# REQUIREMENTS DOCUMENT

|  |  |
| --- | --- |
| **Date Submitted:** | March 29, 2021 |
| **Application Title:** | Product List |
| **Purpose:** | This program is designed to allow the user to create a list of products with corresponding numbers and save that information  to a text file. The user can change the product’s name, add or delete products, search for specific products in the directory by number, and display the full directory of products. The data will save every time the program ends. |
| **Program Procedures:** | 1. Once the program runs the user will be prompted with a menu. 2. The user will be asked to select type 1, 2, 3, 4, 5. Each number corresponds with an option in the menu. 3. 1 will display the directory, 2 will allow you to look up emails, 3 will allow you to add a new email, 4 will allow you the change the current emails, and 5 will delete any of the entries. 4. Once you decide and follow the prompts given, you will be asked whether you want to continue or not. If you choose “yes” the menu will redisplay for you to make another decision and if you choose “no” the program will close. 5. All the data will be saved in a file called productList.txt. This same file will allow the program to remember the change you made to the directory for the next time you run the program. 6. Do not delete this program or you will lose your data. |
| **Algorithms, Processing, and Conditions:** | 1. Import the pickle attachment 2. Define the main function 3. Declare a Boolean variable for the while loop to be called 4. Create an empty dictionary 5. Open the file with the dictionary of products that was previously saved, use a try and except method to check for a previous file with the content required. 6. Use the pickle to load the information into the program. 7. Close the file 8. Use a while loop to execute a continue feature 9. Call the displayMenu function. 10. Prompt the user to enter a command that corresponds with the menu 11. Use an if statement to call the corresponding function. 12. Asking the user if they would like to continue after the function has finished. 13. Using an if statement to et the boolen variable to continue or not. 14. Pickle the changed information into the file to be saved. 15. Define the displayMenu function 16. Use print statements to define the menu 17. Define the displayProducts function. 18. Loop through all the keys in the dictionary and print each key and value to the console. 19. Define the lookup function. 20. Ask the user the number of the product to find. 21. Use an if statement to check if that number is in the directory. 22. Print the product in question. 23. Define the addNewProduct function. 24. Ask the user for the number and name of the new product. 25. Add the product to the directory and display the new directory. 26. Print out a confirmation. 27. Define the changeProduct function. 28. Ask the user for the number of the current product. 29. Check if the number is in the directory. 30. Tell the user the number was found and ask them for the new product name. 31. Save the new name to the number. 32. Define the deleteEntry function. 33. Ask the user for the number of the product they want to delete. 34. Check if the product is in the directory. 35. Delete the product. 36. Print a confirmation for the user. 37. Call the main function. |
| **Notes and Restrictions:** | You must use the productList.txt file to change the information. |
| **Comments:** | The information should always be saved through the productList.txt file. DO NOT delete this file. |