**Channeling Hygieia: A Graphical Guide to Attaining Healthy Living**

*Senior Project*

*In Partial Fulfillment of Bachelor of Science in Computer Science*

Yessica Rodriguez

*Supervisor: Dr. Mahmoud K. Quweider*

*December 10, 2018*

**Table of Contents**

1. Deliverables, Timeline and Status for the Project
2. Abstract
3. Research and Database
4. GUI
5. Retrieval and Collection of Data
6. Appendix
7. References

**I.** **Deliverables, Timeline and Status for the Project**

|  |  |  |
| --- | --- | --- |
| **#** | **Deliverable** | **Expected Date of Completion** |
| 1 | Learn JavaFX and MySQL | Sep 9 |
| 2 | Collect data | Sep 23 |
| 3 | Set up database | Sep 30 |
| 4 | GUI design | Oct 21 |
| 5 | Uploads user information to database | Oct 28 |
| 6 | Retrieve and display data from database | Nov 4 |
| 7 | Adapts recommendations to user information | Nov 11 |
| 8 | Testing and bug | Nov 18 |
| 9 | Testing and bug (… continued) | Nov 25 |
| 10 | Demo & presentation | Dec 10 |

**II. Abstract**

This project uses a graphical user interface to allow the user to view and input information into a MySQL database. Information about foods, exercises, diseases and condition, and allergies are already stored in the database. In the Profile tab, the user can create an account and input their personal and medical information. If the user already has an account, he or she can sign in. Once signed in, a user can view their information stored in the database and edit it. In addition, the user can get recommendations based on the information they have entered. Using the data stored in the database, recommendations are given to the user. Under the Diet tab, a user is able to get nutritional information of foods in the database based on the food’s category or by searching the food by name. In addition, the number of calories consumed can be calculated based on the grams of food eaten. Furthermore, information about exercises can be found under a tab titled Exercise based on exercise type or muscle worked by that exercise. There is a tab called Additional information. This tab has university contacts, local organization contacts, and website links to provide other ways the user can obtain further information or assistance on achieving a healthy life.

Deliverables are in the form of a graphical user interface (GUI) that is JavaFX-based. The application was developed using NetBeans with Java as the primary programming language. MySQL was used to create databases and SQL was used for database manipulation.

**III. Research and Database**

**A. Research**

Since dietary, exercise, and medical information will be provided to the user, research was done to collect all the necessary information. The data on diseases and condition, allergies, foods, and websites with information on healthy living was obtained from the database of UTRGV Fall 2017 Software Engineering I Final Project, which is a project I worked on previously [1]. For diseases, conditions, and allergies, the data obtained contained an id, name, and description for each [14-15]. For the dietary information, the major food groups were researched, and the types of food found in each. For several foods the the nutritional information – name, food group, calories, fats, cholesterol, sodium, potassium, carbohydrates, proteins – was obtained from the database [13]. After this information was obtained from the Fall 2017 Software Engineering I Final Project, information pertaining to the exercises was obtained. The major muscle groups in the body were researched and recorded [22-24]. In addition, major exercise types were researched and recorded [19-21]. Then, exercises that worked each muscle were researched and the following information of each exercises was recorded – name, description, image, type of exercise, muscles it worked, the link used to obtain the description, and the link used to obtain the image. Furthermore, research and records were obtained for foods that would be beneficial for people with the diseases previously recorded [2, 5-8, 17-18]. In addition, the amount of exercise and calories to consume recommendations based on age were researched and recorded [3-4]. Finally, options for possible security questions were researched and recorded [16].

**B. Database**

The database was created using MySQL, and the MySQL Workbench was used to create the tables, in which the information obtained would be added. In addition, the user’s information would be added once entered by the user. A schema called seniorproject was created along with the following tables:

allergy medical

disease muscles

exercises personal

exercisetypes profilepicture

exermusc userallergies

food\_groups userdis

foods users

disfood security\_questions

recfood\_m recfood\_f

recexer\_age

The allergy, disease, foods, medical, personal, and profilepicture tables were created using the sql files from the UTRGV Fall 2017 Software Engineering I Final Project [1, 9-15]. Appendix A demonstrates the Enhanced Entity-Relationship (EER) Diagram of the schema created. The diagram demonstrates the tables and the relationships between them. The users, personal, medical, userallergies, userdis, and profilepicture tables are connected by a common field, user id number. The food table and the food\_groups tables are connected by the common field group id number. Finally, the exercise tables – exercisetypes, exerdis, and exermusc – are connected with the common field exercise id number. Furthermore, some tables although not connected are related. These tables include the userallergies to allergy, userdis to disease, and exermusc to muscles. However, these tables were created in a different format than the others since the relationship is many-to-many. An id, either for user or exercise, was only done once while the other component – allergy, disease, or muscle – was not. A column for each of the available options for the other component was created, in which a Boolean variable could be inserted. For instance, in the table userdis, the attributes are userID, dis1, dis2, dis3, dis4, dis5, dis6. If the other component was applicable for the id of either user or exercise, then the value would be ‘1’ as opposed to ‘0.’ Furthermore, the tables recfood\_m and recfood\_f both contain the following attributes – age\_l, age\_h, sedentary, moderately active, and active. Similarly, the recexer\_age table contains the columns age, low, high, and message. Furthermore, the security\_questions table contains the attributes id, and q, which would contain the questions. The remaining table, disfood, contains seven columns that are the source, disease id, and five columns for the food id of five foods beneficial for people with the disease or condition. After the database was created, the information obtained through research and from the Fall Software Engineering Final I Project was inserted into the appropriate tables.

**IV. GUI**

A JavaFX-based GUI application was created using NetBeans. The application was coded in the MVC Model. Each tab and subtab will have at least three classes the Model, View, and Controller. The View class will focus on what will be displayed in the pane. On the other hand, the Model will involve obtaining the data from the database. Finally, the Controller will connect both the Model and View and define the actions of the buttons. The skeleton of the overall design was obtained from the GUI application TestingMVCModelMultipleTabs created by Dr. Quweider, who provided the project to every student taking the FALL 2017 Software Engineering I course [35]. In addition, the overall idea for this project was designed from the UTRGV Fall 2017 Software Engineering I Final Project [1]. The application contains four tabs – Login (Profile), Diet, Exercise, and Info (Additional Information). Each tab has an icon obtained from online [25-26,31,33].

**A. Login (Profile)**

The Login View can be viewed in Appendix B. There are two text fields for the user to type their username and password. In addition, it includes several buttons – the ‘Create Account’ button (see point a), the ‘Forgot Password’ button (see point b), and the ‘Sign In’ button (see point c).

1. Create Account View

Figure B.1 displays the three possible Views. Each of the views will have a button that reads “Continue” that are named c1, c2, and c3. In the points below, the views will be described. The first view is displayed when the ‘Create Account’ button is pressed. The following point is displayed if the c1 button is pressed and all the information asked for in the previous view are entered by the user. If not, an error message will be displayed in the previous view. If the c2 button is pressed, the third point is displayed. The c3 button will return the view to the original Login View. Similarly, each view will also have the ‘Back’ button at the top that will return to the original Login View.

1. The view consists of six labels (messages or text), six text fields, and three drop-down menus. The user is prompted to type into three of the text fields their username and their password twice by three of the labels. The other three labels notify the user the number of each security question. The three drop-down menus are located next to those labels and allows the user to choose their security questions [16]. The three text fields left are located below each menu for the user to type into the answer of each security question. At the bottom, there is the c1 button.
2. The view contains the Edit View personal box with the c2 button at the bottom. This view is described in more detail below (see point d).
3. The view contains the Edit View medical box with the c3 button at the bottom. This view is described in more detail below (see point d).
4. Forgot Password View

There are three possible views that could be displayed, which are shown in Figure B.2. The views will contain buttons that read “Enter” that are named e1, e2, and e3. All three of these buttons have conditions that if not met when pressed an error message will appear. In addition, e1 button can even return to the original Login View if its conditions are not met. On the other hand, if the conditions for these buttons are met the view displayed will change. Furthermore, the first point below describes the view displayed when the ‘Forgot Password’ button is pressed. If the conditions for e1 button are met, it will display the view described in the second point. Similarly, the view described in the third point is displayed if the conditions for e2 are met when it is pressed. Finally, the e3 button returns to the original Login View, if the conditions for e3 are met. Similarly, each view will also have the ‘Back’ button at the top that will return to the original Login View.

1. The view contains one label and one text field. It prompts the user to enter their username. In addition, it contains the e1 button at the bottom.
2. The view consists of three labels and three text fields. The three labels are the security question number and security question. The three-text field are located underneath each label. In addition, there is the e2 button at the bottom.
3. The view has two label and two text fields. The two labels are asking for the new password. The two text fields are located next to each label. In addition, the view has the e3 button at the bottom.
4. Home View

The Home View can be viewed in Figure B.3. It consists of three sections. It consists of the Personal section, the Medical section, and the Profile Image section. In the Personal section, all the user’s personal information stored in the database will be displayed such as name, gender, address, and more. On the other hand, the Medical section, the user’s medical information is displayed. The medical information consists of weight, height, diseases, conditions, and allergies. Finally, the Profile Image section displays the uploaded user image. Each section has an edit button that has an icon obtained from online [28]. If the any of the edit buttons are pressed, the Edit View (see point d) is displayed. In addition, underneath the Profile Image section there is the recommendations button. If it is pressed, the Recommendations View (see point e) is displayed.

1. Edit View

Based on the section where the edit button is pressed, the Edit View changes. There are three possible edit view, which Figure B.4 displays. The Personal edit, Medical edit, and Profile Image edit are described in the points below respectively. At the bottom, there is an ‘Okay’ button that returns to the main Home View.

1. The Personal edit contains labels and text fields. The labels identify the personal information that a user can enter or has entered – first name, last name, gender, address, and more. The text fields contain the information in the database for the current user. The information can be modified by the user.
2. The Medical edit has the labels, text fields, and check boxes. The labels identify the medical information that the user can enter or has entered – weight, height, diseases, conditions, and more. In the text fields the user can enter or modify their recorded height and weight. Furthermore, there is a check box for each disease, condition, and allergy stored in the database. The user can click on the check box to mark it if they suffer or have the condition, disease, or allergy next to the check box.
3. The Profile Image edit consists of two buttons. The first button, ‘Upload Image,’ opens a File Chooser that allows the user to select an image saved in their computer. The code for this was obtained from an online website [36]. Once an image is selected, the image is displayed in the same size as it is in the Home View. Furthermore, there is a ‘Cancel’ button at the bottom that will return to the Home View.
4. Recommendations View

The Recommendations View, which is demonstrated in Figure B.5, contains labels and sometimes images. If the information being displayed is too long or too wide, a scrollbar appears to allow the user to view all the information. Furthermore, the information being displayed are calories per day, exercise time, and five foods to eat based on diseases or conditions.

**B. Diet**

The Diet View consists of three buttons – ‘Food Categories,’ ‘Calorie Calculator,’ ‘Search’ – and can be seen in Appendix C. Each of these buttons contains an icon that was obtained from the internet [29, 32, 34]. In addition, each button displays a different View that correlates with the button clicked. The ‘Food Categories’ button displays the Categories View. The ‘Search’ button displays the Search View. On the other hand, the ‘Calorie Calculator’ displays the Calorie View. Each view has at the top the ‘Back’ button to return to the Diet View.

1. Categories View

It consists of a heading and a box of buttons, one for each food category. Once a button is clicked, the View is changed to a view consisting of a scroll pane and a button, which return to the original Categories View. Figure C.1 demonstrates both possible views for Categories View. Furthermore, the scroll pane consists of boxes with the food’s nutritional information that are of the food category chosen. The box consists of a picture, name of the food, food group, calories, fats, cholesterol, sodium, potassium, carbohydrates, proteins.

1. Search View

The Search View, which is shown in Figure C.2, consists of a text field and a button. In the text field the user enters the name of the food he or she is looking for. To prompt the machine to search for the food in the database, the user must click the Search button.

1. Calorie View

Figure C.3 is an image of the Calorie View, which consists of two text fields, one slider, three buttons, two option or radio buttons, pie graph, and drop-down menus. The menus are to select the food category and the food the user ate. Then, one text filled displays the number of calories por portion. The slider is used to get the number of portions the user ate. Then, the total number of calories based on the calories per portion and the number of portions. In addition, the three buttons names are as follows: ‘Add to Plate,’ ‘Clear Food Choice,’ ‘Clear Plate.’ The ‘Add to Plate’ displays the new added food and calories into the graph. The clear food choice removes a food previously added to the graph. In addition, the two choice buttons allow the user to choose whether the graph will be divided by categories or food items. Finally, the pie graph displays the calorie, total calories, and categories of the food items entered to be calculated.

**C. Exercise**

Appendix D demonstrates that the Exercise View consists of two buttons, Types and Muscles. Both buttons contain icons that were obtained online [27, 30]. The Type button displays the Types View described below, while the Muscles button displays the Muscles View discussed below. Each view contains the ‘Back’ button at the top to return to the Exercise View.

1. Types View

It consists of a heading and a box of buttons, one for each type. Once a button is clicked the View is changed to consist of a scroll pane and a button, which return to the original Types View. Figure D.1 demonstrates both possible views for Types View. Furthermore, the scroll pane consists of boxes with the exercise information that are of the type chosen. The box consists of a picture, title (name of exercise), description of exercise, and two links that open in the computer’s default browser the website that the image or description were obtained.

1. Muscles View

The Muscles View, which is shown in Figure D.2, is very similar to the Types View, except instead of the exercise type it is muscle. The buttons in the original view are made up of one button for each muscle. In addition, the exercises whose information are displayed are those that work the muscle whose button was clicked by the user.

**D. Info**

The Appendix E demonstrates the Info tab. It consists of text that allows the user to know about the information that is provided in the website below. In addition, it includes the links of each website underneath the appropriate description. The links open the website in the default browser of the device being used. Most of the code was obtained from the UTRGV Fall 2017 Software Engineering I Final Project [38]. Small differences to the background type and overall design such as color scheme, size of font, and more were modified. However, overall the code remained the same.

**V. Retrieval and Collection of Data**

A JavaFX-based GUI application created using NetBeans was connected to the database with the Jar file mysql-connector-java-5.1.44-bin.jar and a Database class, which was obtained from the UTRGV Fall 2017 Software Engineering Final Project [1]. The class included the database host, username, and password that are used to create a connection using the functions predetermined in the Jar file and SQL java library. In addition, the retrieval and modifying of the data stored in the database are done using SQL commands defined in the Model class for each View and Tab, which call on the Database class.

**A. Login (Profile)**

The Login contains six main views that connect to the database for a variety of reasons. In the points below each view’s connection to the database and manipulation of the data are described in the points below.

1. Login View

After a user enter a username and password and presses the ‘Sign In’ button, the username is searched in the users table. If the username is found, the password for that username stored in the database is compared to the password entered by the user. If correct, the id number for the record with both the username and password are sent to the Home. If the password doesn’t match the password stored in the database for the username or the username entered doesn’t belong to any user in the database, the text fields are cleared for the user to try again.

1. Create Account View

In this view, it is ensured that the user has entered all their information into the text fields – username, password, password confirmation, and the answers for the three security questions. In addition, the three security questions must be chosen. Moreover, the password and password confirmation must match. If all the information has been correctly entered and the user presses the ‘Ok’ button, a new record with the same user id is created in the tables – users, userdis, userallergies, profilepicture, personal, and medical. Except for users, all the other tables have every other attribute as nothing (null) or ‘0’ only containing the user id. As for users, all of the information obtained the username, password, security questions ids and answers are stored in the corresponding fields with the appropriate user id.

1. Forgot Password

There are three possible views for the Forgot Password view. In the first view, as is described above, the user enters their username. Once the ‘Enter’ button is pressed in the first view, the database is searched for the username entered. Once it is found, the user id is returned. Then, in the next view, using the user id obtained in the previous view, the security question ids for the questions chosen by the user when he or she created their account are obtained. Using theses ids, the security questions are obtained and displayed for the user to see and enter the appropriate answers. Once the user has entered the answers for the three security questions and presses the ‘Enter’ button, the answers are compared to the user’s stored answers that were obtained when the user created their account. In the last view, the user is prompted to enter their new password twice. If the passwords match, the users table is update to modify the password stored for the user to the new one entered.

