|  |  |  |
| --- | --- | --- |
| Use case ID |  |  |
| Use case Name |  | Lettery game |
| Created by &  Date |  | Taha Aflouk  20/12/20 |
| Description |  | This use case allowed the player or user to play a lettery game that user starts with a hundred€ then user must choose 5 different letters from the English alphabet from A-Z  When user finish picking 5 different letters the system will get 5 different random letters and they have to be removed to do not let the system pick them again , then the system will compare the user picked letters with the random letters that created by the game and show how many matches and the system must calculate how much the user wins and show that to the user then the system has to add that to his money and ask the user if wants to go again |
| Primary Actors |  | System |
| Secondary Actor |  | User |
| Precondition |  | Must have the files of the game and import what we need like handy.py to use some functions from that file ,file is running , Cash valid, user must enter 5 different letters uppercase only English alphabet system does not take the letter twice and no numbers or symbols or empty input so it should be valid input after that, the game has to create 5 different random letter and they have to be removed from the game, the program creates only five letters uppercase it has to be a letter so same length and both are alphabet to compare them |
| Post condition |  | The system let the user play one or more than lotto game |

|  |  |  |
| --- | --- | --- |
| Use case ID |  |  |
| Use case Name |  | Gen\_drum |
| Created by &  Date |  |  |
| Description |  | A function that creates a list of alphabets from A-Z and return it and that function has only one argument with default value = 5 |
| Primary Actors |  | System |
| Secondary Actor |  |  |
| Precondition |  | I did have to import string from the library to make that list |
| Post condition |  | List of the English alphabets created |

|  |  |  |
| --- | --- | --- |
| Use case ID |  |  |
| Use case Name |  | Get\_ticket |
| Created by &  Date |  |  |
| Description |  | A function that gets 5 different letters from the user and return that as new list it takes only one argument with default value = 5 |
| Primary Actors |  | User |
| Secondary Actor |  | System |
| Precondition |  | I did have to import string from the library to check that if it is from the alphabet and not more than one character, and it should not be picked already |
| Post condition |  | A list of 5 alphabets created by the user |

|  |  |  |
| --- | --- | --- |
| Use case ID |  |  |
| Use case Name |  | Do\_draw |
| Created by &  Date |  |  |
| Description |  | A function that creates a list of 5 alphabets from A-Z randomly it takes two arguments one for the whole alphabets which is list and the second argument is a constant variable with default value = 5 |
| Primary Actors |  | System |
| Secondary Actor |  |  |
| Precondition |  | We must create a function of all the letters first then we must get randomly one letter each time and the random letter should be removed from the list so that the system will not get the same letter in the next go |
| Post condition |  | List of the English alphabets created randomly by the system |

|  |  |  |
| --- | --- | --- |
| Use case ID |  |  |
| Use case Name |  | Check\_result |
| Created by &  Date |  | Taha |
| Description |  | A function that compare the list of the user with the random list that created by the system and checks if there any match between the two lists and it calculate how many matches between the two lists it has two arguments and the both (lists type) of letters |
| Primary Actors |  | System |
| Secondary Actor |  |  |
| Precondition |  | We must have the gen\_drum function and get\_ticket function and do\_draw function they have to be lists of letters letters and same length to compare them |
| Post condition |  | A number of the matches elements between the two lists |

|  |  |  |
| --- | --- | --- |
| Use case ID |  |  |
| Use case Name |  | Dramatic\_print |
| Created by &  Date |  | sean |
| Description |  | A function that creates a dramatics dots to show the user that the system is loading and comparing the two lists together it has two arguments one is string type( statement ) and another argument with default value = 6 |
| Primary Actors |  | System |
| Secondary Actor |  |  |
| Precondition |  | We did have to import time file to be able to use sleep method |
| Post condition |  | A dramatic dots created and print it |

