**LizCoach Project**

Muslima Rizayeva and Kamila Otabekova

Computer Languages, Webster University in Tashkent

COSC 2110 3T FA 2024

Professor Sanjar Bekov

December 9, 2024

**Introduction.**

As we know obesity is one of the hugest problems in the modern world where by statistics 43% and more of adults have extra weight. At the beginning of the project, this was the main reason why we decided to create a project called LizCoach. The decision to create a program that would not just be entertaining, useful, or uphold the status of a completed final project, but rather a program that would somehow contribute to improving the world and combating obesity in the modern world. The symbol of the LizCoach program is a lizard, which serves as the title of the program as Liz (Liz from the word lizard, like a lizard coach). This inspired us because, by nature, lizards have a remarkable ability to regrow their limbs, such as their tails. While for other animals, the loss of a limb would be considered quite sad, given that it would be permanent, lizards continue their full lives with the process of regeneration. However, despite this, they continue their full lives with the process of regeneration, which is symbolically linked to the fact that people, thanks to the LizCoach program, will begin their process of recovery towards a more flexible and healthier life, without losing faith in their best lifestyle and healthy body. The program tackles obesity issues through several features, including calorie counting and a calculator, as well as exercises and fitness plans for different body parts. However, of course, the main lever of success is still the user themselves and their desire to change through the program. The program is divided into two main parts: the calorie calculator and workouts. The calorie calculator allows the user to see a list of products, which includes more than a hundred items, but if the user does not find the desired product in the list, they will have the chance to add their own product through a special feature for adding custom products in the calculator section. The uniqueness of the program lies in the fact that, first and foremost, users will have the chance not only to see which exercises they should do, but they will also have the opportunity to view the visual execution of these exercises through pre-provided links to YouTube videos of the exercises. Combining these two main functions as a solution to the problem, or at least some improvement, we can call it another lever on the path to self-improvement. A user who has decided to tackle weight loss can use our program as a full-fledged daily routine for counting calories and performing exercises. Additionally, our program can be used as a free bonus or supplement for people who are already engaged in fitness in professional settings, providing convenient extra tips and the ease of controlling food intake for balancing not only external but also internal actions. The target audience for our program will be middle-aged people, as the foundation of the exercises is based on recommendations from trainers who advise middle-aged individuals. Additionally, it was chosen to focus on the fact that these exercises are not entirely suitable for people over forty, considering the health conditions of older individuals. There was also an age restriction set for those under 14 years old, since the number of eating disorder victims is increasing, particularly among those aged 10-14, due to pressure on teenagers and the standards of the "ideal figure," as well as issues with nutrition and anorexia.

**Requirements**

**Functional Requirements**

* User Profile Management:
* Users are able to enter and update their personal information (name, age, weight, height, and gender) by editing or uploading files such as (data.txt).
* The application puts age limits with a range of 14 and 40 years of age.
* **Dynamic Gender Customization:**

Users would be able to set and amend their gender which in turn changes menus and functionality.

* The gender information gets saved into data.txt.
* **Workout variations:**

The application contains a ‘menu exercise’ which is prepared with hands, legs, stomach and full body.

Each of the seating workouts has the equivalent activity that has been preset in the defined placeholder functions (hand\_open, leg\_open, etc).

**Navigation:**

* Users can toggle from one page to another which includes the main menu, the gender changer and the workout pages.
* “Back” buttons are available for users to undo an action and come back to the menu that they previously were on.
* **Error messages:**

Validation are done for all empty fields and invalid inputs (e.g. when the user inputs their age in fraction instead of a whole number or when the age entered is less than the rang or greater than the range).

All the error messages activated by the showerror action messagebox are printed whenever an error occurs.

* **Persistence:**

The application is designed in such a way that it will help write and also retrieve user data (data.txt) that will be the same after several logs in and out of the application.

* **Non Functional Requirements**
* User Interface:

The application is designed in a nice fashionable design with appealing fonts, colors and a decent touch of a modern look that has been achieved by the use of customtkinter.

