

ITE 2152 - Introduction to Mobile Application Development

Assignment #02 MAD Word Guessing Game

Project Overview:

In this assignment, you have to develop a mobile application, which is a word guessing game. The application should pick a random word from an online API such as [1] and [2] at the start of the game, however, keep it a secret from a user. The challenge is to guess the word correctly, fast, and in a minimum number of attempts. User will get 100 points at the beginning, with 10 attempts of guessing the secret word. Each of the wrong guesses and obtaining clues will have a penalty as described later-on. User may also choose to get a tip/clue if the player fails to guess correctly during the first five attempts.

Player can take maximum of 10 guesses. If its not successful, a new word and a fresh score of 100 points will be given to re-start the game.

Upon a successful guess, the player will go to a higher level with more complicated, longer words.

The following features should be implemented.

Features:

1. **Onboard:** When the user is installing the app, the app should ask the name of the user. Then the name should be remembered whenever the user is reusing the app. Suggestion: You may use a shared preference for this. [5 marks]
2. **Guess the word:** The user can type a word and submit, so that the app will tell whether the guess was correct or not. 100 points is allocated at the beginning. For each incorrect guess, 10 marks will be deducted. If the marks reach 0, the user is considered failed the attempt and a new word will be given. [25 marks]
3. **Occurrence of a letter:** The user can pick a letter at a cost of 5 points, so that the app will tell back how many instances of that letter is present in the secret word. [5 marks]
4. **How many letters in the secret word?** User should be able to ask how many letters there are in the secret word, at a cost of 5 points. [5 marks]
5. **Give a tip about the word:** for example, give a one rhyming word [3] or a similar word [4]. You may give only a one tip after the fifth attempted guess. [5 marks]

6. **Time to guess?** Should have a timer to record how long it takes to do the correct guess.

[5 marks]

7. **Leaderboard:** You can maintain a global leaderboard of the scores and times via an API service like [5]. You can test it with multiple instances of the app, or team up with few of your colleagues to share a same secret to have a dynamic and active leaderboard. You might need to handle the ssl issues as described in [6].

[10 marks]

Other aspects to obtain marks:

1. **User Interface(s) Design:** Design of user interface for the MAD Word Guessing Game application. Consider usability and intuitive navigation and careful choice of number and types of layouts, widgets etc. For example you can decide on how to design the interface so that its intuitive to know how a user should ask for a letter vs guess the word in the same graphical user interface without a confusion. Completeness of features, processes and interactions should also be considered. [20 marks]
2. **Explanation of code, quality, problems faced and design choices** [5 marks]
3. **Error Handling, Testing and Debugging:** Test the application thoroughly to identify and fix any bugs or issues. Ensure that all features work as intended across different devices and screen sizes and should be demonstrated. [5 marks]
4. **Presentation:** Prepare a presentation as a user guide to demonstrate the functionality of the application. Present the user interface design, key features, implementation and testing/debugging details and any unique aspects of the application.
5. **Demo:** Prepare a demonstration and record a screen capture while explaining the process in a video containing all the application's features, functionalities, and implementation details.
6. **Prepare a short video** covering your logic and coding, layout and design decisions, by combining your presentation, demonstration and a code review. You must cover your effort on each of the other allocated marks in the demo presentation to receive the allocated marks for each section (ideally in that order), since the evaluators may only just follow your demo video during the marking process.

[10 marks]

Summary of Evaluation Criteria:

- Functionality and Features Implementation – **60 marks**
- User Interface Design (UI/UX) – **20 marks**
- Code Quality, app design choices and organization – **5 marks**
- Error Handling and Testing – **5 marks**
- Demonstration Quality and Communication Skills – **10 marks** (And other marks are also at stake if you don't present it properly!)

Students are required to submit:

- The zip file with the mobile app and the simple backend along with the source code and project files.
- Recorded demonstration video of the app with narration (unlisted YouTube/ google drive link), covering the evaluation criteria aspects mentioned above. If the demo video is not submitted or does not contain the sufficient details according to the abovementioned evaluation criteria, the assessment of your submission will be capped at maximum of 60%.

References

- [1] <https://api.api-ninjas.com/v1/randomword>
- [2] <https://random-word-api.herokuapp.com/word>
- [3] <https://api-ninjas.com/api/rhyme>
- [4] <https://api-ninjas.com/api/thesaurus>
- [5] <https://dreamlo.com/>
- [6] <https://julien-bouffard.medium.com/build-an-android-app-using-a-non-secure-api-in-android-9-88c9bafbc12>