1. Home

In this view, the user id is obtained from the original Login View. Using the id, the information stored in the personal, medical, userdis, userallergies, and profilepicture tables is displayed.

1. Edit

In this view, there are three possible views that update the information in the personal, medical, userdis, userallergies, and profilepicture tables. In the first view known as Personal edit, the information stored in the personal table can be modified. Online websites were used to obtain the SQL commands to update the users “Date of Birth” [41]. Furthermore, as for the second view Medical edit, the information stored in the medical, userdis, and userallergies Finally, the last view Profile Picture edit, modifies the profile image of the user stored in the profilepicture table. Online research was done to know how to read the image, convert the image to the appropriate data type to enter into the database, and the SQL command to update a file [42].

1. Recommendations

From the Home, the user’s diseases, conditions, date of birth, and gender are obtained. Using a code obtained online, the user’s age is calculated based on the date of birth entered [37]. Based on the user’s gender, the recfood table to obtain the information is chosen. If the user is male, the table is recfood\_m. On the other hand, the table is recfood\_f when the user is female. Once the table is chosen, the user’s age is compared to the attributes age\_l and age\_h, which each record in these tables has. If the user’s age is with the range (greater than age\_l and less than age\_h), then these are the calories a day recommended to the user. Furthermore, from the table recexer\_age the exercise time recommendations are obtained. The user’s age is compared to the table’s attribute age. If the user’s age is less than the age in the database but greater than the previous records age, the exercise time recommendations in that record are displayed to the user. Finally, based on the user’s diseases and conditions, the disease and conditions ids are used to obtain the five food ids that would be beneficial for someone with those diseases or conditions to eat, retrieving this information from the disfood table. Using the food ids, the food names are obtained from the foods table and displayed to the user.

B. Diet

The Diet connects to the database and obtains food information stored in the database. Several buttons in this tab and its view retrieve food information and display it. Each button is listed below with a description of what and how it is obtained and displayed.

1. Categories

The number of food categories and the name of each are obtained from the food\_groups table. Once a food category button is chosen, the food category id is used to obtain all the foods from that food category from the foods table. All the nutritional information is obtained – food name, food group, calories, fats, cholesterol, sodium, potassium, carbohydrates, proteins.

1. Search

The image and nutritional information for each food is obtained from the database and displayed. All the data saved for the food that contains the food name entered by the user is retrieved from the database and stored. Then, the information for specific field or attribute is obtained from the result and assigned to the appropriate variable in the Search Model. Then in the Controller, the text fields from the View are obtained and their text are set to be the corresponding variable from the Search Model. Thus, the data obtained from the database is displayed. Most of this code was obtained from the UTRGV Fall 2017 Software Engineering I Final Project [39]. Minor changes were done to ensure all the nutritional information of the food is obtained at once instead of individually and the name of the database schema was changed.

1. Calorie

The database is accessed to obtain the options for the two drop down menus. The first drop down menu contains an option for each food category obtained from the table food\_groups. Each food category option contains a different set of options for the second drop-down menu. The options for the second drop down menu are obtained from the foods table. The records are filtered by the ones that have the id of the corresponding food category in the column, food group. Thus, the second drop down menu contains the names of the foods that belong to the food category chosen in the first drop down menu. The options for the drop down menus are displayed to the user. Once a user choses an option for both drop down menus, the slider moves to the appropriate place, the number on the number line that matches the weight in grams of the serving size of the food. Most of this code was obtained from the UTRGV Fall 2017 Software Engineering I Final Project [40].

C. Exercise

The Exercise tab connects to the database and obtains information pertaining to exercising from the database to display to the user. Several sections in this tab retrieve from the exercise table the image, image source, description, and source for each of the exercises based on each exercise’s id number. The information of the exercise is displayed by adding the components to the scroll pane from the View in the Controller. However, it differs on how the exercise id numbers are obtained that are of a specific type or work a specific muscle.

1. Type

It searches through the data in the exercises table and checks the type field to see which ones have a matching type id. If it does, it is added to the total number of exercises that are of that type. Then, an array of integers of the size of the total number obtained is created in which all the exercise id numbers of the exercises of that type are going to be entered. Then, the information in the exercise table is reviewed again and the id number of the exercise with the matching number in the type column with the type id is stored into the array. This array is then used to obtain the exercise information.

1. Muscle

The information in the table exermusc is obtained. The column of the corresponding muscles is checked if it is true, then it is added to the total count of the exercises that work that muscle. Then, the array of integers for the id numbers is created of the size of the total count. The information in the exermusc table is obtained again. The column of the corresponding muscle is checked if it is true, then the id number of the exercise is stored in the array. The array is then used to obtain the exercise information.

**VI. Appendix A**

**A screenshot of a cell phone

Description automatically generated**

**VI. Appendix B**

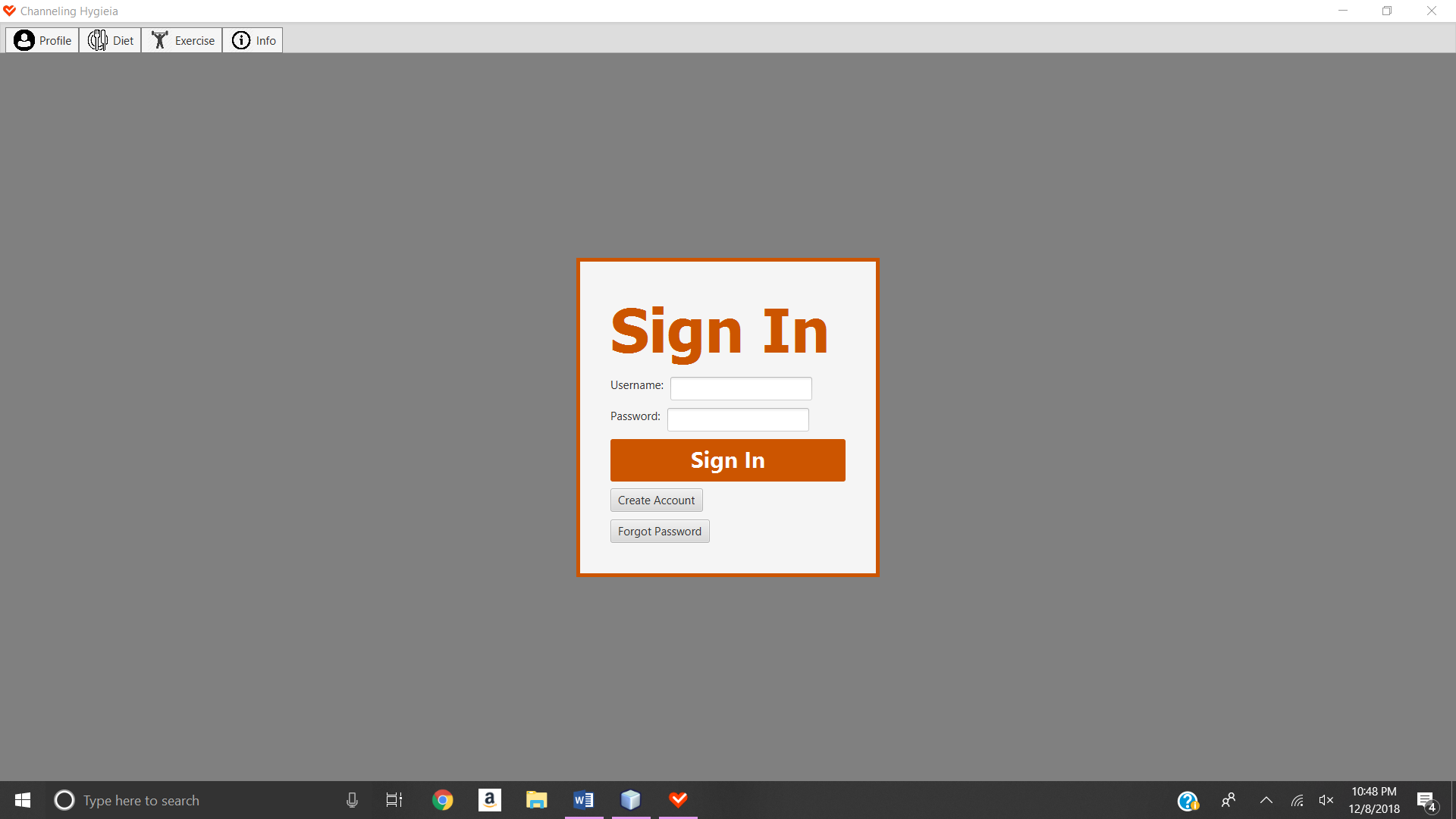
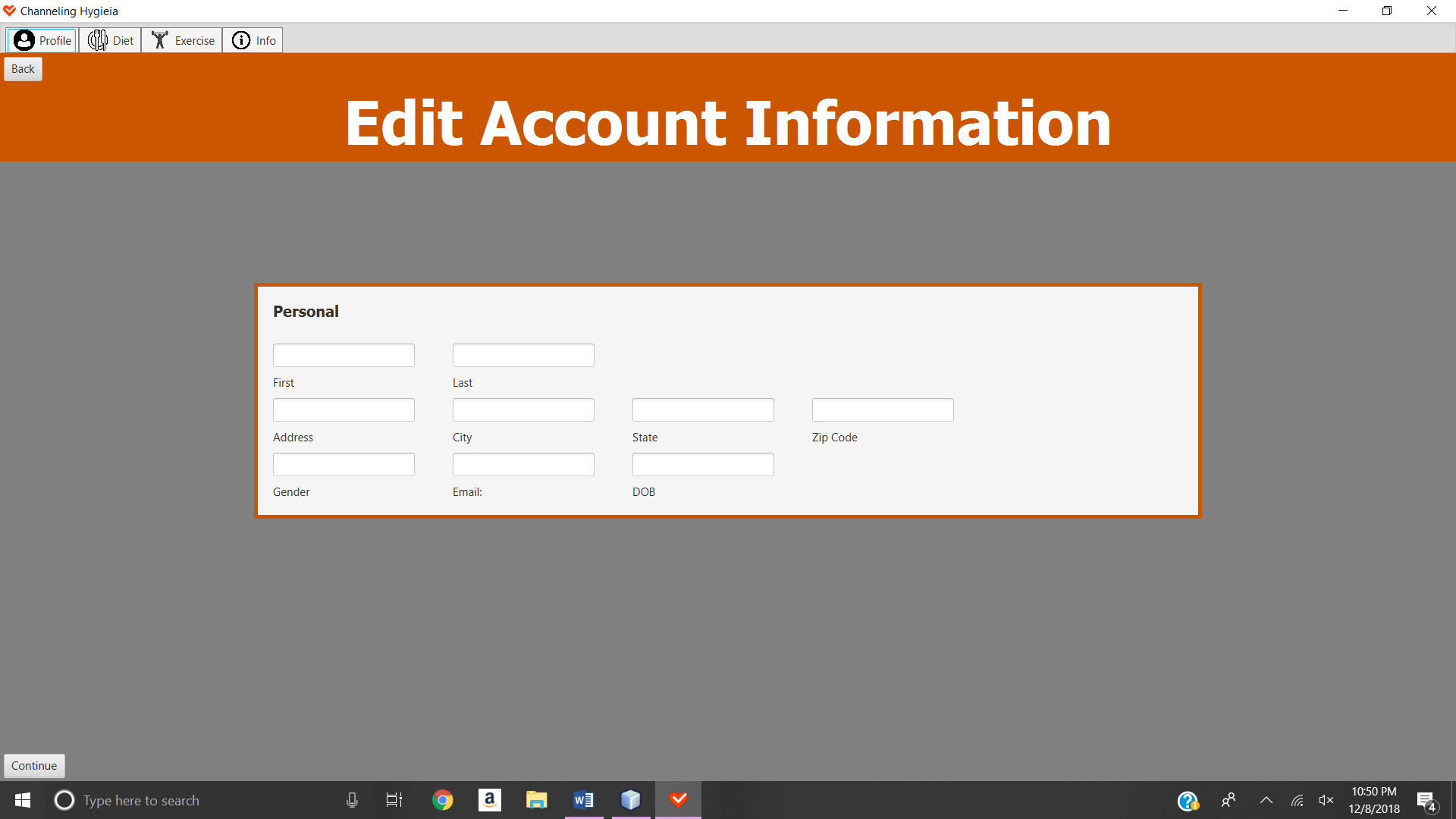
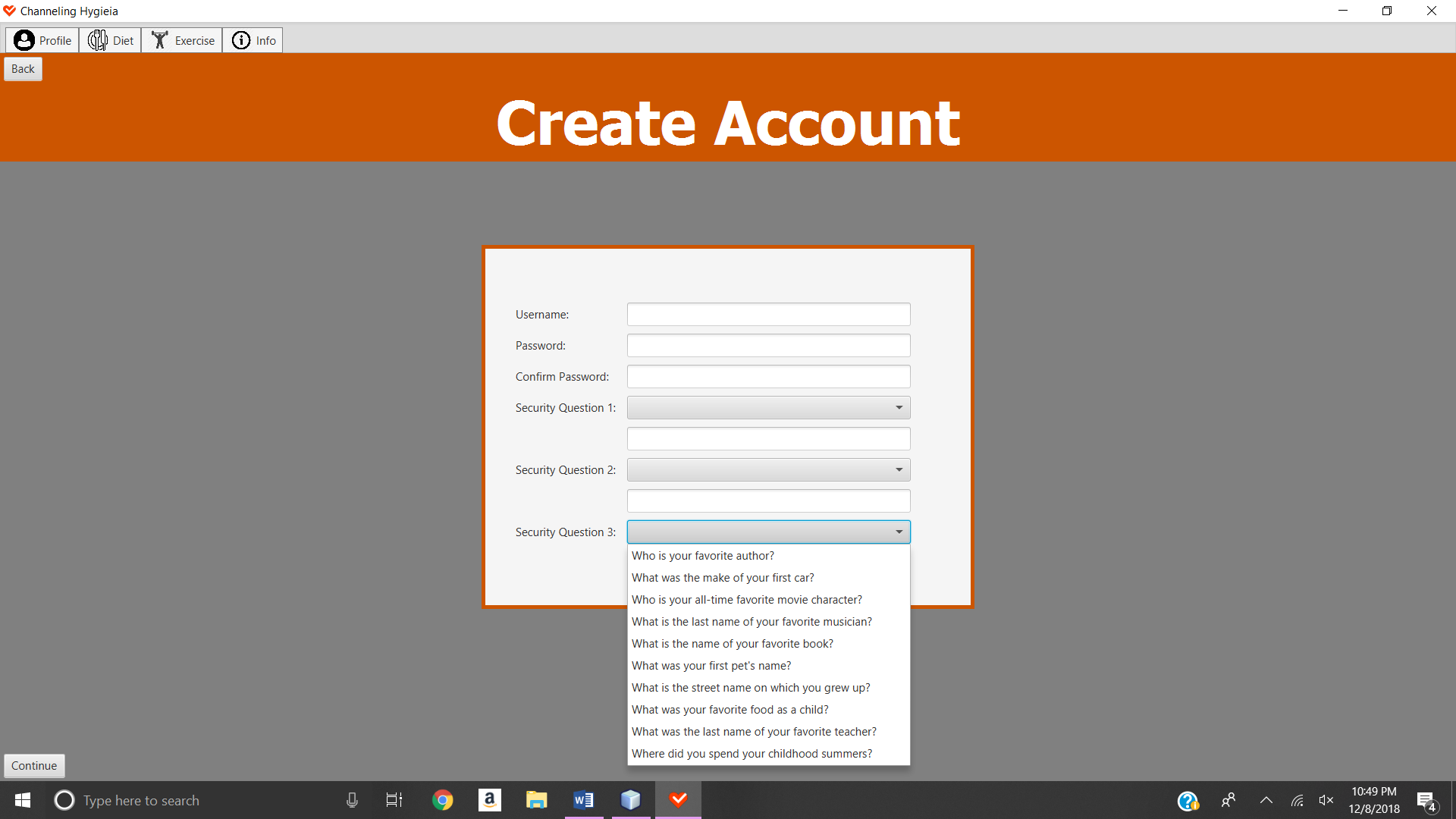


