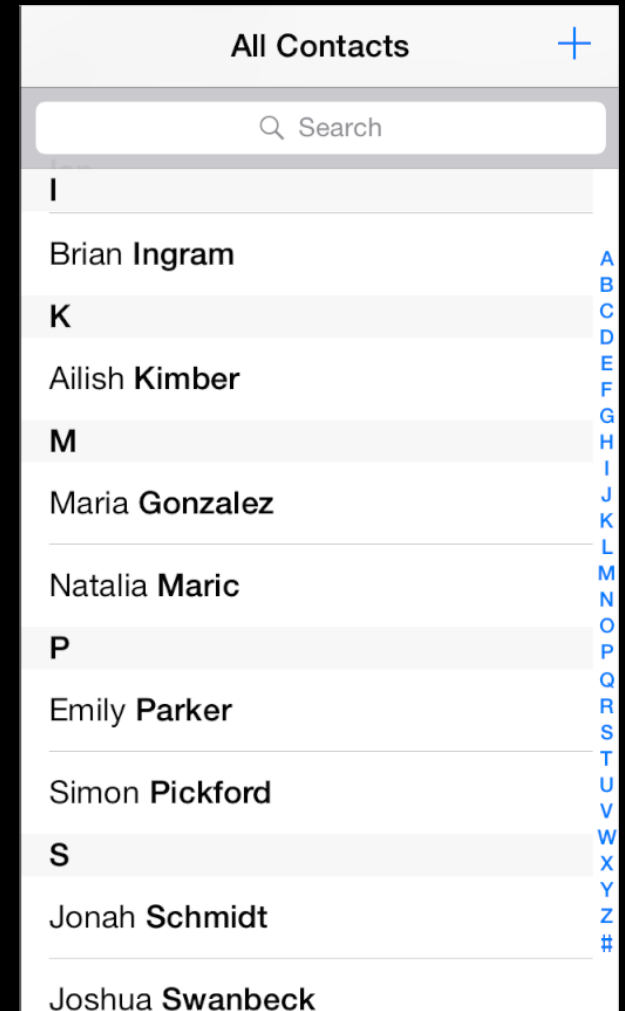


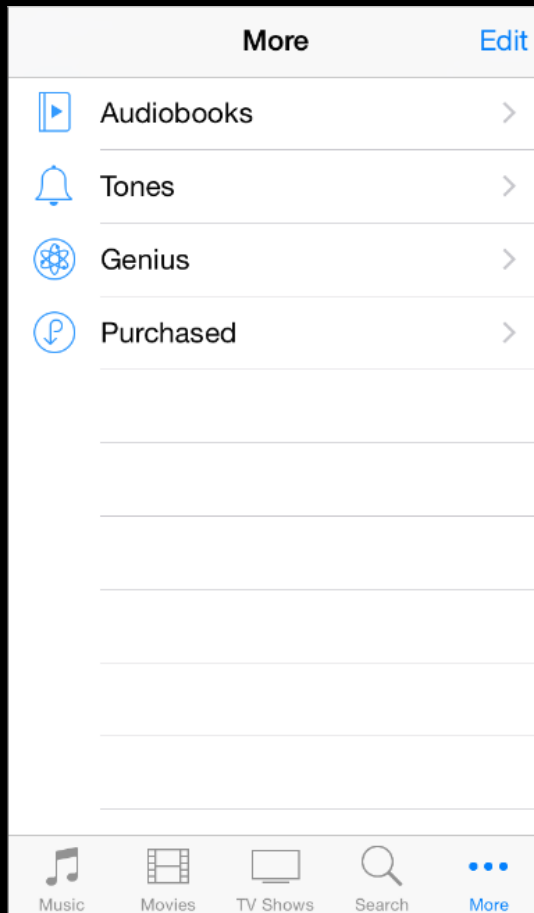
Table Views

- Table views can display
 - a selectable list of options
 - hierarchical structured data
 - conceptually grouped information
 - Indexed list of items
- Table views are usually combined with navigation controllers to chain together hierarchical lists of data