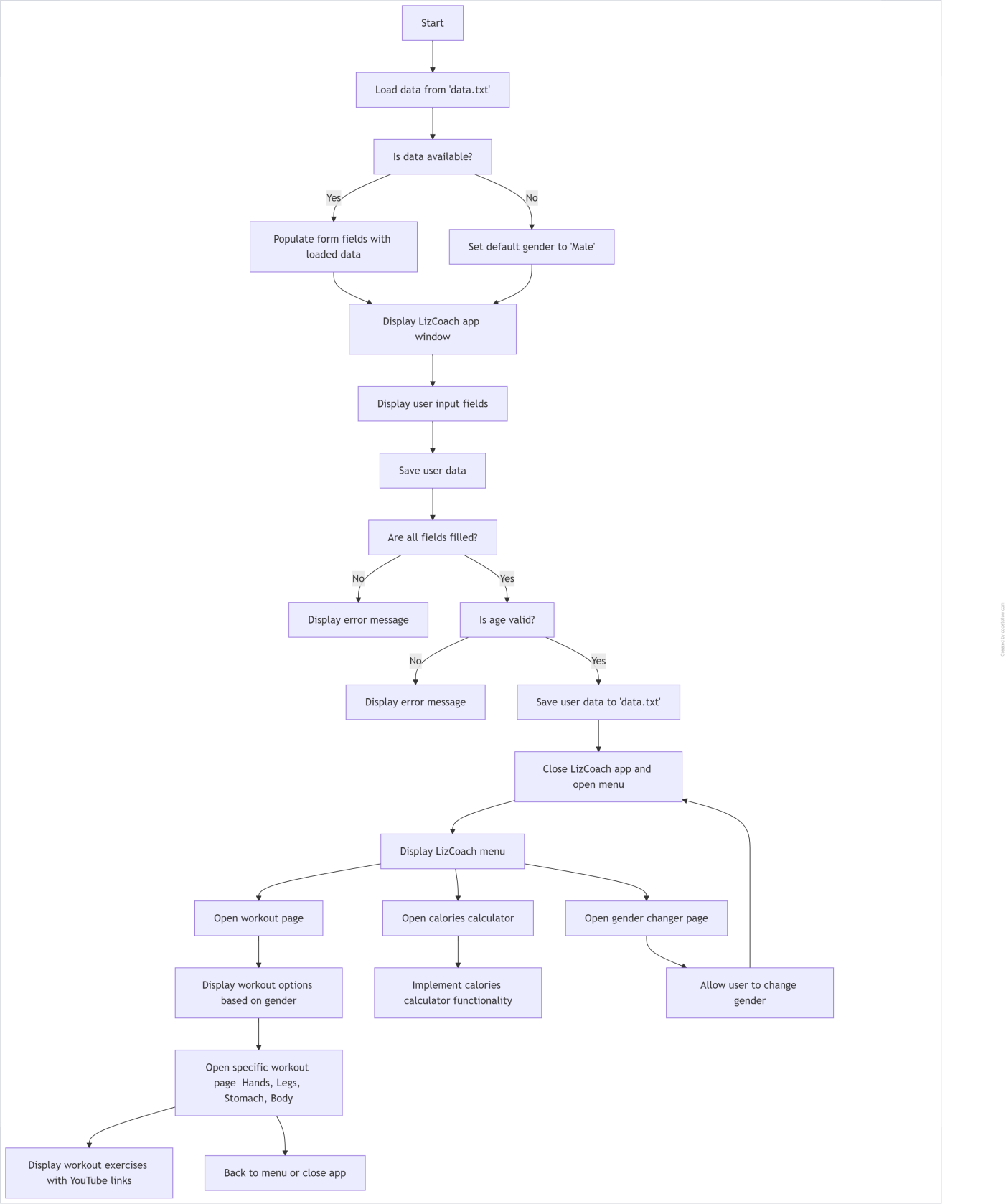
The application is built in such a way that it will have a constant padding and sizing of buttons and labels and still have a responsive layout.

* Data storage:

The data files for the user are stowed within a plain text file known as data.txt to allow easy retrieval of data files.

Error Notifications Error messages are easy and straightforward, as the user is well informed on any input that has been omitted or is wrong. Reusability Features such as the close\_and\_open\_menu() and CTkButton commands using parameters enhance reusability of code.

**System Architecture:**



**A diagram of a flowchart

Description automatically generated**

**Implementation:**

The development of the system for the LizCoach program was carried out using the Python programming language, with the tkinter library for creating the graphical interface as the foundation and an improved version of customtkinter, which has more enhanced and new features. Also, during the development, the PIL library and a database were used. The program contains a user interface, which is created through a single window for entering user information, a main menu window that changes depending on the selected gender, which can be changed through the gender change function, which also includes a single window. This function does not create a separate menu or separate windows; it specifically controls the output of information on the screen, divides the information into two parts, and displays each according to the selected parameters. Separate windows were also created for the introductory parts of the workout function and the calories calculator function, where starting in the workout, the function has its own window for selecting the main type of exercises and menu-style buttons for selecting and entering new parts of the program, where all the newly created buttons like “hands,” “legs,” “stomach,” and "body" have their own windows for displaying information (which also depends on the gender selection).