Figure B.1



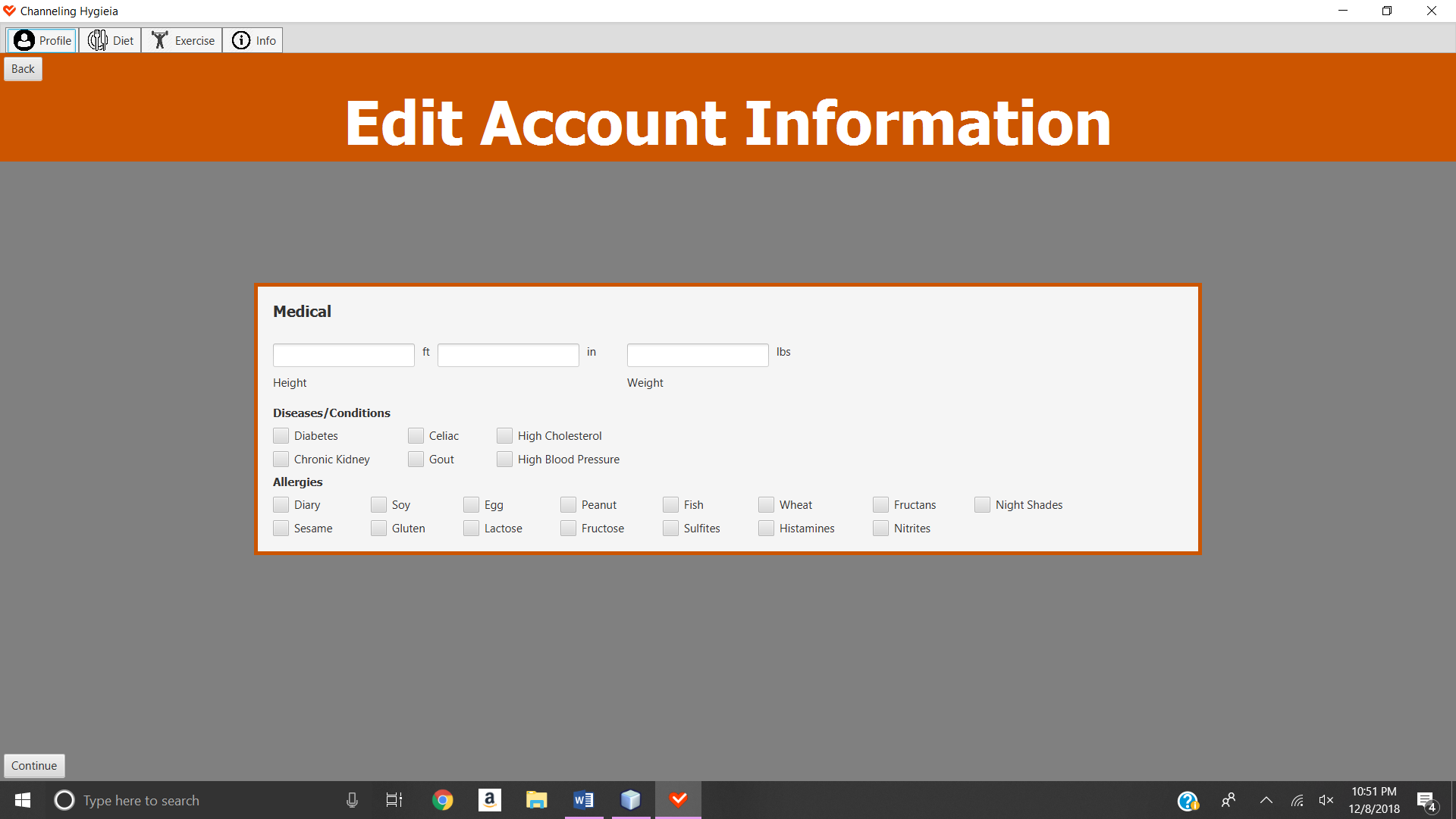
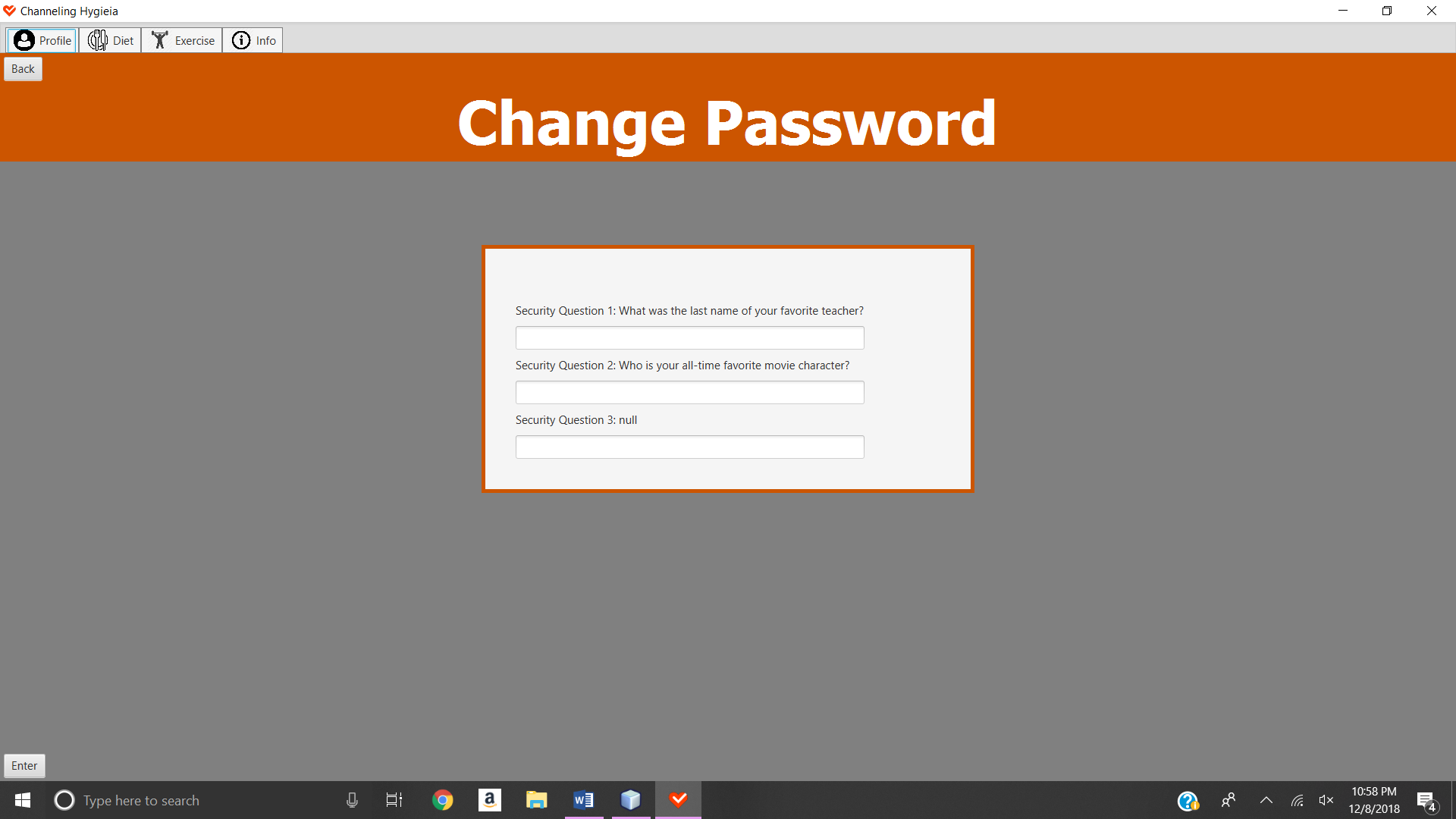
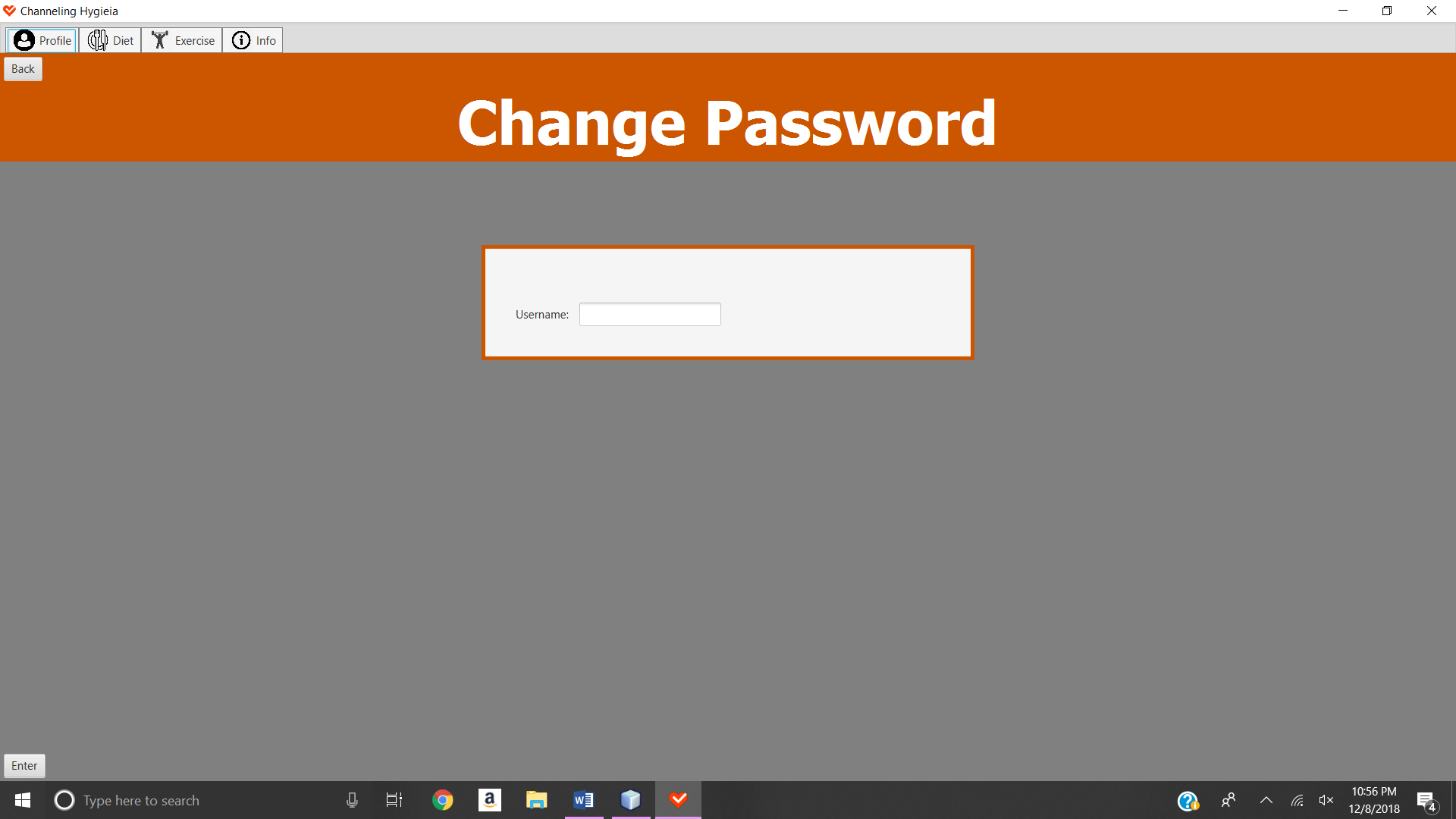


