GitHub Username: figengungor

The Dictionary

Description

Search words and expressions in English Dictionary with more than 350 000 entries powered by Oxford Api.

Search history will be recorded for easy access.

Home widget is provided with latest ten search history entries.

Intended User

This app is for native or non-native English speakers who like to improve their vocabulary knowledge.

Special Notes

- Java will be used for app implementation.
- The stable versions will be used for all libraries.(See the requested stable versions inside Describe libraries part)
- The search results don't have a detail screen. It will simply show definitions for each lexical category inside a CardView.
- App keeps all strings in a strings.xml file and enables RTL layout switching on all layouts.
- Content descriptions will be added to UI components for accesibility.
- AsyncTask will be used for some db operations.
- Android Studio 3.1, Gradle plugin 3.1.2, google-services plugin 4.0.0

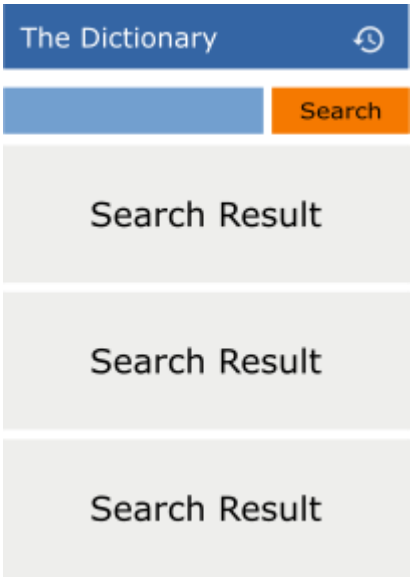
Features

- Search words and expressions
- View search history records
- Add widget for history records

User Interface Mocks

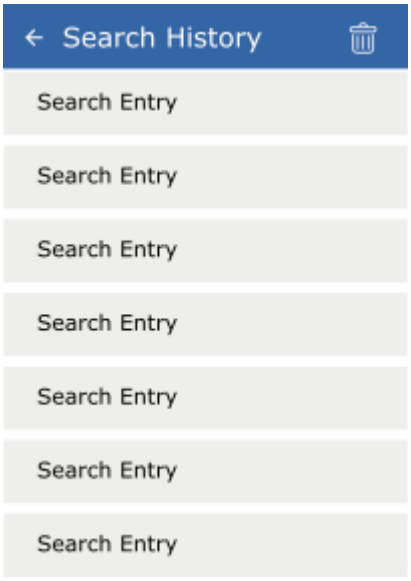
These can be created by hand (take a photo of your drawings and insert them in this flow), or using a program like Google Drawings, www.ninjamock.com, Paper by 53, Photoshop or Balsamiq.

Screen 1



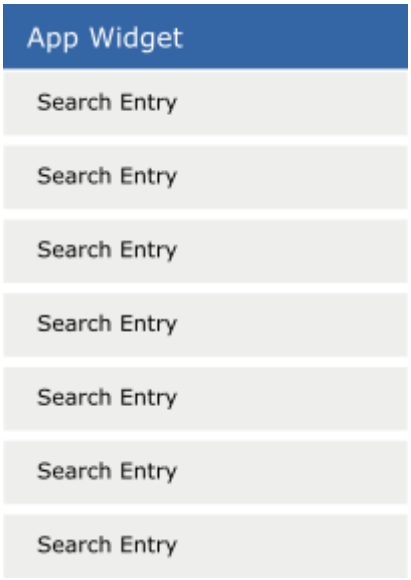
Home screen which provides search functionality. The search results dont have a detail screen. It will simply shows definitions for each lexical category inside a CardView.

Screen 2



Search History screen which displays history records. When search entry is clicked, HomeActivity will be opened with that entry search.

Screen 3



App Widget which displays last ten history records. When search entry is clicked, HomeActivity will be opened with that entry search.

Key Considerations

How will your app handle data persistence?

Search entries will be saved to history database when a search is successful. The history records can be viewed in Search History screen. The user can delete the history records one by one or all of them at once. Room persistance library will be used to save history records.

Describe any edge or corner cases in the UX.

No edge case.

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

Retrofit => Retrofit is a REST Client for Android and Java by Square. It makes it relatively easy to retrieve and upload JSON (or other structured data) via a REST based webservice.

Butterknife => Bind Android views and callbacks to fields and methods.

Android Architecture Components => (LiveData, ViewModel, Room) A collection of libraries that help you design robust, testable, and maintainable apps. Start with classes for managing your UI component lifecycle and handling data persistence.

Stetho => Easily view persistance data

RecyclerViewEnhanced => Swipe to delete UI and functionality

Stable Versions:

supportLibraryVersion = '27.1.1'
constraintLayoutVersion = '1.1.0'
picassoVersion = '2.5.2'
retrofitVersion = '2.3.0'
okHttp3Version = '3.9.1'
archComponentsVersion = '1.1.1'
roomVersion = '1.1.0'
lifecycleVersion = '1.1.1'
materialishProgressVersion = '1.7'
butterknifeVersion = '8.8.1'
stethoVersion = '1.5.0'
recyclerViewEnhanced = '1.1.0'
firebaseAdmobVersion = '15.0.1'
firebaseAnalyticsVersion = '16.0.0'

Describe how you will implement Google Play Services or other external services.

Firebase Ads => Display a banner ad in Home Activity
Firebase Analytics => Log events to see which features are used in the app

Next Steps: Required Tasks

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

Task 1: Project Setup

- Add library dependencies
- Add vector support

Task 2: Implement UI for Each Activity and Fragment

- Build UI for HomeActivity
- Build UI for HistoryActivity
- Build UI for app widget

Task 3: Implement Remote Data Retriving

- Create an app to use Oxford api and define app key and app id.
- Create models for Json data
- Create OxfordApi service
- Create DataManager and add methods to use OxfordApi service

Task 4: Implement Local Data Persistance

- Create Dao model
- Create Dao Interface
- Create TypeConverter for Date
- Create AppDatabase

Task 5: Implement ViewModels

- Create ViewModelFactory and ViewModel for HomeActivity
- Create ViewModelFactroy and ViewModel for HistoryActivity

Task 6: Create App widget functionality

- File -> New->Widget->App Widget
- Create RemoteViewsFactory
- Create RemoteViewsService
- Create AppWidgetProvider
- Add update code wherever a db change occurs

Task 7: Add SignIn Configuration

- Create a keystore
- Add SignIn Configuration

Task 8: Add Firebase ads

- Create a Firebase project and add google-services.json to app
- Add a banner ad to the bottom of Home Activity.

Task 8: Add Firebase Analytics

- Send log events to see which features are used in the app(search btn clicked, history is opened)

Submission Instructions

- After you've completed all the sections, download this document as a PDF [File → Download as PDF]
    - Make sure the PDF is named "**Capstone_Stage1.pdf**"
- Submit the PDF as a zip or in a GitHub project repo using the project submission portal

If using GitHub:

- Create a new GitHub repo for the capstone. Name it "**Capstone Project**"
- Add this document to your repo. Make sure it's named "**Capstone_Stage1.pdf**"