- hands - is divided into two parts, just like all the other parts, which are divided into female and male. Each gender has its own different videos and exercises. Hands has its own window, which has back and close buttons at the bottom, allowing the user not only to exit but also to completely close the program and return to the main menu. The main part of Hands consists of three embedded types of exercises, such as push-ups, triceps dips, and dumbbell bicep curls. Despite the program showing the same exercises for both genders, the execution of these exercises is different, and through the links embedded in the code, which are displayed as the exercise names themselves, both genders can see how to perform these exercises and also view additional exercises for these types. Videos are embedded differently for both genders using the webbrowser library.

- Legs: For the legs option, a special window has also been added to display leg exercises. The program will be divided into two parts where the main types of exercises for both genders will be squats, lunges, and calf raises. Just like in the hand section, each title has a background link to a video that is gender-specific and shows a YouTube video of the exercise being performed.

- Stomach - The stomach option, which is the third button on the workout window, opens a new window with stomach exercises that include flutter kicks, plank taps, and leg raises. Each exercise includes a YouTube performance that is categorized by gender.

- Body - and the last part of the workout includes a full-body workout with three exercises such as heel touches, crunches, and jumping jacks, which will help warm up the body and stretch before starting exercises for any body part.

Since the main aspect of the workout section is not the selection of exercises but the exercises themselves, a special visual feature for the exercises has been added to the program, where the user will have the chance not only to view the list of exercises and their sets but also to click on an exercise to be redirected to a YouTube video demonstrating that type of exercise and to watch new bonus exercises. These YouTube videos will help the user understand the program in more detail, specifically how each exercise is performed. And compared to other workouts, this method will motivate a person much better, somewhat simplifying their search for these exercises online while also combining sets of exercises for a specific body part and providing general YouTube videos on the topic. Each main type by which the workout program is divided has its own main page for convenient user use and clearer visibility. Next in the structure, we have the calories calculator function. For the calorie’s calculator menu, a main window was created, which includes five buttons and one label, where one of the buttons has a back function and the other four are "product list," "add product," "calculate calories," and finally "total calories calculated." The buttons lead to their own windows for these functions, where the tasks of the functions are:

- product list - to show the user a list of main types of products as buttons to choose from, which lead to the list of products of those types. The list includes 11 products such as fruits, vegetables, oils, beverages, sweets, snacks, grains, proteins, meat products, milk products, and finally user-added products. Each type has its own product window. The window is made using a frame, to which a scrollbar has been added that contains information about the name and calories of the product for one fruit or vegetable or 100 grams of the product, and for liquids, per 100 ml.

- Add product - a window that will allow the user to add their own product in case the product the user wants to calculate is not on the list. The program will use CTkEntry to ask the user to enter the product name and the calorie content per serving or per hundred grams. And when the save button is pressed, a message box will appear where, if the product name is found in the file, the message box will show a warning that the product already exists in the list, and the bot will not add it to the product list in the products list function, as well as to the general product list in the text file productslist.txt.

- Calculate calories is the most important part of the calories stage, where through its own window it will show the user a photo via PIL, a label for the product name, and two CTkEntry fields for entering the product name and portion (the portion is based on one hundred grams of the product, or one piece if the product is a type of vegetable). When the calculate button is pressed, the program first searches for the product name in the productslist.txt file, and upon finding it, takes the value of the product (since the file has a dictionary structure, it saves the product name as the key and calories as the value) and multiplies it by the entered portion. When the operation is complete, the program displays the total calculation through a message box. In case the product is not found, it issues a warning that the product was not found in the list. Upon successful calculation, the numbers are added to the total variable to keep track of how many calories the user has consumed. The information about the consumed product and the portion multiplied by the calories is also saved in the dictionary of consumed items, which stores everything that has gone through the calculator.

- Total calories calculator: Just like the calculator, it has its own scrollbar created on a frame that allows you to see the total amount of calories consumed and the list of consumed items (based on the products and portions consumed in the calculator) at the top of the window, displaying everything in the scrollbar.

**Challenges and solutions:**

When working on the program, the main difficulty, of course, was dealing with the core libraries like tkinter and customtkinter. GUI libraries in Python, thanks to which we were able to implement the idea of creating a visually ready program that gives the user a chance to use the program as conveniently as possible without unnecessary effort. First and foremost, this difficulty for us was resolved by studying the topic through supplementary YouTube videos, as well as thematic articles and websites. Also, the difficulties of adding and combining other Python libraries such as PIL/Pillow, database, web browser, and AST can be considered, which were also studied using this approach. Another challenge in creating this code involved working with files for the user section and the product list, where the former required handling constantly updated information and inputting it into a new file, while for the calculator, the file needed to be opened to search for product names from the filled file and, if necessary, to input new information that the file would retain. When working with the file for the calculator, this difficulty was resolved by the ast library. (abstract syntax tree). Ast was added to safely output calorie information from the file into a Python object. Since the complexity itself involved searching the file specifically by the key, value criteria, the file was initially fully read by the program and transferred to data, and then the line was transferred to products using ast. Ast was the solution that made working with the file easier and simplified it by converting the information from the file into a string. Also, later during the search, a simplified version of the product name was used through CTkEntry, and the search for this product was initiated in the products string. To input a new product, a line of code was written to work with the file using open and write. Next, the difficulties were in establishing links to show YouTube performances of the exercises, which were also resolved by using a new library called webbrowser. This library allowed us to simplify the connection with other websites and redirect the program's work directly from the compiler to the execution browser. Overall, this code did not require significant complexity, and the issues in achieving our program's goals took only a few extra hours thanks to the convenient work provided by Python and the versatile libraries that simplified our work in many areas. Also, as needed, many presentations shown during the lesson were used, and the basis of our knowledge was the PPT on GUI and the mini project provided to us during our Python lesson.