Figure B.2



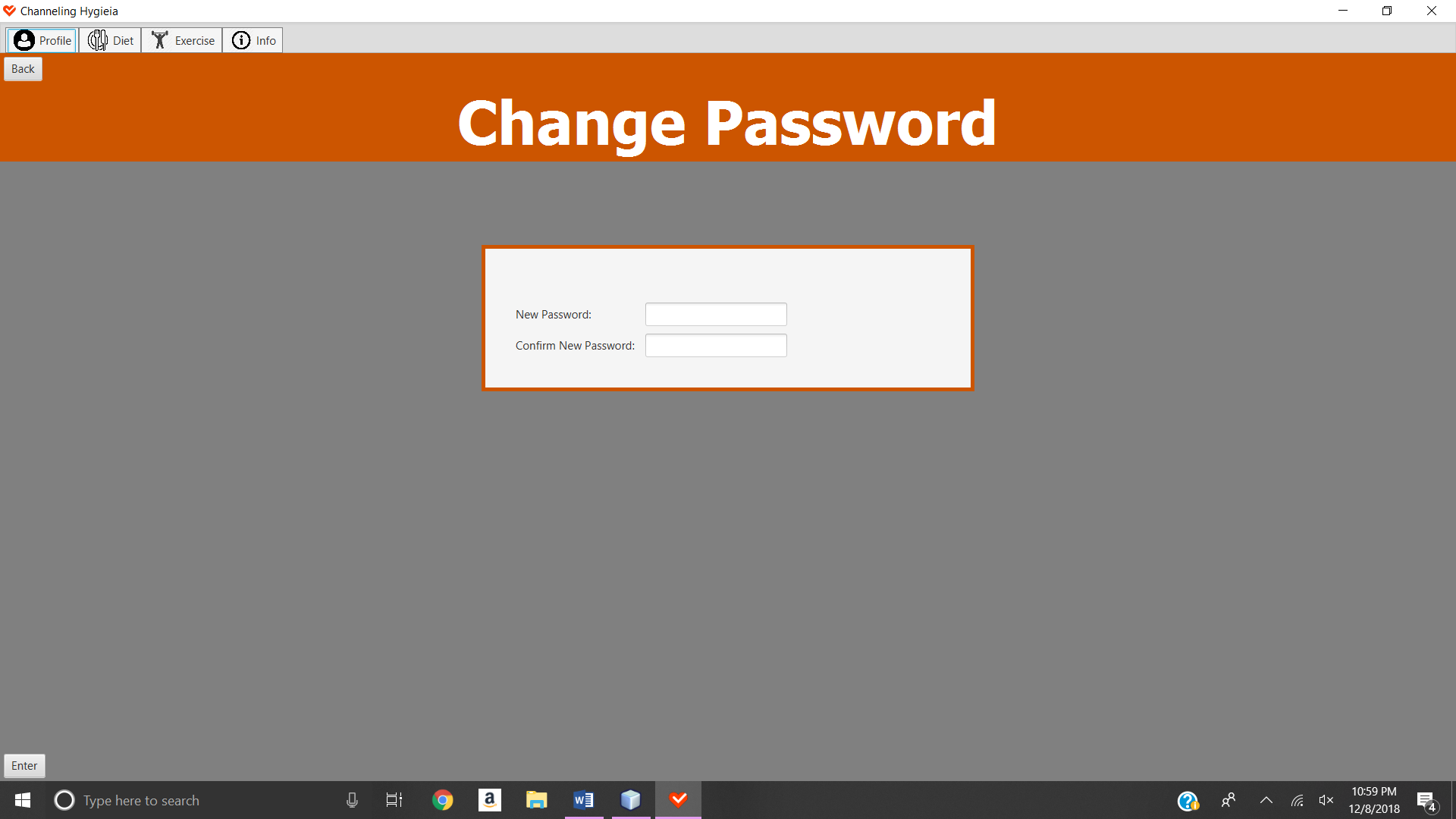


Figure B.3

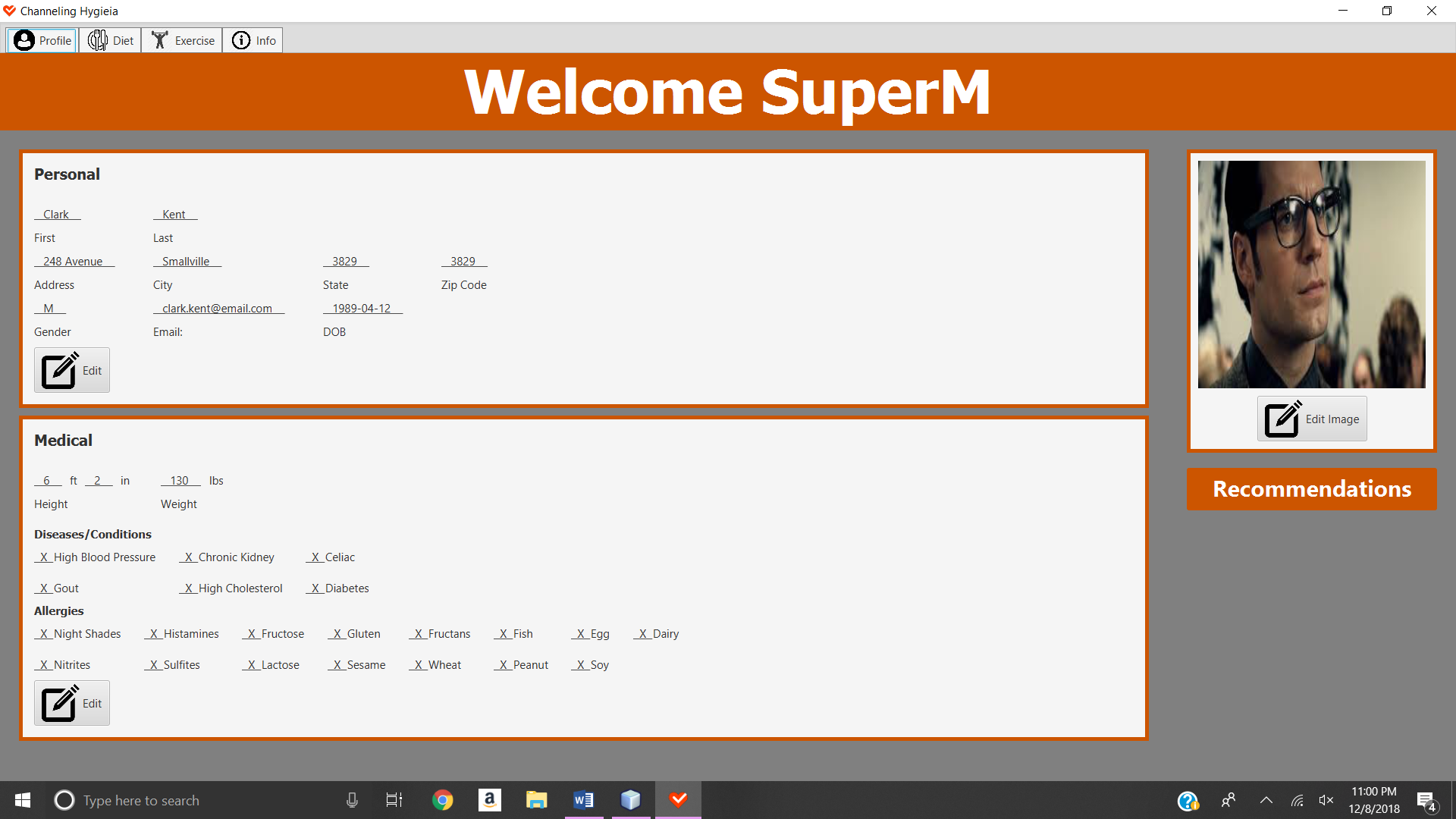


Figure B.4

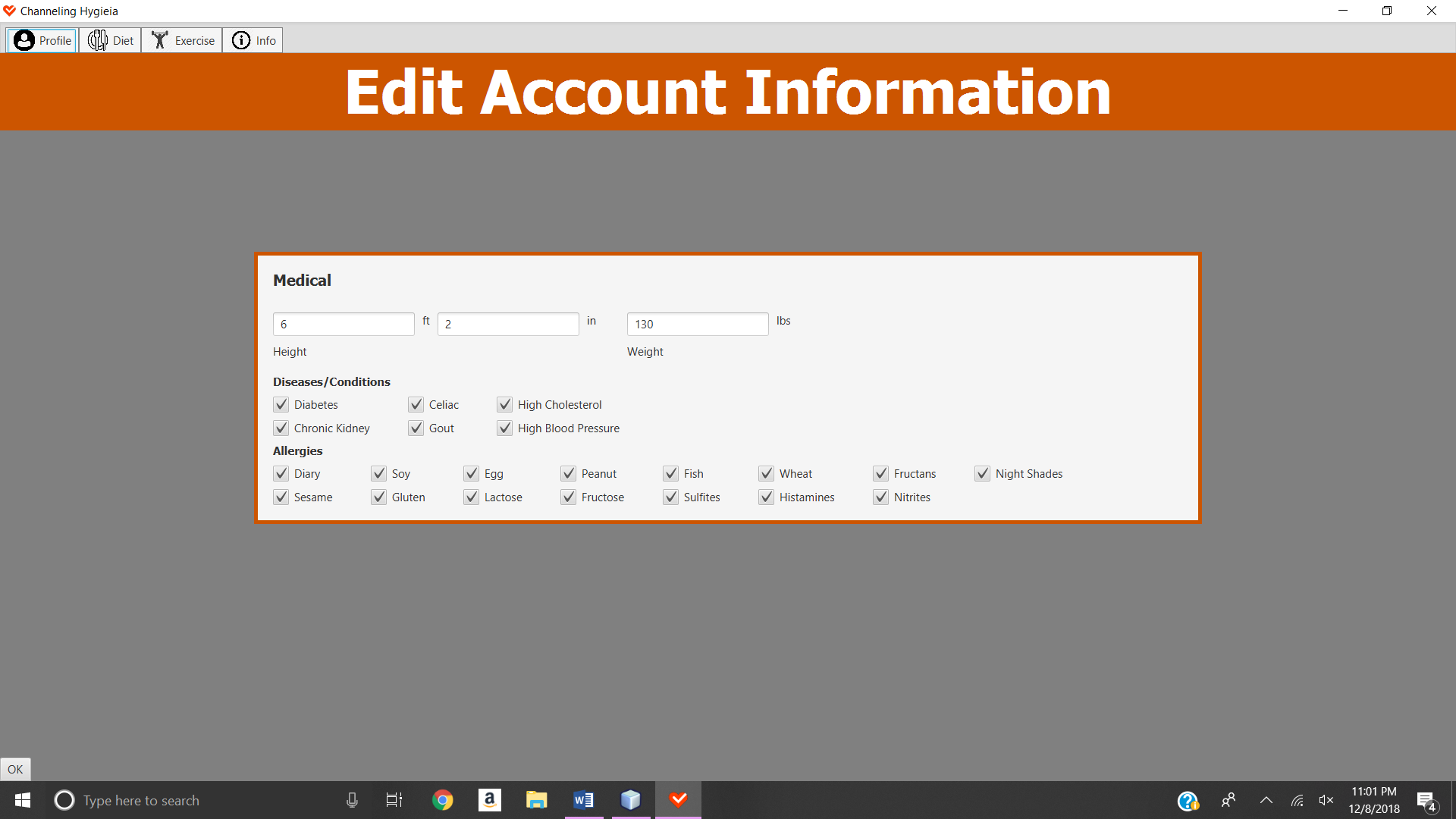
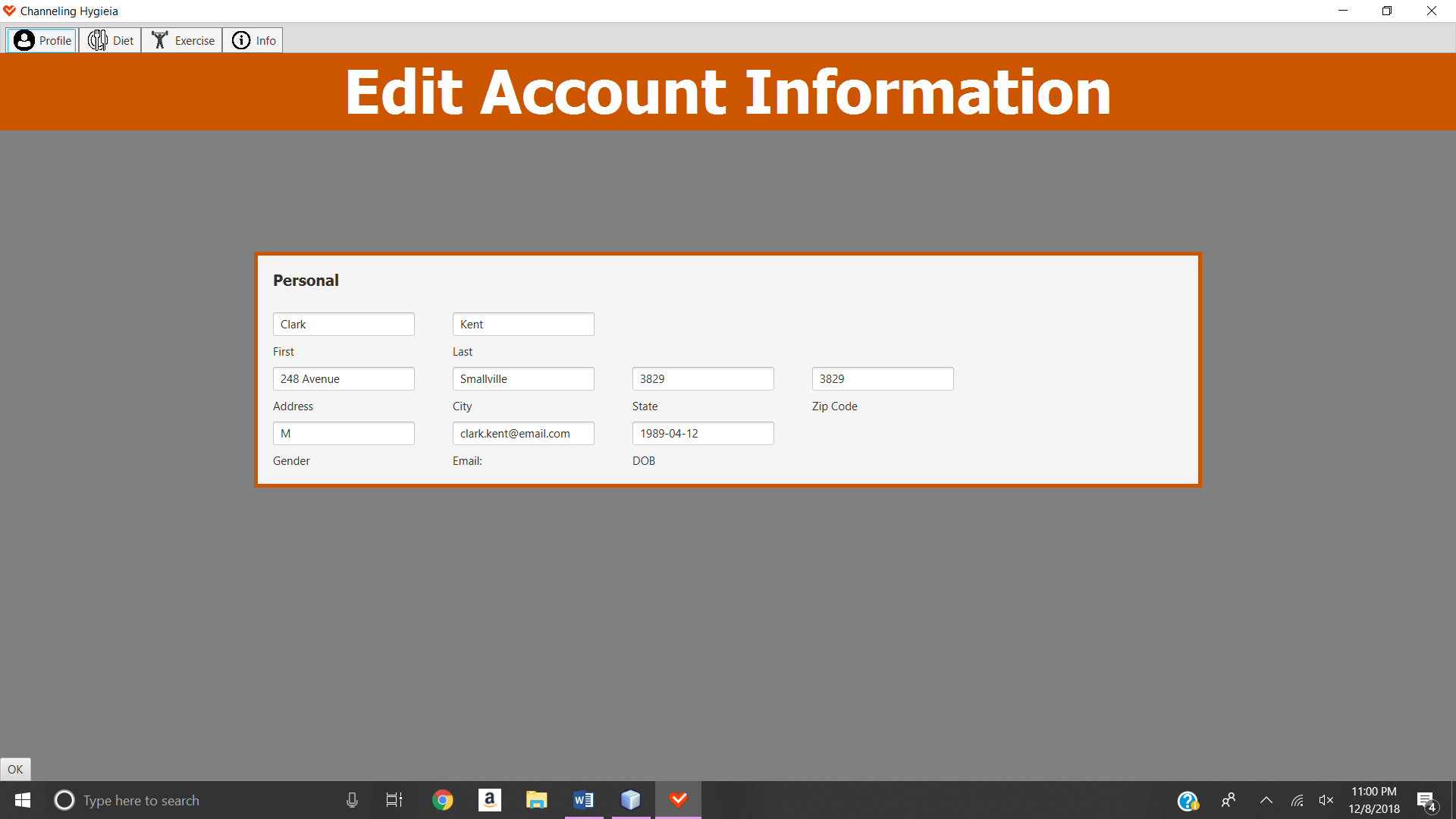
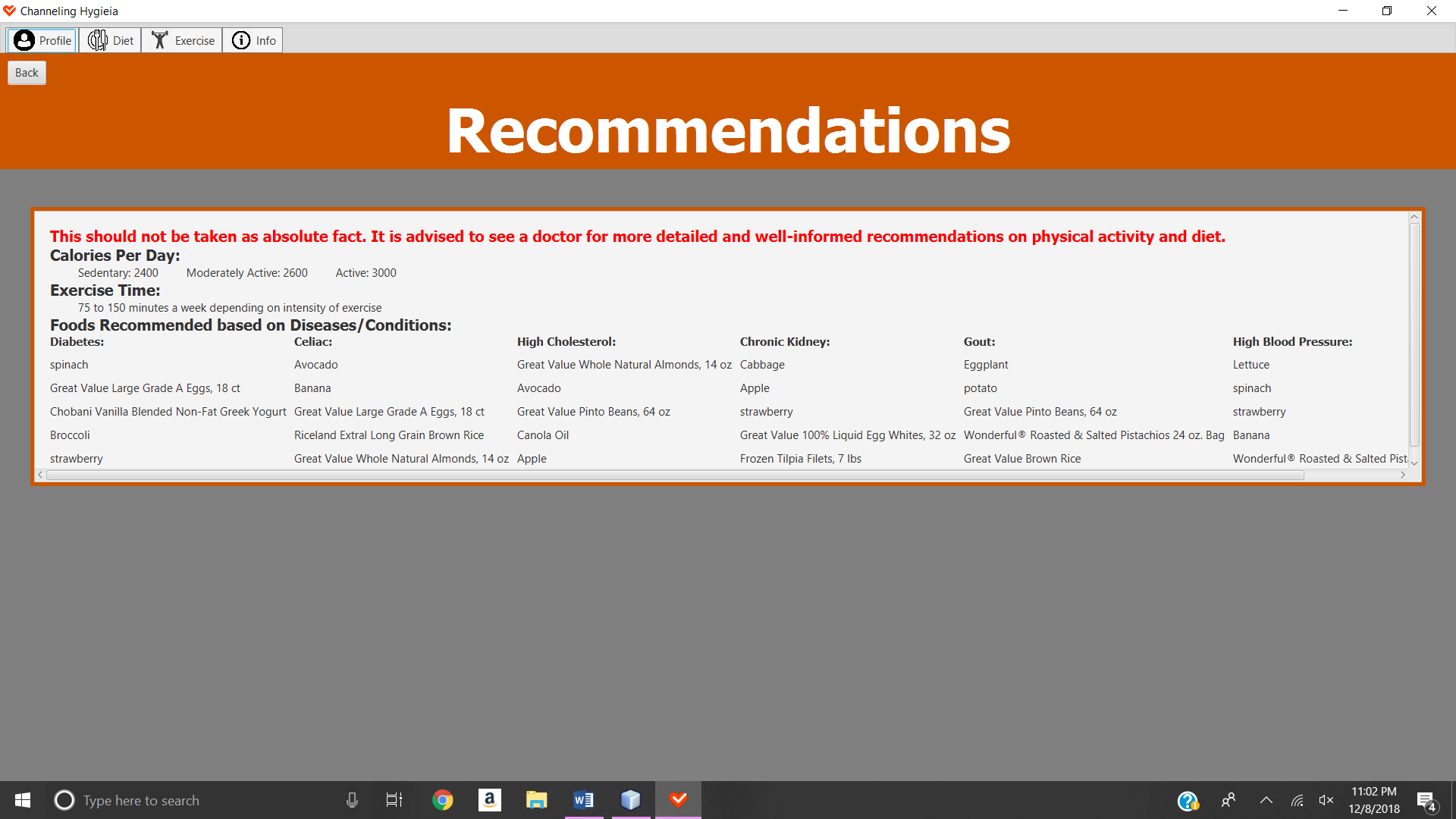


