

Assignment 4 – Wordy

Deadline: Thursday, November 14, 2023, 11:59 pm

Objectives:

- ✓ Familiarize yourself with integrating a cloud database in your Android app

Requirements:

Version – Your app should work on Android 13.0 or higher (API 33)

Task:

Create a wordle clone app named Wordy. Wordle is a game in which a five-letter word is chosen and the player's goal is to guess the chosen word within six tries. After every guess, each letter is marked as either green, yellow or gray: green indicates that letter is correct and in the correct position, yellow means it is in the answer but not in the right position, while gray indicates it is not in the answer at all. Multiple instances of the same letter in a guess, such as the "o"s in "robot", will be colored green or yellow only if the letter also appears multiple times in the answer; otherwise, excess repeating letters will be colored gray.

– Modified From Wikipedia

The app will have two main features. The user should have access to one activity to “play wordy” and a second activity to “Add a new word to the wordy word bank”.

- ✓ Word bank [6 points]
 - Contains all the words used by the app (Must be stored in Firebase)
 - A word is a simple 5-letter string (e.g., “cream”, “Greed”, “color”, “abcde”)
- ✓ Wordy Activity [16 points]

This activity should have:

 - Six rows for user input [5 points]
 - An “Add word” button that switches to the “create word” activity [5 points]
 - A way to retrieve a random word from the word bank (the target word) [5 points]
 - Questions are drawn from the Firebase database [20 points]
 - A “Submit” button that will check if the provided word matches the target and lock the input row [5 points]
 - A “Restart” button that will restart the game with a new random word each time it is tapped [5 points]
 - A “Clear” button. When tapped the current wordy game will clear out all the previous guesses allowing you to redo the current game [5 points]
 - Notify the user of a victory or loss via the UI [10 points]
- ✓ Create Word Activity [44 points]

A second activity will allow the user to enter a word.

 - Have a cancel button that switches to the “Ask Question” activity [4 points]
 - When the Add button is pressed, the question will be added to the word bank on Firebase if it meets the restrictions outlined below [4 points]:

- It is not empty **[1 point]**
- It is exactly 5 characters long **[1 point]**
- It contains only alphabetical characters (letters only) **[1 point]**
- The same word is not already in the wordy word bank (case insensitive) **[2 points]**
E.g., “dog” and “DoG” are considered the same word
- If all the checks pass, display a Toast indicating that the information was stored in the database **[1 point]**
- If one or more errors occur:
 - Change the color of the question label to purple **[1 point]**
 - Notify the user by way of a Toast about the errors that occurred **[1 point]**

Extra credit:

- Add a Hard Mode in which each subsequent guess has to have all the letters that were guessed correctly in previous guesses **[15 points]**
- Add a “Clear Database” button in the add word activity. When tapped, all the word in the *Word bank* should be deleted **[10 points]**
- Come up with your own extra credit feature this will be judged to be determined, if it merits extra credit **[up to 1 question] [up to 20 points]**

Grading Rubric:

Item	Points
Word bank using Firebase	6
Wordy Activity <ol style="list-style-type: none"> 1. Button to ask a question [5 points] 2. Button to clear current question [5 points] 3. Button to go to the add question activity [5 points] 4. Random question selected from Firebase’s question bank and shown in the app [25 points] 	40
Add Word Activity <ol style="list-style-type: none"> 5. Add Button [4 points] 6. Cancel Button [4 points] 7. Input field [2 points] 8. Error validation [14 points] 9. Toast messages [6 points] 10. Word is added to Firebase successfully [14 points] 	44
Code Quality <ul style="list-style-type: none"> ▪ Code comments, formatting, and meaningful variable naming 	10
Total Points	100

Deliverables:

On GitHub:

- A GitHub repository with the contents of your Android Studio project. **Make sure you double-check that the repository is public.**

On Canvas:

- A zip file containing the Android Studio project with the implementation of the app. **Make sure you double-check that the file was uploaded correctly on Canvas!** I will not be accepting excuses after the deadline that the file was not uploaded correctly.
- A link to the GitHub repository with your code.