How to Use this Template

1. Make a copy [File → Make a copy...]
2. Rename this file: “**Capstone_Stage1**”
3. Replace the text in green

Submission Instructions

1. After you’ve completed all the sections, download this document as a PDF [File → Download as PDF]
2. Create a new GitHub repo for the capstone. Name it “**Capstone Project**”
3. Add this document to your repo. Make sure it’s named “**Capstone_Stage1.pdf**”

GitHub Username: jubby2000

Proclaim: LDS Quote Guide

Description

Proclaim is a topical reference guide for quotes from LDS Church leaders.

Features:

- Browse for quotes based on the topic you're researching
- Search by author or keyword with custom suggestions
- Filter by most popular quotes in the community

- Upvote your favorite quotes
- Share your favorite quotes with other people and other apps (try Scripture Mastery to memorize!)

Proclaim is a curated selection, and the result of decades of work. Drop me a line if you find an error or a quote is misattributed.

Intended User

Members of the LDS faith (15m people)

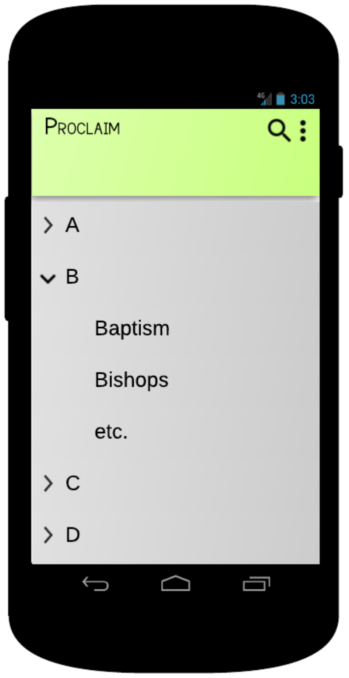
Features

- Displays quotes by topic (and possibly author?)
- Aggregates user preference, and sort by that preference (show me most favorited first)
- Share quotes to external sources
- Search for specific quotes by author or keyword

User Interface Mocks

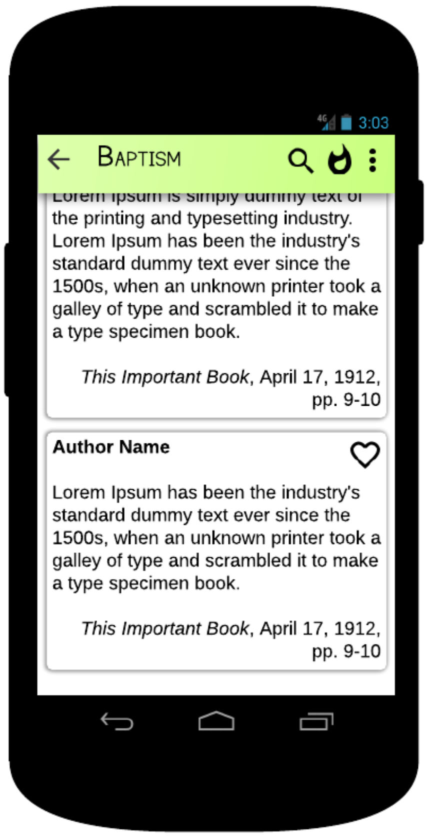
These can be created by hand (take a photo of your drawings and insert them in this flow), or using a program like Photoshop or Balsamiq.

Screen 1



On this screen, users will be able to select the topic that they're researching, broken down by letter (there are A LOT of topics). Tapping the search icon will allow the user to type in a query that will use the ContentProvider to do custom search against the database and return results existing within the app. In the overflow menu there might be an about page, settings to adjust the theme (for reading preference). Not shown here, I might include a slide out drawer for users to transition to an Author view instead of Topic. Also included there would be a Favorites option that will display all the quotes that users have clicked the Favorite button on. I'm unsure if I want to include those options in the overflow menu, or in a drawer. I'm leaning toward a drawer.

Screen 2



Lorem ipsum is simply dummy text of the printing and typesetting industry. Lorem Ipsum has been the industry's standard dummy text ever since the 1500s, when an unknown printer took a galley of type and scrambled it to make a type specimen book.

This Important Book, April 17, 1912, pp. 9-10

Author Name

Lorem Ipsum has been the industry's standard dummy text ever since the 1500s, when an unknown printer took a galley of type and scrambled it to make a type specimen book.