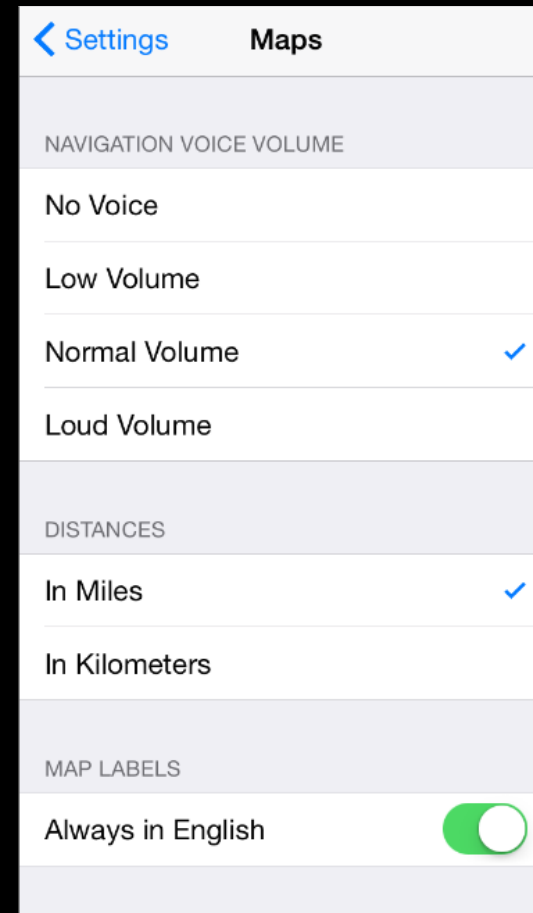
Table View Styles

- Table views have two styles: plain or grouped
 - The plain style can also have sections with an index along the right side
 - The grouped style must have at least 1 group and each group must have at least 1 item
 - Both styles can have a header and footer

Table View Styles



Plain



Grouped

Table Views

- The **UITableView** class displays a table view
- The **UITableViewController** class manages table views
- The **UITableViewCell** class manages the cells in a table view
- The **UITableViewHeaderFooterView** class implements headers and footers in a table view

Table View Delegates

- The **UITableViewDelegate** protocol handles configuring and managing table views.
- The **UITableViewDataSource** protocol handles the row data in table views.
- Table views can have an unlimited number of rows.
- Table views can display only a few rows at a time.







Table Views







- Table views avoid having to load all rows as some might never be seen.
- As rows scroll off the screen they are placed in a queue to be reused.
- `dequeueReusableCellWithIdentifier(_ , forIndexPath)` dequeues an existing cell if one is available or creates a new one
 - The reuse identifier is a string used to identify a cell that is reusable
- Reusing cells is the best way to guarantee smooth table view scrolling performance.







Table Views

- When using table views
 - Use succinct text labels so it's not truncated
 - Provide feedback when a user selects an item
 - Display data as soon as possible
 - Show what data you have while the rest is loading
 - Show 'stale' data while waiting for new data to load
 - Use a spinner to indicate that data is loading

Table View Cell Styles

Default Cell Style	
	Text Label
	Dahlia
	Daisies
	Dandelion
	Echinacea
	Lavender

Subtitle Cell Style	
	Text Label Detail text label
	Dahlia This is a dahlia
	Daisies These are daisies
	Dandelion This is a dandelion
	Echinacea This is echinacea
	Lavender This is a field of lavender

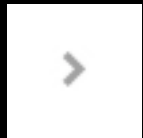
Value 1 Cell Style		
	Text Label	Detail text label
	Dahlia	This is a dahlia
	Daisies	These are daisies
	Dandelion	This is a dandelion
	Echinacea	This is echinacea
	Lavender	This is a field of lavender

Value 2 Cell Style		
	Text Label	Detail text label
	Dahlia	This is a dahlia
	Daisies	These are daisies
	Dandelion	This is a dandelion
	Echinacea	This is echinacea
	Lavender	This is a field of lavender

- `textLabel` holds the larger text
- `detailTextLabel` holds the smaller text
- `imageView` stores a `UIImage`

Table View Cell Accessories

Table View Cell accessories. The default is none.



Disclosure indicator

`UITableViewCellAccessoryDisclosureIndicator`



Detail Button

`UITableViewCellAccessoryDetailButton`



Detail disclosure button

`UITableViewCellAccessoryDetailDisclosureButton`



Check mark

`UITableViewCellAccessoryCheckmark`

You can also add your own accessory control.

Table View Cells

- Storyboards allow you to design your table cells in the storyboard editor
- You can create different types of cells in one scene
- Dynamic prototype cells let you design one cell and use it as a template for other cells in the table

Static Cells

- An alternative to dynamic cells when you know what the table looks like at design time
- Enables you to design a table with a fixed number of rows
- These are static in the sense that they will exist every time the app is run
- The content of static cells can still change

Search Bar

- The **UISearchController** class incorporates a search bar, **UISearchBar**, into a view controller that has data
 - In a table you can assign the search bar to the **tableHeaderView** property
 - The **searchResultsUpdater** property is provided the search results from the search controller
- The **UISearchResultsUpdating** protocol must be adopted and its 1 required method handles the search bar interaction