# OLYMPIA: A dictionary with categories, filtering and self-check

Project Olympia is an Android dictionary app that uses public APIs of popular English dictionaries and has one distinct feature: it allows to add labels to words, putting them (words) into categories. Categories are a user defined text and are created to their own needs or interests. The application is intended to help learners of English language to memorize new words. Words can naturally belong to multiple categories which can be altered at any time. Application can aggregate multiple dictionaries.

Example. User can add category “sounds” and when he encounters in a book or on TV words “thud” or “squeak” he will place them to “sounds” category, while learning what is “contemptuous” or “condescension” he will mark them as belonging to “emotions” category. With time this personal vocabulary will grow and the user will be able to review words by categories and learn words better.

# APIs:

Oxford <https://developer.oxforddictionaries.com/documentation>

# Use cases (functionality)

**Get word definition** – allows a user to input a new word in the input field and get its definition via web API. Opens in a new activity as a Word Card.

**Words List** – entered word is placed in a list of words looked up by the user. List of words is shown on the main page.

**Extended Words List** – allows to see not only the word in the list, but its short definition (limited number of characters).

**Go to word definition** – opens a Word Card when clicked on a word in the Words List.

**Word look up upon typing** – allows a user to see suggested words as he types it. Supported by some APIs.

**CRUD operations for user** – even though a mobile device is normally used by only one person, accounts/profiles must be supported as API keys will be different for different users.

**CRUD operations for category** – within the application Category is a unique id with visible text name that can be applied to words and manipulated – created, read, renamed and deleted.

**Dictionary of words** – for every API there is a key-value pair between the word and its JSON object received from the API.

**Add/remove category to a word** – on a Word Card the user can add and remove categories into which a word falls.

**Clear (forget) the word** – user may want to remove a defined word from his vocabulary, so it’s removed from the DB and the Words List.

**Words List pagination** – the list of words may be long and not usable, so pages for 50-100 words should be introduced.

**DB support** – the list of words must be saved to disk/database on every termination and restored on start up. As terms of use of most APIs forbid saving JSON data received from the server, the app will only store original words (input by user) and their categories (if any).

**Switch dictionary (API)** – the app supports multiple APIs and must have a possibility to switch between them.

**Change API key** – keys may expire and may need to be updated.

**Export database to file** – in event user wants to make a backup or use the dictionary on another device the app should be able to form an SQL dump and save it as a file.

**Import database from file** – reversed operation to set current database.

**Self-check** – user picks a category(s) and the app tests knowledge of memorized words. Given a definition the user must write word correctly to get highest score.

**Search for similar words** – allows the user to search for words in his vocabulary that differ by 1, 2, 3 letters.

**Sort a Words List** – allows to sort words alphabetically, by date when the word was added etc.

**Filter** **a Words List** – shows only words satisfying some criteria. Can be regex, category, starting letter etc.

**Statistics** – page with various parameters like number of words per day, score in quizzes etc.