Figure B.5



**VI. Appendix C**

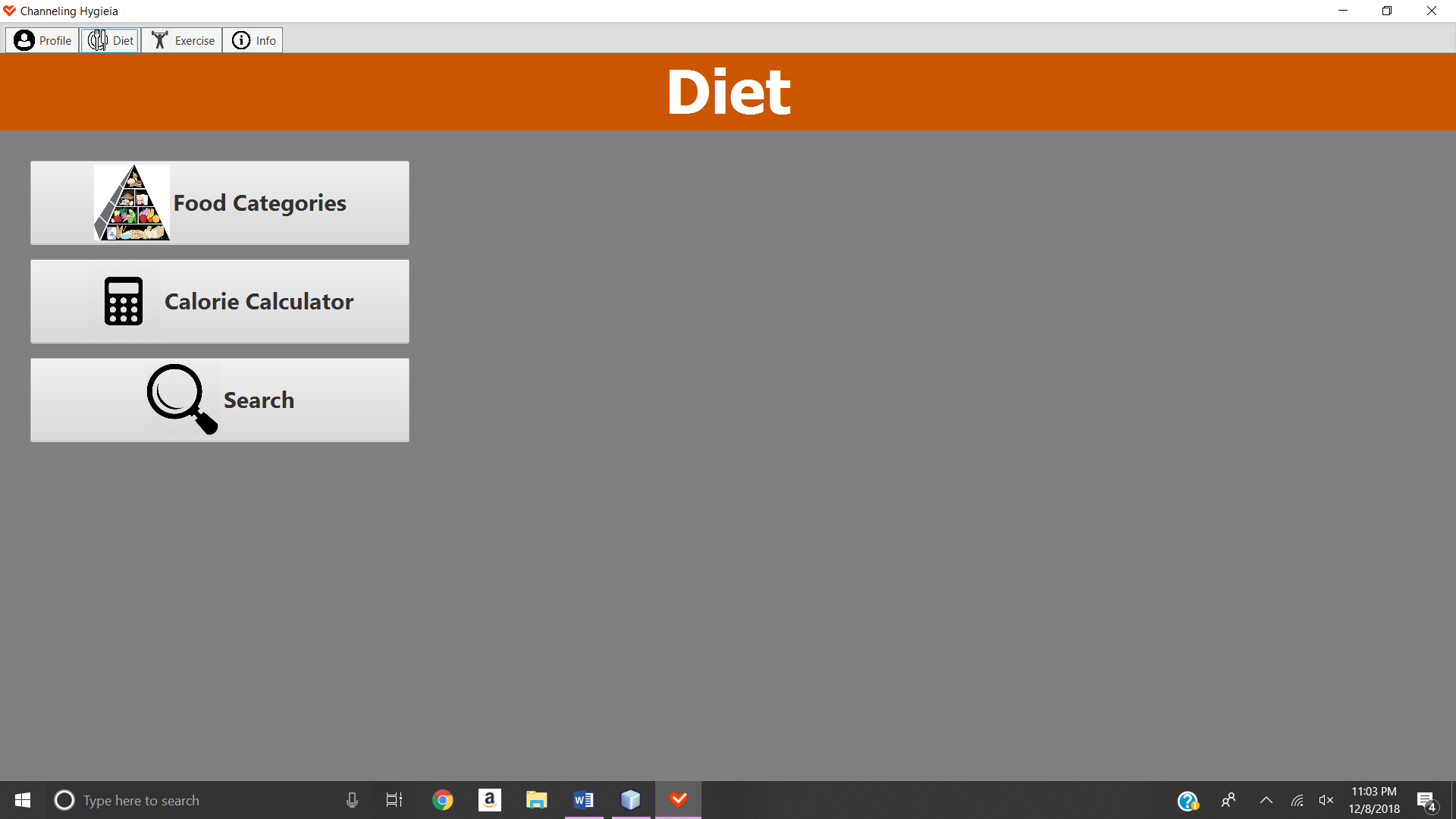


Figure C.1

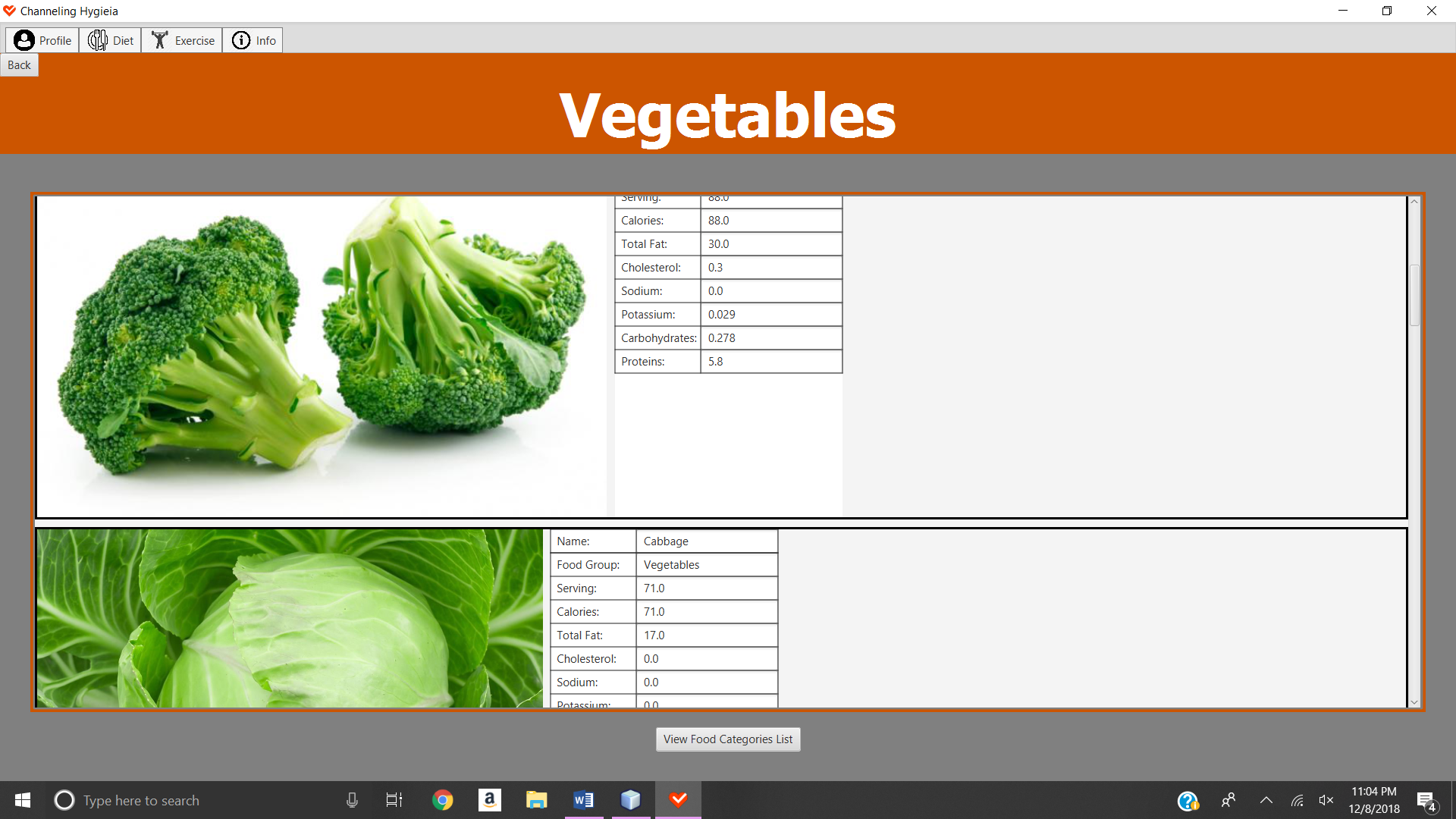
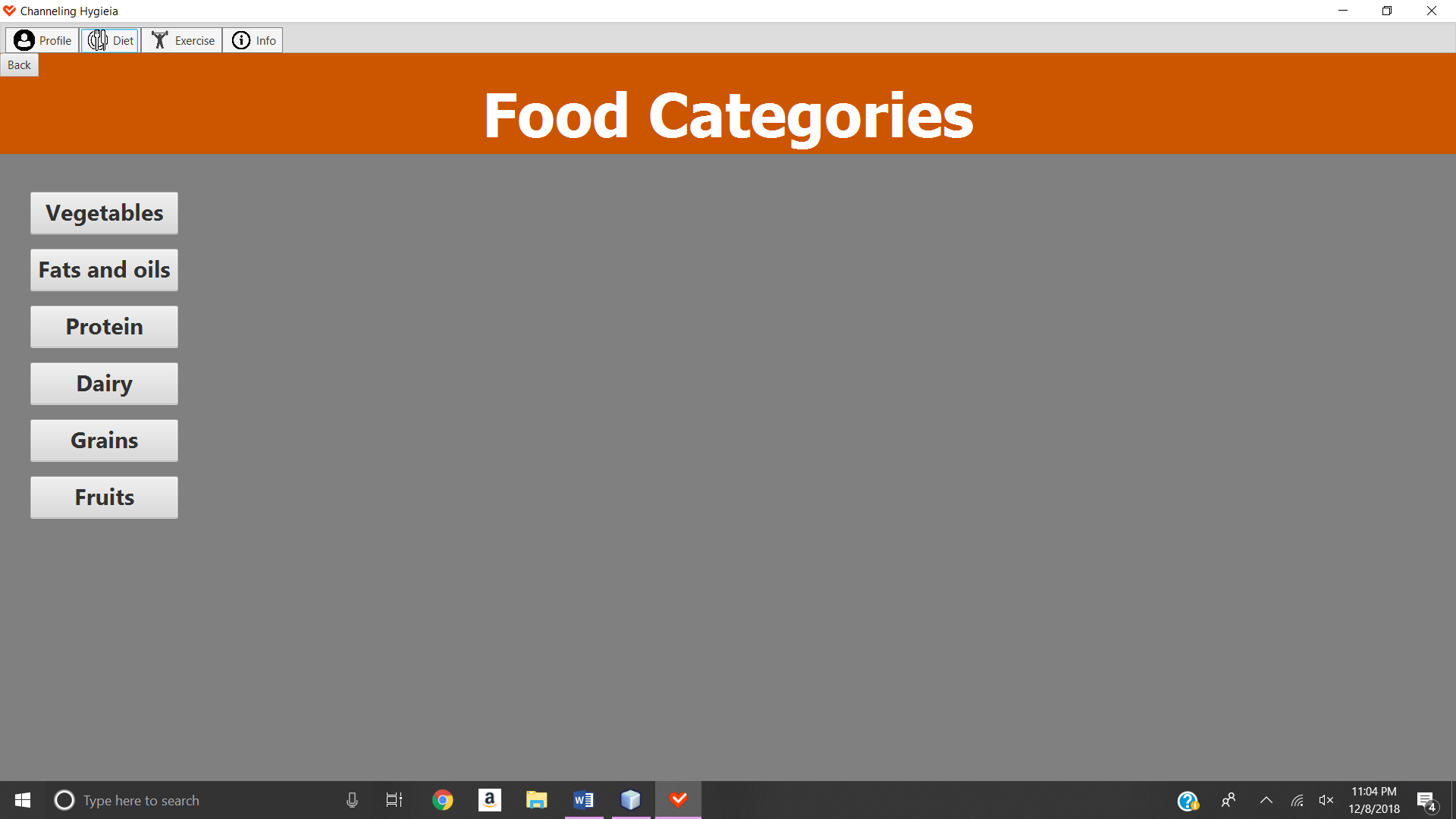