|  |  |  |
| --- | --- | --- |
| Use case ID |  |  |
| Use case Name |  | Game |
| Created by &  Date |  | Taha |
| Description |  | A function has no arguments that generate the drum based on the main gen\_drum function and get ticket from the user then print the ticket and the random balls and then it checks the two lists and it prints the user ticket and the random balls and prints the ticket, print the dramatic function, it checks if any matches between them, print the random list, print the matches, calculates the money and add them to the cash |
| Primary Actors |  | System |
| Secondary Actor |  | User |
| Precondition |  | We must have done the gen\_drum function to get random chars from it and we must have get\_ticket function that allows the user to enter 5 letters and do\_draw function to get 5 random letters and we must import handy.py and import math and import random import string , import doctest and import time we must make check\_results function to check the two lists and we must have the dramatic function done and in handy.py we must have title\_gen and menu\_gen functions done |
| Post condition |  | User plays the game |

|  |  |  |
| --- | --- | --- |
| Use case ID |  |  |
| Use case Name |  | Main function |
| Created by &  Date |  | Sean |
| Description |  | A function has no arguments it generate the title and the cash or a the menu in general title, cash, goodbye, |
| Primary Actors |  | System |
| Secondary Actor |  |  |
| Precondition |  | We must have done the menu\_gen done in handy.py before we wright the function because we generate the title and all the menu in that function |
| Post condition |  | Design a menu for the game |

|  |  |  |
| --- | --- | --- |
| Use case ID |  |  |
| Use case Name |  | Menu\_gen |
| Created by &  Date |  | Taha |
| Description |  | A function in handy file to generate a the menu items with the number of the has one argument list of str user will pick a valid number from the list to enter or quit |
| Primary Actors |  | System |
| Secondary Actor |  | user |
| Precondition |  | We must have the list of the item to make them as a menu |
| Post condition |  | User entered the game |

|  |  |  |
| --- | --- | --- |
| Use case ID |  |  |
| Use case Name |  | Title\_gen |
| Created by &  Date |  | Taha |
| Description |  | A function that has three arguments two of them have default values one for a symbol and another argument the type in to generate or design a title |
| Primary Actors |  | System |
| Secondary Actor |  |  |
| Precondition |  |  |
| Post condition |  | Title is created |

Test case

Use case table for get\_ticket function

|  |  |  |
| --- | --- | --- |
| Application |  | Lettery game |
| Actor involved |  | sassad  User, System |
| Description |  | A lettery game that user starts with 100 euro but with the English Alphabets that user must pick 5 different  Letters from A to Z then the system will print the letters and get 5 different random letters and print that as well and calculate the matches letters between the user picks and the 5 random letters then add the winning money to the cash if there was any based on how many matches if 3 or more matches letter win half odds less than 3 it cost user euro for one go |
| Preconditions |  | 1. User on desktop 2. User has the files inside a folder on a desktop 3. User has handy.py 4. User has lettery.py |
| Post-conditions |  | User must have some handy.py and lettery.py in the same folder  and some imports files like handy, random, string…  User must have cash to start the game  User must press 1 to start the game  User must pick 5 different letters from A to Z  User must press Enter to continue  The first round in the game is done and system print the wins ask if user wants to play or exit |
| Triger |  | User play a lotto game with letters user has to pick 5 different letters and the game generate 5 different random letters wins depending on how many matches user gets between user ticket and the random letters |
| Basic flow |  | User has to start with 100 euro in the cash   1. User runs lettery.py 2. System prints title and cash 3. System asks user to enter a valid number from the menu 4. User enters 1 from the menu 5. System asks user to enter a letter 6. User enters English letters from A to Z (5 times) one letter each 7. System prints the letter user has entered 8. System prints dramatics that the balls are drawing 9. System print 5 different letters randomly 10. System prints odds 11. System prints matches 12. System calculates matches and print how much won 13. System adds money to the cash 14. System asks user to enter (Enter) to continue 15. User enter to continue 16. System will go back to the starts with updates to the cash |
| Alternative flow |  | 3.a if user enters a number that is not from the menu the system will ask to enter a valid number from the menu like a fractions number  5.a if user enters more than one letter the system will show a message to enter a single letter and ask again to enter a single letter  5.a if user enter a number or symbols the system will show a message enter a letter and go again  5.a if user enter a letter more than a time the system will show a message the letter is already entered  14.a if user did not hit enter the game does not continue |

|  |  |
| --- | --- |
| Exception flow | If user does not have cash the system cannot run  3.a if user enters something else not number the game will stop working |