Android Project Plan

Project Summary

In this project, an Android application will be developed to allow programmers to reference source code and frequently asked questions.

User Experience

This app will:

- Upon launch, present the user with a search box with voice input capability.
- Allow the user to filter results via a setting:
  - Comments and question feedback can be disabled in shared preferences.
- Allow the user to view reference material on a detailed results screen:
  - Question feedback
  - Share link button
  - Comments
  - Answers
  - Scores
  - Tags

Requirements

CRITERIA	DOESN'T MEET SPECIFICATIONS	MEETS SPECIFICATIONS
GENERAL PROJECT GUIDELINES		
App conforms to common standards found in the Android Core App Quality Guidelines.		
USER INTERFACE - LAYOUT		
A search input box is displayed centered in the main layout.		
UI contains an element (e.g., a spinner or settings menu) to toggle question feedback and comments in search results.		
UI contains a screen for displaying the search topics for a certain search query.		
UI contains a screen for displaying the search details for a selected search topic.		
Search Topics layout contains a search input on the ActionBar.		

CRITERIA	DOESN'T MEET SPECIFICATIONS	MEETS SPECIFICATIONS
Search Topics layout contains a view (e.g. ListView) of search topics displayed with rich text.		
Search Details layout contains a share button on the ActionBar.		
Search Details contains title, tags, score, question, answer, and comment content.		
Search Details layout contains a TextView of search results displayed in HTML.		
Tablet UI uses a Master-Detail layout implemented using fragments. The top fragment is for inputting search keywords. The left fragment is for discovering search topics. The fragment located in the center and to the right displays the search details view for the currently selected search topic.		
USER INTERFACE - FUNCTION		
When a user changes the feedback display (question and comments) the search details view gets updated correctly.		
When the search input is submitted on the main screen, the search details screen is launched [Phone] or displayed in a fragment [Tablet].		
When search input is changed in the ActionBar, content in the ListView of the topics screen is highlighted.		
When a search topic list item is selected, the search details screen is launched [Phone] or displayed in a fragment [Tablet].		
NETWORK API IMPLEMENTATION		
App requests for relevant search topics for a certain query via the /questions/{ids}/ endpoint in a background thread and displays those topics after the user submits a search query.		
App requests for relevant comment and answer content for a particular search topic via the /answers/{ids}/ and /comments/{ids} endpoints in a background thread and displays those topics when the user selects a search topic.		
DATA PERSISTENCE		
Apps loads programming source code references in HTML format from a database using default ids.		
App saves a “Display Feedback” option for questions and comments to SharedPreferences or a database using a default id.		

CRITERIA	DOESN'T MEET SPECIFICATIONS	MEETS SPECIFICATIONS
When the “display feedback” setting options are selected, the detail view displays the additional contents in the view area.		
CONTENT PROVIDER		
App persists display feedback settings using SharedPreferences.		
App persists recent search query results (topics and details) using a database.		
App displays source code queries (questions, answers, comments) even when offline.		
App uses a ContentProvider to populate search query details.		
SHARING FUNCTIONALITY		
Search Details View includes an Action Bar item that allows the user to share the source code URL generated from the query.		
App uses a share Intent to expose the external StackExchange URL for the search query.		

Implementation

This project will utilize a SQLite database in conjunction with the StackExchange API to parse JSON and deliver HTML content. The Android Studio IDE will be used to develop and test the Java source code.

All project related source files will be publicly committed to the GitHub repository listed below.

https://github.com/nshid/CodeFind

API

The StackExchange API v2.2 will be used to return questions, answers, and comments:

- Returns data in JSON format
- Requires API key from StackApps
  - Allows for access tokens (if necessary) and more requests per day
- API Query Return Type Examples
  - Questions
    - questions/{ids}
    - tags/{tags}/faq

        - answers/{ids}/questions
    - Answers
        - answers/{ids}
        - questions/{ids}/answers
    - Comments
        - comments/{ids}
        - answers/{ids}/comments
        - questions/{ids}/comments
- Stack Exchange API Documentation

DATABASE

The SQLite database will be used as an online cache and offline reference:

- Returns data in HTML format
- Online Cache
    - Stores recent API query results for quick offline/online lookup
- Offline Reference
    - Stores quick reference lookup tables for various programming languages
        - Java
        - C
        - C#
        - C++
        - Python
- SQLite Installation Tutorial

Additional Resources

Supplementary tools include:

- LG LS675
- Pencil Project
- Git Subversion
- Microsoft Paint
- Adobe Photoshop
- Android SDK Tools
- Reference Manuals
    - Java2s
    - CodePath
    - TutorialsPoint
    - Android Developers

UX Mockups

Phone UX

Tablet UX