Figure C.2

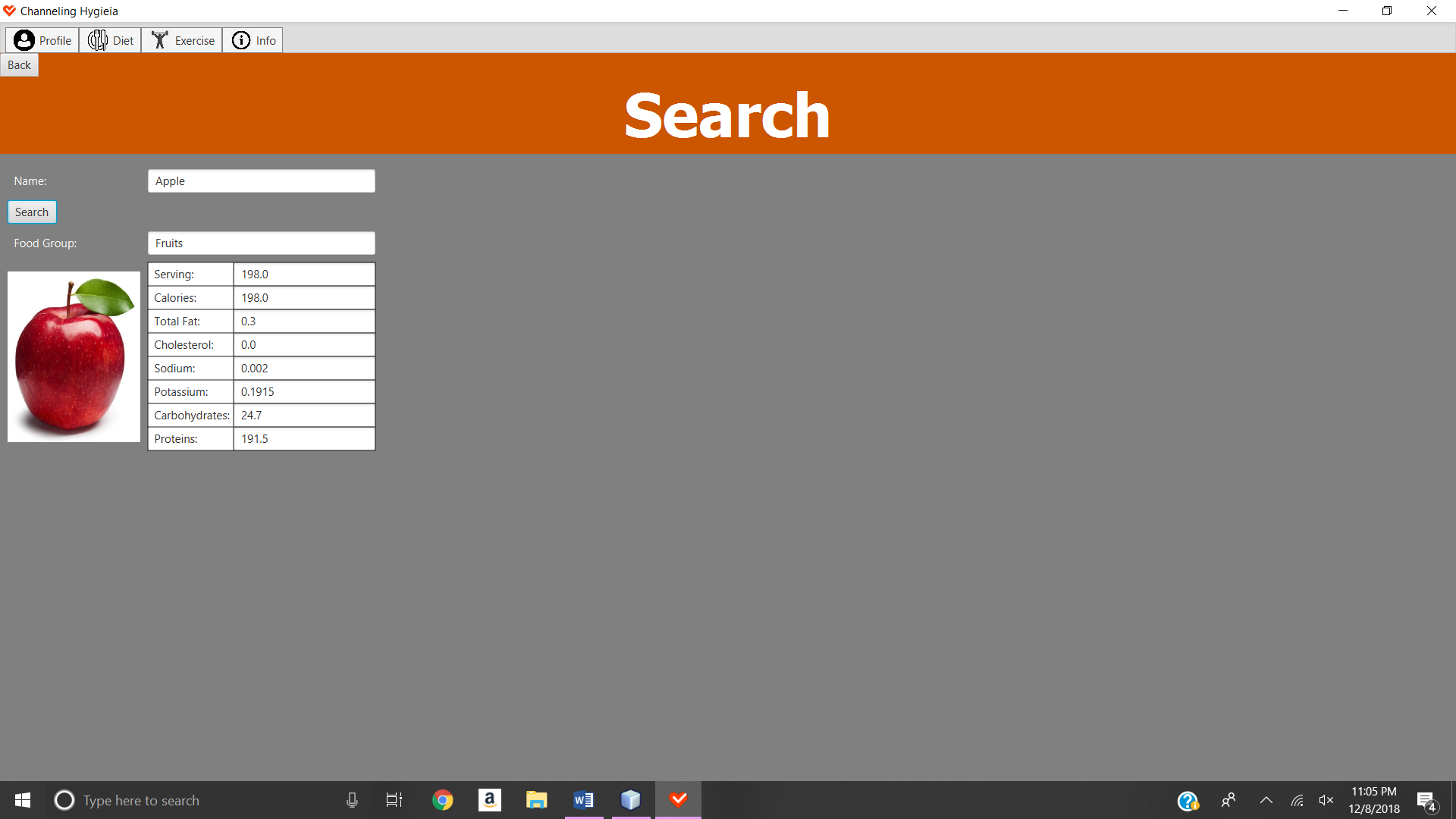
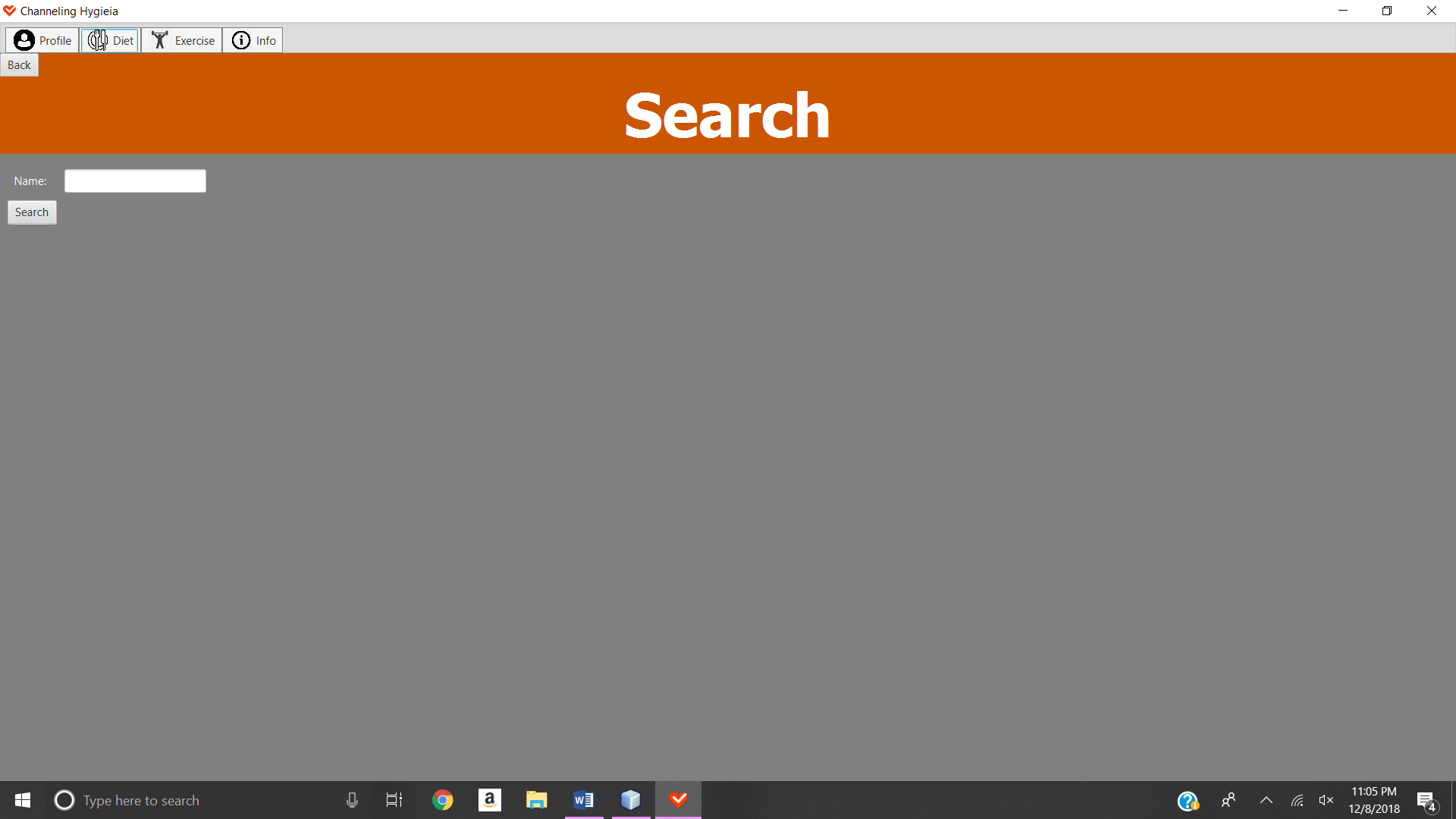
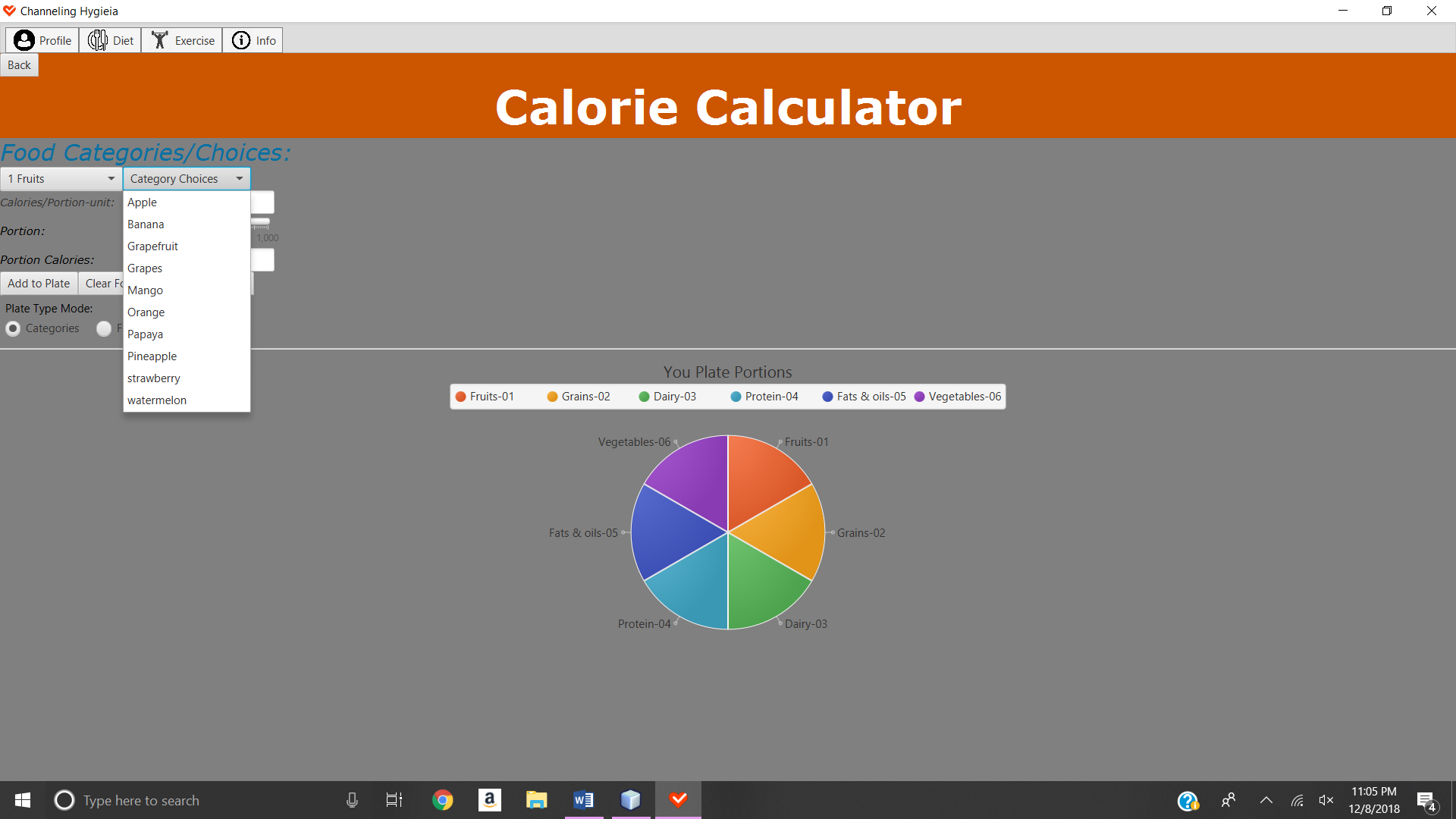
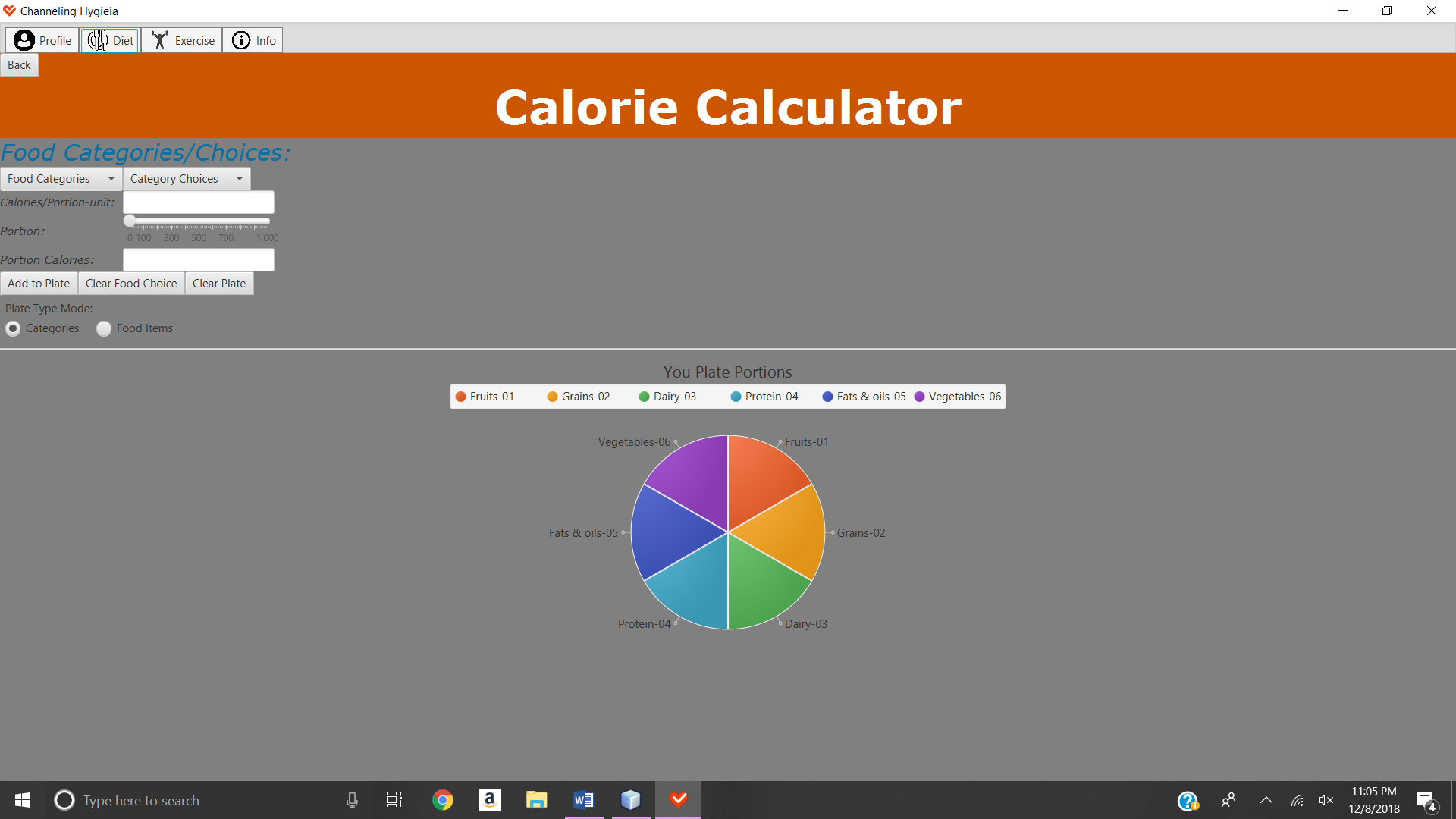
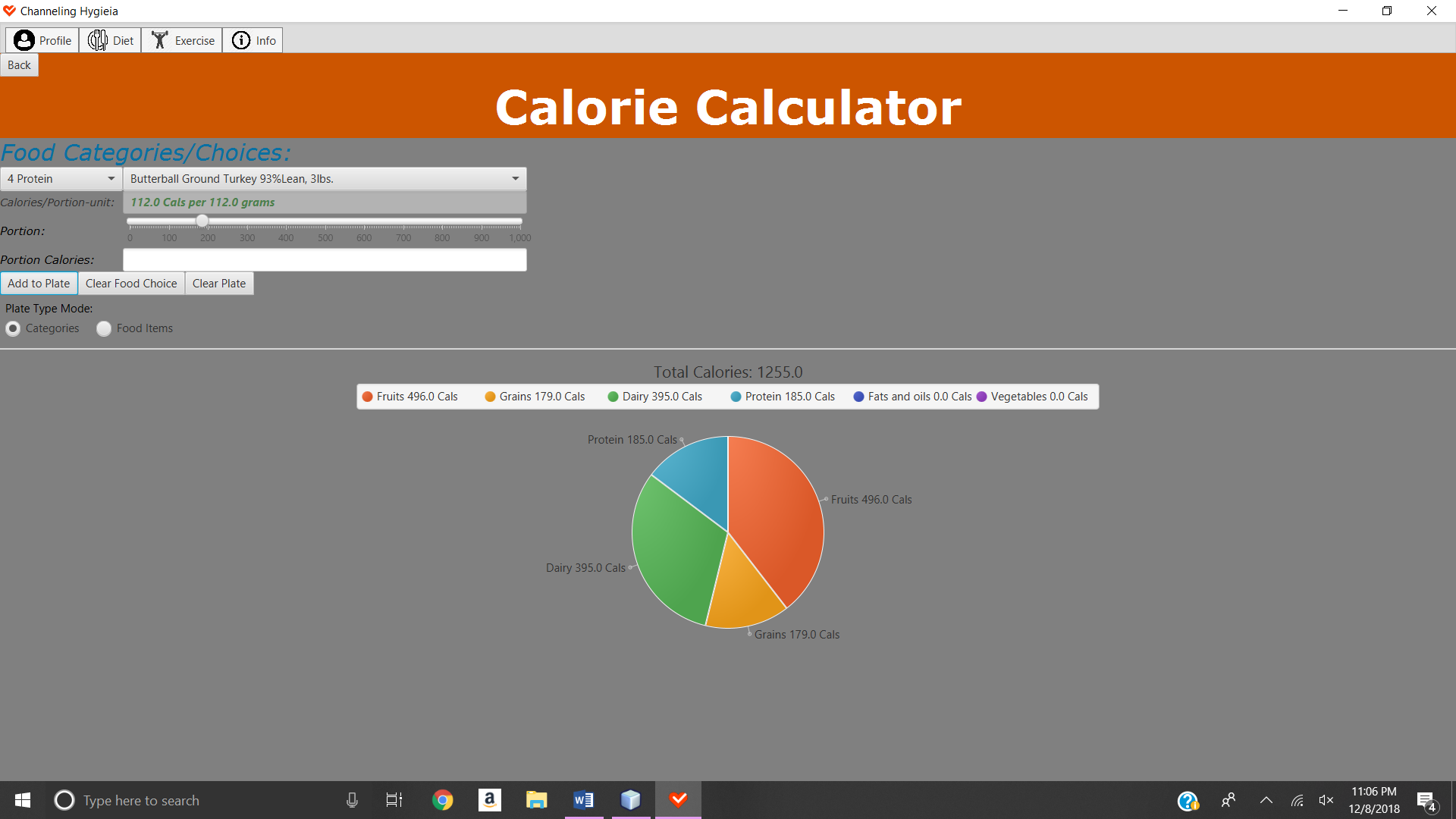


