

[Description](#)

[Intended User](#)

[Features](#)

[User Interface Mocks](#)

[Screen 1](#)

[Screen 2](#)

[Key Considerations](#)

[How will your app handle data persistence?](#)

[Describe any corner cases in the UX.](#)

[Describe any libraries you'll be using and share your reasoning for including them.](#)

[Describe how you will implement Google Play Services.](#)

[Next Steps: Required Tasks](#)

[Task 1: Project Setup](#)

[Task 2: Implement UI for Each Activity and Fragment](#)

[Task 3: JSON Parsing for the API](#)

[Task 4: Create SQLite Database](#)

[Task 5: Implementation for Speech to Text](#)

[Task 6: Shared preferences for different language support](#)

[Task 7: Implementation for Widget](#)

GitHub Username: [Ishmeetsingh97](#)

Word Clue

Description

Word Clue is an online dictionary application, which fetches data from Yandex Dictionary API and has different language support for users.

Intended User

Word Clue is intended for everyone who want to use a user friendly dictionary and especially for those people who want their meanings to be in different languages other than English such as French, German, Spanish and Russian.

Features

The main features of the app are:

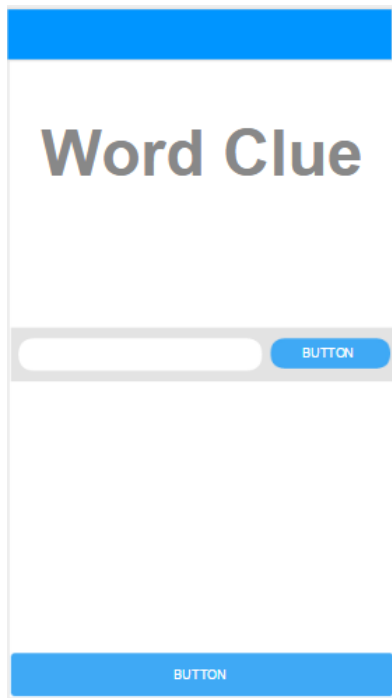
- Displays the meaning of a particular word.
- Can translate the meaning into different languages.
- Stores the search history in the database with content provider.

- Can take input from speech (Speech to text Implementation).

User Interface Mocks

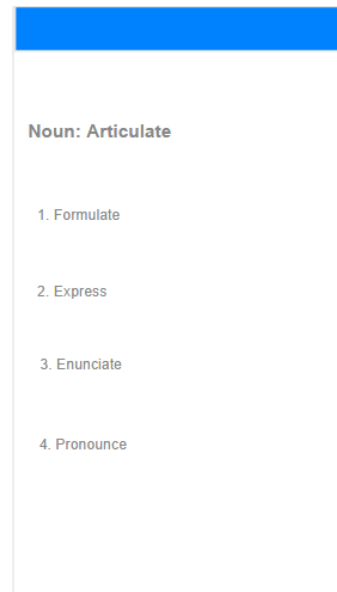
These are some basic Mocks for Word Clue.

Screen 1



MainActivity for Word Clue. Helps to search the meaning of a particular word.

Screen 2



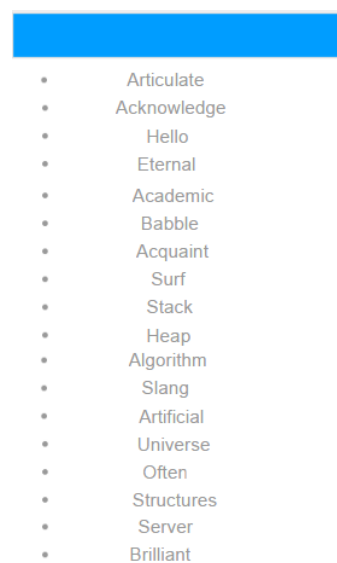
The screenshot shows a blue header bar at the top. Below it, the text "Noun: Articulate" is displayed. A list of four numbered items follows: "1. Formulate", "2. Express", "3. Enunciate", and "4. Pronounce".

Noun: Articulate

1. Formulate
2. Express
3. Enunciate
4. Pronounce

This is the Meaning activity, which shows the meaning in English.

Screen 3

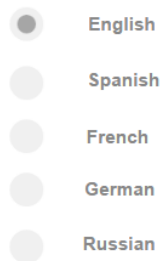
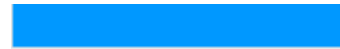


The screenshot shows a blue header bar at the top. Below it, a list of 20 words is displayed, each preceded by a bullet point. The words are: Articulate, Acknowledge, Hello, Eternal, Academic, Babble, Acquaint, Surf, Stack, Heap, Algorithm, Slang, Artificial, Universe, Often, Structures, Server, and Brilliant.

- Articulate
- Acknowledge
- Hello
- Eternal
- Academic
- Babble
- Acquaint
- Surf
- Stack
- Heap
- Algorithm
- Slang
- Artificial
- Universe
- Often
- Structures
- Server
- Brilliant

This is the History activity, which shows the history of searched words.

Screen 4



This is the Shared Preference menu, which shows the different languages available.

Key Considerations

How will your app handle data persistence?

Storing the searched words in a content provider and displaying the searched history in a list view in the history activity.

Describe any corner cases in the UX.

N/A

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

- EasyFont for different fonts.
- Some library to make UI better. (Not decided yet)

Describe how you will implement Google Play Services.

- Google AdMob.
- Google+ Sign in.

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

- Configure the Yandex Dictionary API, receive the API key and implement search bar for sending queries to the API.
- Implementation for EasyFont library for Fonts.

Task 2: Implement UI for Each Activity and Fragment

- Build UI for MainActivity.
- Build UI for History Activity which displays a list view of searched words.
- Build UI for Meaning Activity which displays the meaning of the searched word.

Task 3: JSON Parsing for the API

- JSON Parsing for the Yandex dictionary API.
- Received the meaning/acronym for the searched word.

Task 4: Create SQLite Database

- Implementation for SQLite database to save history with content providers.
- Implementation to display the content provider activity in a list view.

Task 5: Implementation for Speech to text

- Implementation for Recognizer Intent.
- Implementation to receive the response of recognizer intent in the search bar.

Task 6: Shared Preferences for language support

- Modifying the API link for different languages support using Shared Preferences.
- Able to receive the meaning in 5 Different languages.

Task 7: Implementation for Widget

- Implementation for the widget which displays the different languages supported by Word Clue in an interactive way on the home screen.