This Important Book, April 17, 1912, pp. 9-10

Upon tapping a topic, users are brought to this detail activity which is a collection of all the quotes that match that topic. The order is unsorted, and possibly randomized upon entry so that each quote gets a fair shot at exposure. Tapping the Favorite icon on an individual quote adds that quote to a user's collection of Favorites, and also increases that quote's popularity by one. Clicking it again reverses these actions. Clicking the Fire icon at the top sorts the quotes by most popular first, and displays a number next to the Heart, indicating how popular it is. Clicking the search icon allows the user to search within this category and filters out results that don't match. Finally, tapping on an individual quote copies it to the clipboard (with a Snackbar for success) or expands the quote (possibly into an individual activity) for much longer quotes. Action bar icons may be moved to a FAB, if doing so makes sense.

Key Considerations

How will your app handle data persistence?

I think I'm going to build a Content Provider. I'd like the practice. This will primarily be used for custom search within the app. Other data will be stored within a database and hosted on Google App Engine.

Describe any corner cases in the UX.

There doesn't seem to be corner-like operations within this app - it's pretty linear. The only thing I can think of is performing a search, and tapping on a result (Author or Topic). When the user wants to go back, do they intend to search again or do they want to return to the main activity. In this case I would defer to the Android guidelines. Have the up (or home) button return the user to the main activity, and have the back button return the user to perform another search.

Describe any libraries you'll be using and share your reasoning for including them.

There's a specific library that I've used before for an Expandable List View that looked pretty slick and provided the animations that I like. I may include pictures within the detail activity, but if so, I don't think I'd be pulling them from the internet. If I do I would use Glide to display them. I would also like to have the quote cards slide in from the bottom, which I might search for a library to perform for me.

Next Steps: Required Tasks

This is the section where you can take the main features of your app (declared above) and decompose them into tangible technical tasks that you can complete incrementally until you have a finished app.

Task 1: Project Setup

Create a database and a Content Provider for use within the app.

Database:
Should contain the following columns:
- Author First Name
- Author Last Name
- Quote
- Reference
- Date
- Page Number
- Popularity

- Favorite
- User Submitted
- Flagged

The database should work with Google App Engine to update specific columns like Popularity.

Content Provider:
The content provider should provide custom search within the app. If the user taps the search icon from the main activity and type “hin,” Hinckley should show up as a possible result, which is one of the authors. This should likewise work with matching topics.

Within the detail activity, tapping the search icon should filter through author and quote text only, and display only the matching quotes. Highlight the matching terms for an added bonus.

Task 2: Implement UI for Each Activity and Fragment

Create a framework starting with two activities: Main and Detail.

Main:
- Create a collapsing toolbar using CoordinatorLayout
- Create a fragment that uses an Expandable List View for the letters A-Z
    - Assign topics to each letter
- Add icons to search, and an overflow menu for additional items as needed
- For the search button, implement a Content Provider for custom search

Detail
- Create a detail activity that is launched with an Intent extra
- Also implement search here, but within the confines of the detail activity
- Enable a “popularity” button that will sort the results based on the result from the database, and an overflow button
- Create a fragment that will display a list of individual CardViews that match the Intent extra. The CardView should contain:
    - A “favorite” icon that will add that item to a list of other “favorited” items as well as increase a popularity counter by one, and do these things in reverse when clicked again.
    - Author name
    - Quote
    - Reference and date
    - The card should be clickable and should copy the quote to the clipboard when clicked, or expanded to show the truncated text if too long (then copied if clicked again)

Task 3: Materialize

Implement Material Design features:

- Hide toolbar on scroll
- Consider a FAB where appropriate
- Use an app theme, and consider adding alternate user selectable themes for the app or cards
- Use a Collapsing Action Bar
- Use Snackbars over Toasts

Task 4: Widget

Consider adding a widget containing a “Quote of the Day.” It doesn’t have to be a collection widget.

Add as many tasks as you need to complete your app.

Submission Instructions

1. After you’ve completed all the sections, download this document as a PDF [File → Download as PDF]
2. Create a new GitHub repo for the capstone. Name it “**Capstone Project**”
3. Add this document to your repo. Make sure it’s named “**Capstone_Stage1.pdf**”