Figure C.3





**VI. Appendix D**

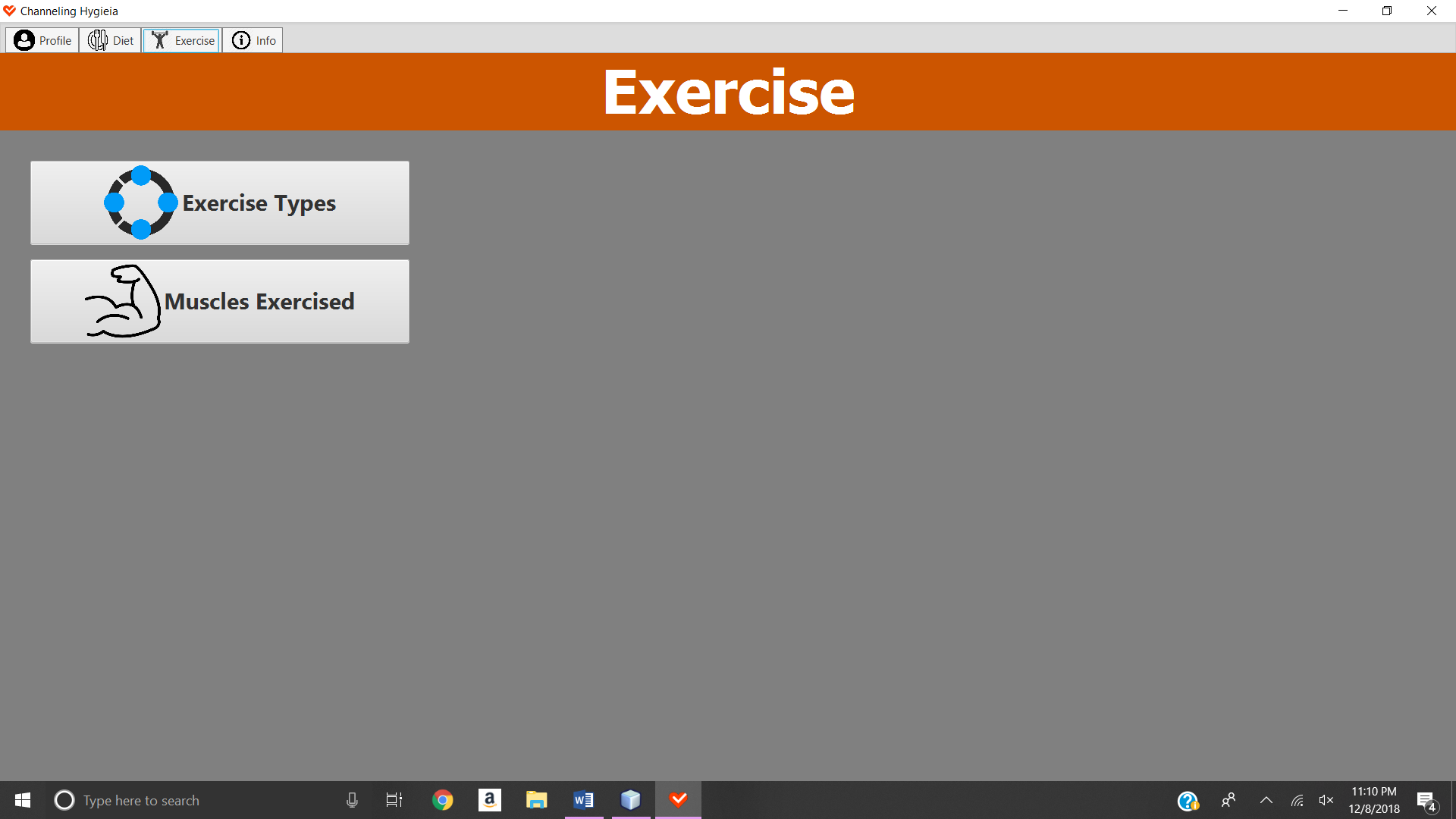


Figure D.1

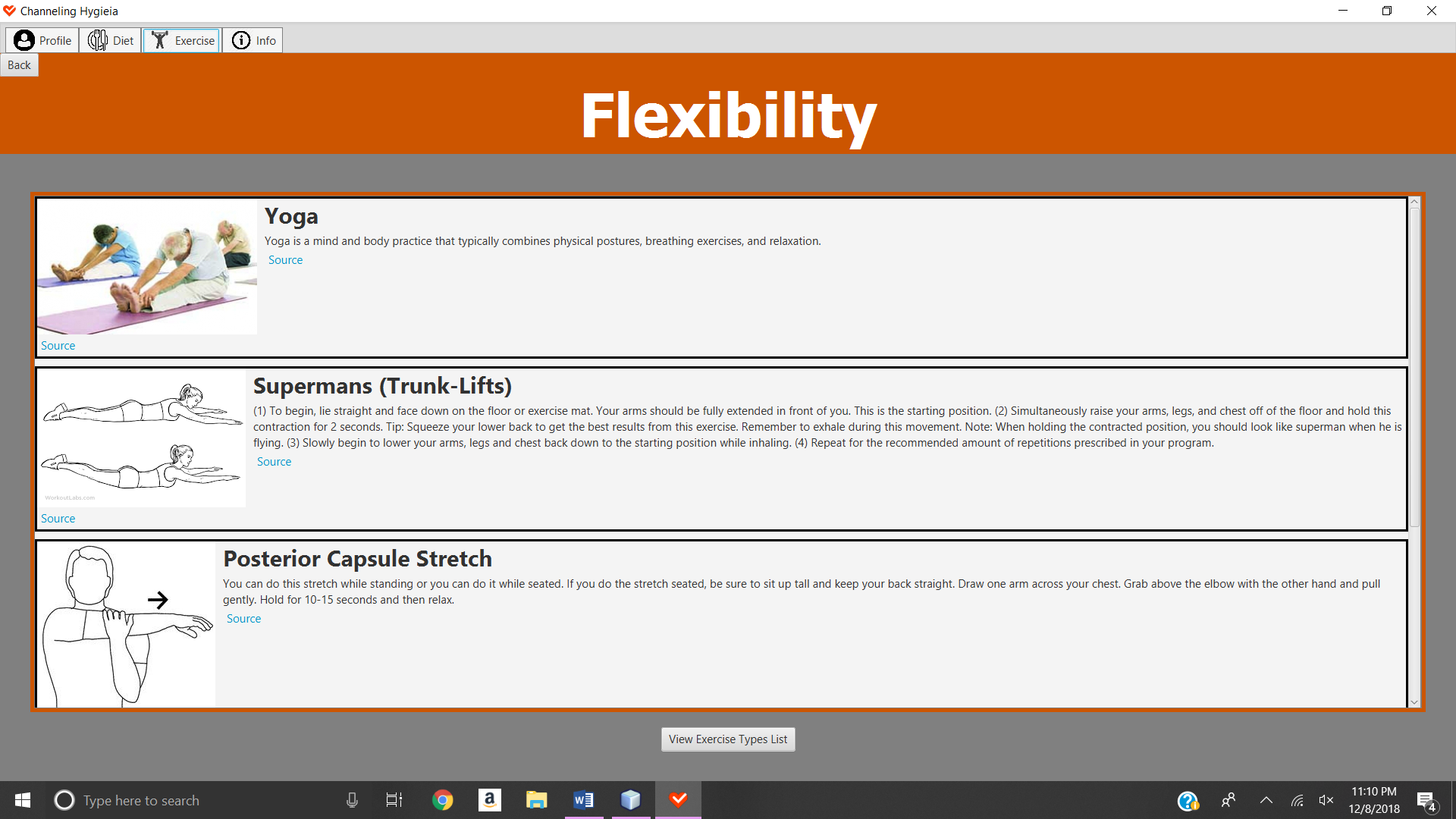
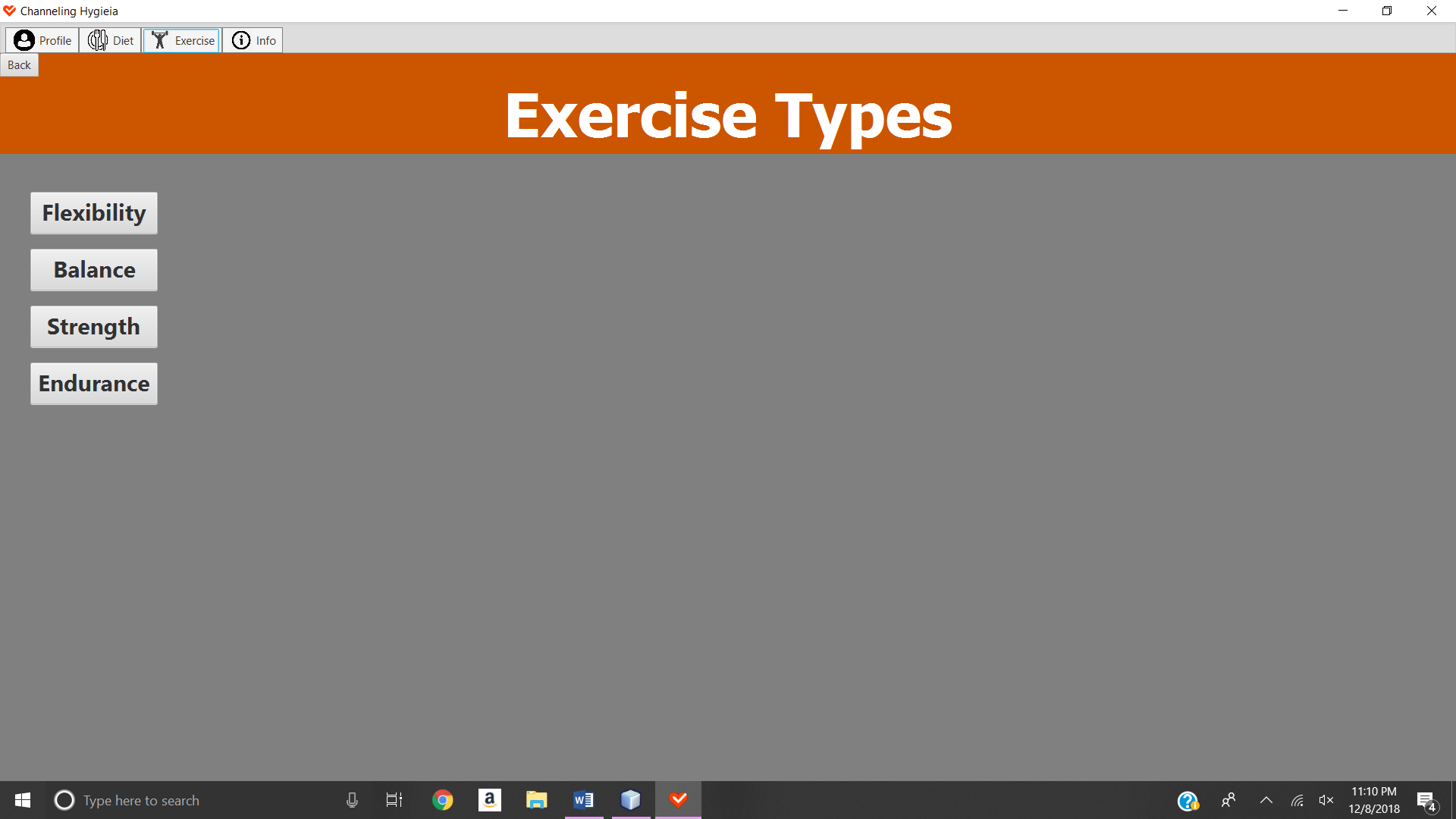
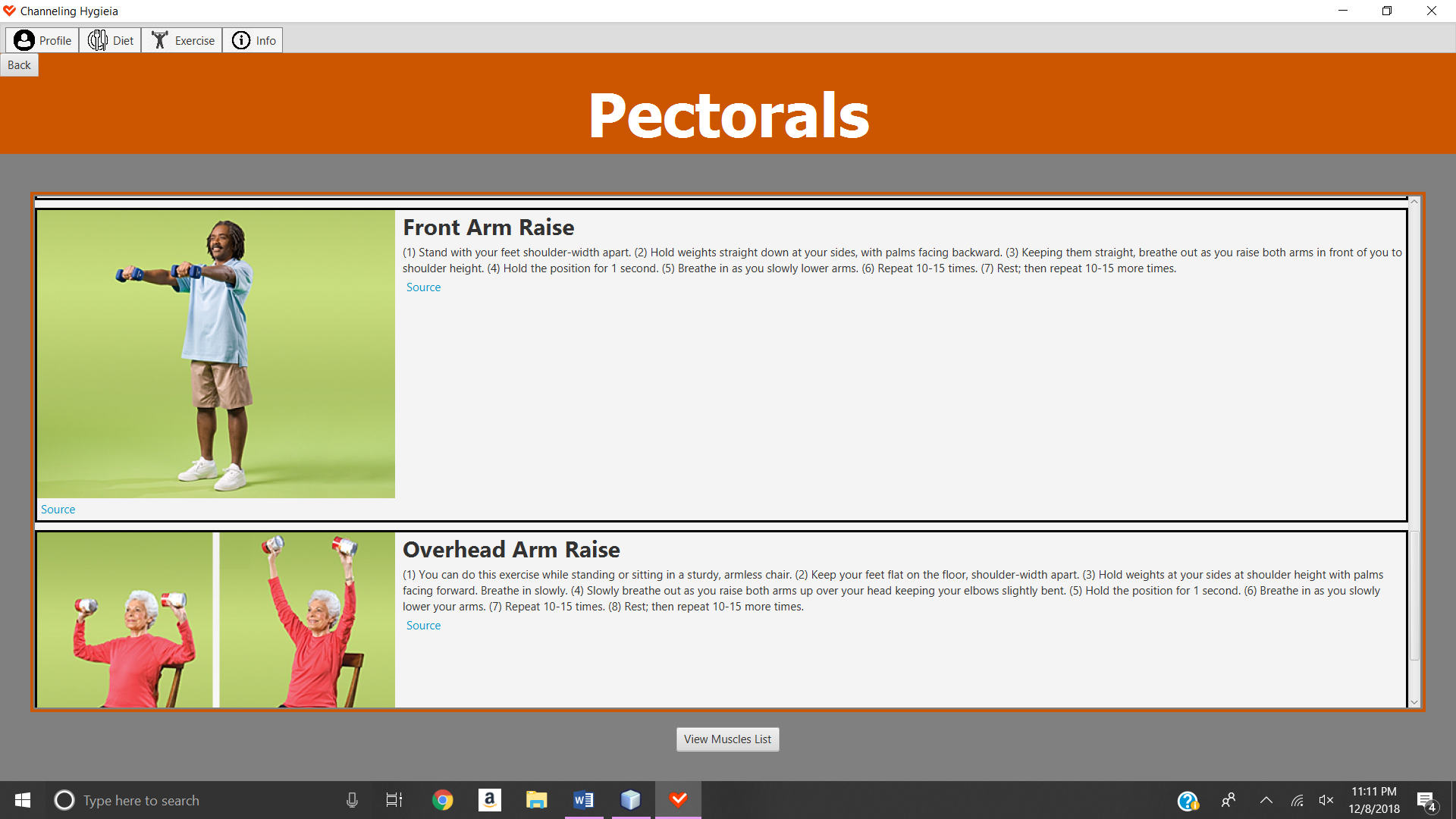
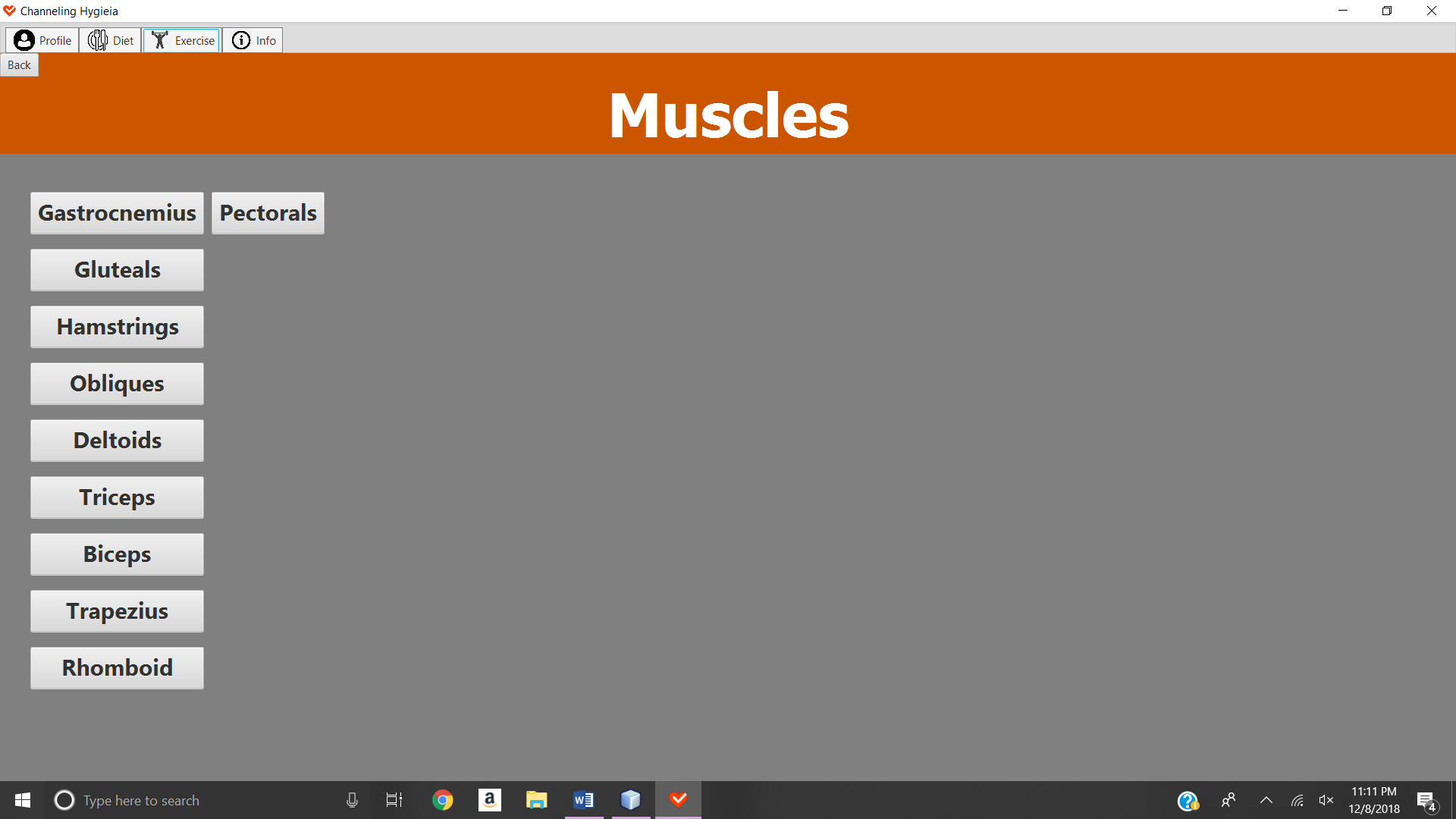
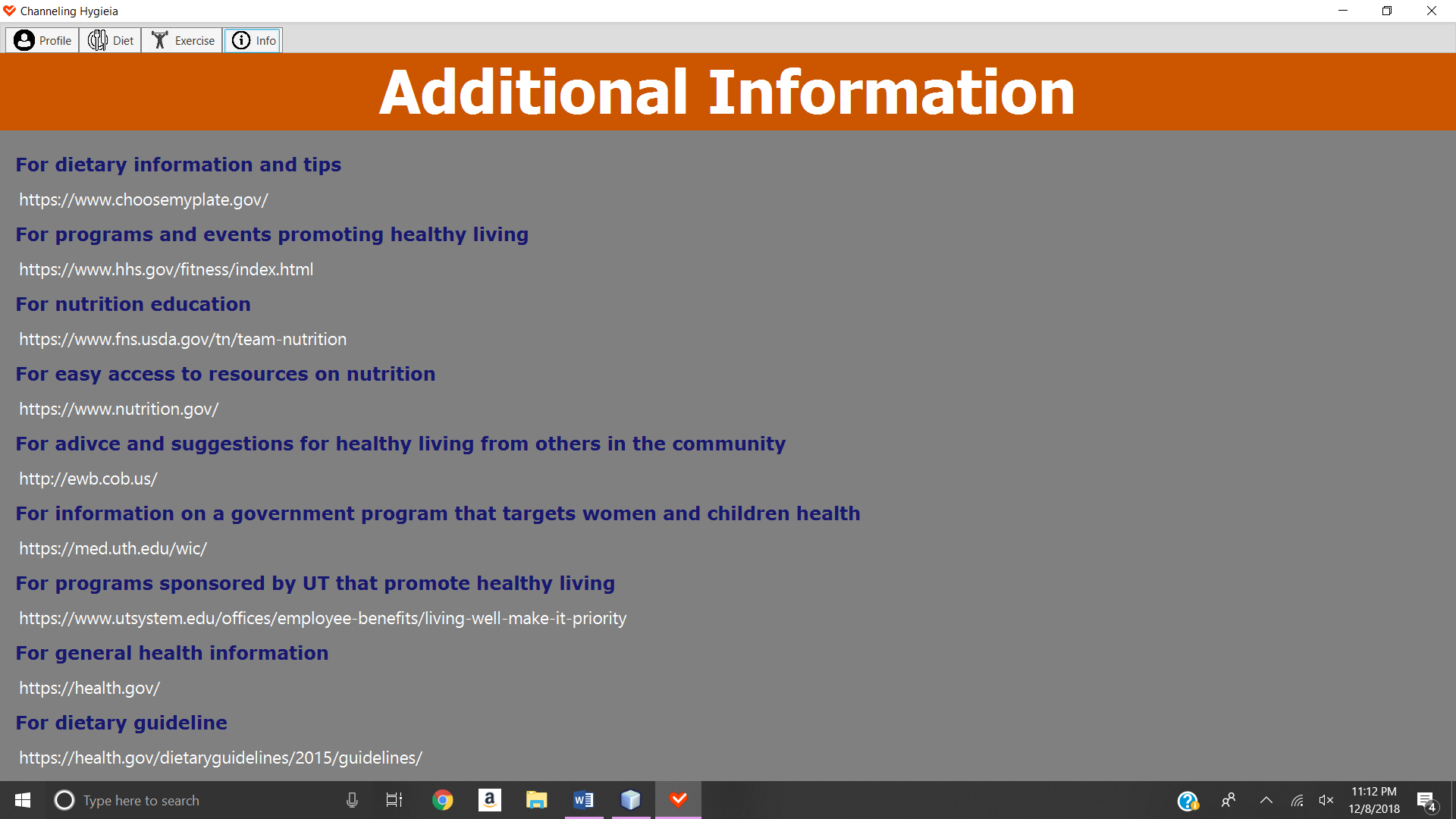