**Conclusion:**

Considering all the functions and features of the LizCoach program, as developers, we can confidently say that this program, if it were public, could have a positive impact on improving the world and, to some extent, add its own link to the enhancement of human health. As we mentioned above, of course, the main driving force is still the person's own spirit, the desire to change something within themselves, the desire to become better. And our program will be an excellent bonus for such people, and to some extent, it will strive to do its best to help its user with this issue. Thus, in the fight against excess weight, our program offers its two main features: body exercises and fitness and calorie counting, which help the user improve not only externally but also enhance their internal health, combining the beneficial with the effective. In the future, we would like to avoid limiting our LizCoach program in stages where it is dependent and instead allow for more flexible use without unnecessary effort. First and foremost, we would like to make the program independent from external files, as it currently requires the presence of necessary files with the required information for full functionality. Improvements may also include installing the program on a server to avoid the need for a constant compiler; however, of course, all these improvements are system related. To enhance the program specifically for users, we as developers would like to improve the program in various aspects, starting with exercises and types of workouts and ending with calculator functions for user convenience. For the workout section, first and foremost, there is a desire to add more body parts for exercises, as well as increase the number of exercises themselves, expanding from three to ten or more. Also, if there is an opportunity to develop the program to more global levels, I would like to negotiate with professional YouTube coaches regarding video workouts specifically created for the LizCoach program to have proprietary workouts from professionals without the interference of unnecessary trainers. This scheme will help keep the program more professional by having our own advice in each area, as well as support the copyrights of other entities that were used in the creation of our program. Next, as part of the program improvement measures, we would add a larger number of products to the list for more extensive use and possibly expand this field with some tips related to nutrition and food, creating a new section that would be related not only to calories but also to healthy recipes with fewer calories. Taking all this into account, we primarily hope for the use of our program for positive purposes, without promoting anything that goes against body positivity or encourages an unhealthy relationship with one's body, such as the development of anorexia, eating disorders, and unhealthy attitudes towards food. LizCoach should not drive people into a bad state but should positively influence thoughts about self-improvement without harming health. Our expectations will be the use of the program for personal purposes, as well as the promotion of a healthy body without any obligations, where every user of our program will have the chance to healthily assess their body and balance their diet with healthy, tasty, and low-calorie food, along with exercises. And most importantly, remember that even though the lizard is small, if it wants to, it can regrow its lost tail.

**References:**

1. *Free AI Art Generator: Create AI Art from Text - SeaArt AI*. (n.d.). Seaart AI. <https://www.seaart.ai/?https%3A%2F%2Fwww.seaart.ai%2Fhome%3F&ad=WWGGC1SEARCH&gad_source=1&gclid=Cj0KCQiAx9q6BhCDARIsACwUxu5o8O3ogX3AijiWbnOFRK0o91Sn5nGQBoUsKXcNSQCX6_sIEpsnce4aAjaeEALw_wcB>
2. Tkinter․com. (2023, August 8). *Modern GUI design with CustomTkinter! - Tkinter CustomTkinter 1* [Video]. YouTube. <https://www.youtube.com/watch?v=Y01r643ckfI>
3. ITH. (2023, January 8). *СОВРЕМЕННЫЙ ГРАФИЧЕСКИЙ ИНТЕРФЕЙС НА PYTHON, используя библиотеку customtkinter* [Video]. YouTube. <https://www.youtube.com/watch?v=ibf2u-rVb6o>
4. TurbineThree. (2023, September 19). *Make Tkinter look 10x better in 5 minutes (CustomTkinter)* [Video]. YouTube. https://www.youtube.com/watch?v=Miydkti\_QVE
5. Coding Lifestyle 4u. (2024, May 26). *Employee Management System in Python with CustomTkinter & MySQL | Full Project For Beginners* [Video]. YouTube. <https://www.youtube.com/watch?v=a5iCRrygWxk>