

Cryptogame (Android Application)

Design a [cryptogram](#) game as an android application which uses a [simple substitution cipher](#). The detailed requirements for the application are provided below.

Requirements:

1. A user will be able to choose to log in as a specific player or log in as the administrator when starting the application. For simplicity, any authentication is optional, and you may assume there is a single system running the application.
2. The application will allow players to (1) choose a cryptogram to solve, (2) solve cryptograms, and (3) view the list of player statistics.
3. The application will allow the administrator to (1) create a cryptogram, (2) create a player, and (3) view the list of player statistics.
4. A cryptogram will have a solution (the plaintext phrase) and a maximum number of allowed solution attempts for each of three difficulty categories.
5. To add a player, the administrator will enter the following player information:
 - a) A first name
 - b) A last name
 - c) A unique username
 - d) A difficulty category: easy, normal or hard.
6. To add a new cryptogram, the administrator will:
 - a) Enter a unique cryptogram name.
 - b) Enter a solution (unencoded) phrase.
 - c) Enter the number of allowed incorrect solution attempts for the easy difficulty.
 - d) Enter the number of allowed incorrect solution attempts for the normal difficulty.
 - e) Enter the number of allowed incorrect solution attempts for the hard difficulty.
 - f) Edit any of the above steps as necessary.
 - g) Save the complete cryptogram.
 - h) View a confirmation that the name assigned to the cryptogram is unique and return to the main menu, or be returned to editing the cryptogram after any error is displayed.
7. The encrypted phrase for the cryptogram will be generated for each player starting a new cryptogram by:
 - a) Replacing each letter with another letter randomly, so that all of any particular letter are replaced with the same other letter, such as all A's becoming C's, and

every letter is paired with a unique encrypted letter.

- b) Preserving the capitalization in the original phrase.
- c) Preserving any non-alphabetic characters (such as punctuation or white space) unaltered.

8. To choose and then solve a cryptogram, a player will:

a) View the list of all unsolved cryptograms alongside their status as in progress or unstated , and choose a cryptogram to solve.

b) View the chosen cryptogram and number of incorrect solution attempts remaining (starting at whatever number is allowed for the player's difficulty level for that cryptogram). If the cryptogram has not been played by this player before, the fully encrypted phrase should be generated and displayed. If the cryptogram is in progress, the previous state of the phrase should be displayed.

c) Match the replacement and encrypted letters together, and view the resulting potential solution.

d) When all letters in the cryptogram are replaced and they are satisfied with the potential solution, submit their answer.

e) Get a result indicating that the solution was successful, or decrementing the number of incorrect solution attempts remaining if it was unsuccessful.

f) At any point, the player may return to the list of unsolved cryptograms to try another.

g) If the number of incorrect solution attempts reaches zero, they will get a result that the cryptogram game was lost, and this cryptogram will be marked as complete, unavailable for this player to attempt again. They will then return to the menu.

h) If the player successfully solves the cryptogram, they will get a result that the cryptogram game was won, and this cryptogram will be marked as complete, unavailable for this player to attempt again. They will then return to the menu.

9. The list of player statistics will display a list of players in descending order of number of cryptograms won. The entry for each player will show their first name, the number of

cryptograms won, and the number of cryptograms lost. An administrator should also see the username and difficulty status of the player.

10. The user interface must be intuitive and responsive.

11. The performance of the game should be such that students do not experience any considerable lag between their actions and the response of the application.