Figure D.2



**VI. Appendix E**



**VII. References**

1. Raymundovilano. (2017, December 14). Raymundovilano/SoftwareEngineeringFall2017Team2. Retrieved from <https://github.com/raymundovilano/SoftwareEngineeringFall2017Team2\> *Previous Group Project*
2. 13 Foods That Lower Blood Pressure. (n.d.). Retrieved from <https://www.healthline.com/health/foods-good-for-high-blood-pressure>

*Database - Disfood*

1. Appendix 1. Physical Activity Guidelines for Americans. (n.d.). Retrieved from <https://health.gov/dietaryguidelines/2015/guidelines/appendix-1/>

*Database - Recexer\_age table*

1. Appendix 2. Estimated Calorie Needs per Day, by Age, Sex, and Physical Activity Level. (n.d.). Retrieved from <https://health.gov/dietaryguidelines/2015/guidelines/appendix-2/>

*Database - Recfood\_f and Recfood\_m tables*

1. Best Diet for Gout: What to Eat, What to Avoid. (n.d.). Retrieved from <https://www.healthline.com/nutrition/best-diet-for-gout#section4>

*Database - Disfood*

1. Can eating certain foods help improve your cholesterol levels? (2018, July 17). Retrieved from <https://www.mayoclinic.org/diseases-conditions/high-blood-cholesterol/in-depth/cholesterol/art-20045192>

*Database - Disfood*

1. Celiac Disease Diet: Best Foods and Supplements | Everyday Health. (2018, March 26). Retrieved from <https://www.everydayhealth.com/celiac-disease/guide/diet/>

*Database - Disfood tables*

1. Harvard Health Publishing. (n.d.). 11 foods that lower cholesterol. Retrieved from <https://www.health.harvard.edu/heart-health/11-foods-that-lower-cholesterol>

*Database - Disfood*

1. Raymundovilano. (n.d.). Raymundovilano/SoftwareEngineeringFall2017Team2. Retrieved from [https://github.com/raymundovilano/SoftwareEngineeringFall2017Team2/blob/master/Final Project/vaqnutri\_personal.sql](https://github.com/raymundovilano/SoftwareEngineeringFall2017Team2/blob/master/Final%20Project/vaqnutri_personal.sql)

*Database - Personal table (Previous Group Project)*

1. Raymundovilano. (n.d.). Raymundovilano/SoftwareEngineeringFall2017Team2. Retrieved from <https://github.com/raymundovilano/SoftwareEngineeringFall2017Team2/blob/master/FinalProject/vaqnutri_profilepicture.sql>

*Database - ProfilePicture table (Previous Group Project)*

1. Raymundovilano. (n.d.). Raymundovilano/SoftwareEngineeringFall2017Team2. Retrieved from [https://github.com/raymundovilano/SoftwareEngineeringFall2017Team2/blob/master/Final Project/vaqnutri\_medical.sql](https://github.com/raymundovilano/SoftwareEngineeringFall2017Team2/blob/master/Final%20Project/vaqnutri_medical.sql)

*Database - Medical table (Previous Group Project)*

1. Raymundovilano. (n.d.). Raymundovilano/SoftwareEngineeringFall2017Team2. Retrieved from [https://github.com/raymundovilano/SoftwareEngineeringFall2017Team2/blob/master/Final Project/vaqnutri\_foods.sql](https://github.com/raymundovilano/SoftwareEngineeringFall2017Team2/blob/master/Final%20Project/vaqnutri_foods.sql)

*Database - Foods table (Previous Group Project)*

1. Raymundovilano. (n.d.). Raymundovilano/SoftwareEngineeringFall2017Team2. Retrieved from [https://github.com/raymundovilano/SoftwareEngineeringFall2017Team2/blob/master/Final Project/vaqnutri\_food\_groups.sql](https://github.com/raymundovilano/SoftwareEngineeringFall2017Team2/blob/master/Final%20Project/vaqnutri_food_groups.sql)

*Database - Food\_Groups table (Previous Group Project)*

1. Raymundovilano. (n.d.). Raymundovilano/SoftwareEngineeringFall2017Team2. Retrieved from [https://github.com/raymundovilano/SoftwareEngineeringFall2017Team2/blob/master/Final Project/vaqnutri\_disease.sql](https://github.com/raymundovilano/SoftwareEngineeringFall2017Team2/blob/master/Final%20Project/vaqnutri_disease.sql)

*Database - Disease table (Previous Group Project)*

1. Raymundovilano. (n.d.). Raymundovilano/SoftwareEngineeringFall2017Team2. Retrieved from [https://github.com/raymundovilano/SoftwareEngineeringFall2017Team2/blob/master/Final Project/vaqnutri\_allergy.sql](https://github.com/raymundovilano/SoftwareEngineeringFall2017Team2/blob/master/Final%20Project/vaqnutri_allergy.sql)

*Database - Allergy tables (Previous Group Project)*

1. Security Questions List 2. (n.d.). Retrieved from <https://www.pcninja.us/security-questions-choose-securely/securityquestions_2/>

*Database - Security\_questions table Code - CreateAccountView: Security Questions options*

1. The 16 Best Foods to Control Diabetes. (n.d.). Retrieved from <https://www.healthline.com/nutrition/16-best-foods-for-diabetics#section1>

*Database - Disfood tables*

1. Top 15 healthy foods for people with kidney disease. (n.d.). Retrieved from <https://www.davita.com/diet-nutrition/articles/advice/top-15-healthy-foods-for-people-with-kidney-disease>

*Database - Disfood tables*

1. 4 Types of Exercise. (2018, July 30). Retrieved from <https://go4life.nia.nih.gov/4-types-of-exercise/>
2. Harvard Health Publishing. (n.d.). The 4 most important types of exercise. Retrieved from <https://www.health.harvard.edu/exercise-and-fitness/the-4-most-important-types-of-exercise>
3. Palo Alto Medical Foundation. (n.d.). Kinds of Exercise. Retrieved from <http://www.pamf.org/teen/health/nutrition/kindsofexercise.html>
4. Josh, Traffics, S., David, & WLC. (n.d.). Home. Retrieved from <http://www.weight-lifting-complete.com/major-muscle-groups/>
5. Muscular Workouts. (n.d.). Retrieved from <https://fitness-for-the-busy-class.weebly.com/muscular-workouts.html>
6. The 13 Major Muscle Groups of the Body and their Functions. (n.d.). Retrieved from <https://www.bookmartialarts.com/news/major-muscle-groups-of-the-body>
7. Bogdan Rosu Creative. (n.d.). Circle, info icon [Circle with lowercase "i" in the middle]. Retrieved October 2, 2018, from <https://www.iconfinder.com/icons/351004/circle_info_icon>

*Additional Information (Info) Tab*

1. Diet Free Icon [Plate with fork and knife]. (n.d.). Retrieved October 2, 2018, from <https://pngtree.com/free-icon/diet_1308949>

*Diet Tab Icon*

1. D’origine, F. (2007, July 8). Fichier:Ubuntu Studio-icon-pd2.png [Now we have a complete set of Ubuntu-like public domain icons!]. Retrieved October 2, 2018, from <https://fr.wikipedia.org/wiki/Fichier:Ubuntu_Studio-icon-pd2.png>

*Exercise Tab - Types Button Icon*

1. Edit [Regular Style]. (n.d.). Retrieved October 2, 2018, from <https://fontawesome.com/icons/edit?style=regular&from=io>

*Home Tab - Edit Buttons Icon*

1. Food Pyramid Clip Art: Healthy-food-pyramid-10874047.jpg [Digital image]. (n.d.). Retrieved October 2, 2018, from <http://www.clipartpanda.com/clipart_images/healthy-food-pyramid-10874047-jpg-2562629>

*Diet Tab - Categories Button Icon*

1. Freepik. (n.d.). Muscular gymnast arm muscles show free icon [Digital image]. Retrieved October 2, 2018, from <https://www.flaticon.com/free-icon/muscular-gymnast-arm-muscles-show_30751>

*Exercise Tab - Muscles Button Icon*

1. IconBaandar Team. (n.d.). Dumbbells, exercise, gym, heavy, weight, weightlifter, workout ico [Silhouette lifting weights]. Retrieved October 2, 2018, from <https://www.iconfinder.com/icons/540747/dumbbells_exercise_gym_heavy_weight_weightlifter_workout_icon>

*Exercise Tab Icon*

1. Kripto. (n.d.). Magnifier, magnifying, search icon [Digital image]. Retrieved October 2, 2018, from <https://www.iconfinder.com/icons/286465/magnifier_magnifying_search_icon>

*Diet Tab - Search Button Icon*

1. Showalter, S. (n.d.). User profile [Person silhouette]. Retrieved October 2, 2018, from <https://thenounproject.com/term/user-profile/363633/>

*Login/Profile Tab Icon*

1. Skaryna, A. (n.d.). Business, calculate, calculation, calculator icon [Digital image]. Retrieved October 2, 2018, from <https://www.iconfinder.com/icons/2036064/business_calculate_calculation_calculator_icon>

*Diet Tab - Calorie Calculator Button Icon*

1. Quweider, Dr. (2018, December 08). Yessica-rodriguez/TestingMVCModelMultipleTabs. Retrieved from <https://github.com/yessica-rodriguez/TestingMVCModelMultipleTabs>
2. IbrahimLikeJava, & MikaelF. (2017, January 30). Open Image from FileChooser in JavaFX. Retrieved from <https://stackoverflow.com/questions/41927994/open-image-from-filechooser-in-javafx>

*Code - EditController: Upload Image*

1. MEMORYNOTFOUND. (2016, October 11). Calculate Age from date of birth in Java. Retrieved from <https://memorynotfound.com/calculate-age-from-date-of-birth-in-java/>

*Code - HomeController: Obtain User Age*

1. Raymundovilano. (2017, December 11). Raymundovilano/SoftwareEngineeringFall2017Team2. Retrieved from <https://github.com/raymundovilano/SoftwareEngineeringFall2017Team2/tree/master/Project/src/Environment/InformationTab>

*Code - Info Tab (Previous Group Project)*

1. Raymundovilano. (2017, December 11). Raymundovilano/SoftwareEngineeringFall2017Team2. Retrieved from <https://github.com/raymundovilano/SoftwareEngineeringFall2017Team2/tree/master/Project/src/Environment/Search>

*Code - Search (Previous Group Project)*

1. Raymundovilano. (2017, December 11). Raymundovilano/SoftwareEngineeringFall2017Team2. Retrieved from <https://github.com/raymundovilano/SoftwareEngineeringFall2017Team2/tree/master/Project/src/Environment/CalorieCalculator>

*Code - Calorie Calculator (Previous Group Project)*

1. SQL UPDATE DATE - javatpoint. (n.d.). Retrieved from <https://www.javatpoint.com/sql-update-date>

*Code - EditModel: Personal Information*

1. SQLite Java: Write and Read BLOB. (n.d.). Retrieved December 2, 2018, from <http://www.sqlitetutorial.net/sqlite-java/jdbc-read-write-blob/>

*Code - EditModel: Update User's Profile Image*

1. R