**Semester Project Documentation**

**Semester Project Title**: Hotel management system

**Student Details**

(Fill according to number of team/group members)

|  |  |  |  |
| --- | --- | --- | --- |
|  | Student Name | Student Reg # | Student Degree |
| Student-1 | Anas Ahmed | 2023114 | Engineering Sciences |
| Student-2 | Hasnain Zaman | 2023235 | Engineering Sciences |
| Student-3 | Awais Akhtar | 2023138 | Engineering Sciences |
| Student-4 | Fasih Jan | 2023201 | Engineering Sciences |

**Main Features**

**Word Selection:** This word serves as the puzzle that needs to be guessed.

**Display of Hidden Word:** The hidden word is displayed as a series of blanks or underscores, indicating the number of letters in the word and their positions. Players guess individual letters to reveal parts of the hidden word.

**Guessing Mechanism:** Players attempt to guess the hidden word by suggesting letters that might be a part of it. The game typically limits the number of incorrect guesses allowed before the game ends. Correctly guessed letters are revealed in the hidden word, while incorrect guesses lead to penalties, such as losing a chance or making progress toward a loss.

**Feedback on Guesses:** After each guess, the game provides feedback to the player regarding the correctness of their guess.

**Win/Lose Conditions:** The game continues until the player either successfully guesses the entire word or makes a certain number of incorrect guesses (reaches the limit of allowed attempts).

**Types of Users**

**The Player (Guesser):**The player is the person trying to guess the hidden word. They provide letter guesses to uncover the word within a limited number of attempts.

**The Host :**The host is responsible for selecting a word, phrase, or term to be guessed.This person knows the word and keeps it hidden from the player.

**Requirements Breakdown**

**Word Bank:**Create a collection of words or phrases that the player will attempt to guess. Store these words in a data structure (e.g., an array, vector, or file).like we did have defined the words using this piece of code “vector<string> words = {"hangman", "computer", "programming", "hello", "world", "gaming", "computer"};

**Word Selection:**Implement a mechanism to choose a random word from the word bank for each game session. You can use random number generation combined with the size of the word bank.

**Display Hidden Word:**Represent the hidden word with underscores or blanks to indicate the letters yet to be guessed. Update this display as the player makes correct guesses.string hiddenWord(word.length(), '\_');

**Guessing Mechanism:**Develop a method for the player to input their letter guesses. Validate the input and check if the guessed letter is present in the selected word.We used the function “**ISLETTERPRESENT”** and defined it to check if the users input is present in the word to be guessed.

**Tracking Attempts:**Set a limit for the number of incorrect guesses allowed.Update the attempts count and signal the end of the game when this limit is reached.”--attempts;” is used to decrement the attempts if the user guesses a wrong character.

**Win/Lose Conditions:** Establish conditions to determine when the player wins or loses. Winning occurs when the entire word is guessed correctly within the attempt limit. Losing happens when the allowed attempts are finished without guessing the word.

**Outcome Display:** Upon game completion, display appropriate messages indicating whether the player won by guessing the word or lost due to exceeding the attempt limit.cout << "Congratulations! You guessed the word: " << word << endl;

    cout << "Oops! You ran out of attempts. The word was: " << word << endl;

**Features to Codding Matrix**

*(In the following table you will mention the following items for each feature, mention the items in each column for each feature of your application)*

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Sr  no. | Feature Name | Concept Used | Functions Created | Variables / Obj Created | Line of Code Written |
| 1 | Word Bank | Strings | No Function | vector<string> words | 41 |
| 2 | RandomWordSelection | Randomization | rand()  time(nullptr) | srand, word | 42-44 |
| 3 | Game Looping | Loop Structure | While Loop | No Variables | 53 |
| 4 | Guessing Mechanism | IF ELSE ControlStructure | isLetterPresent,  updateHiddenWord | isLetterPresent,  updateHiddenWord | 15 & 28 |
| 5 | Feedbacks | Conditional Statements | isLetterPresent | No Variables | 60 |
| 6 | Win/Lose | Conditional Statements | isGameOver | isGameOver  attempts | 53 |
| 7 | User Interaction | IO Statements | Cout and Cin | No Variables |  |
| 8 | Display End Result | Output Message | Cout | cout | 64 & 77 |
| 9 | User Interface | Terminal Interaction | Terminal Input Output | Terminal Input Output |  |