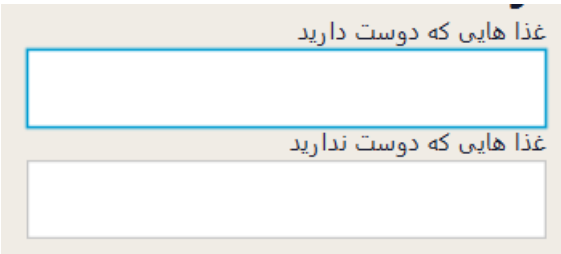
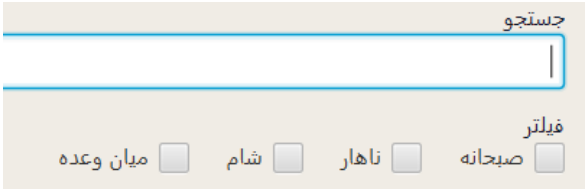
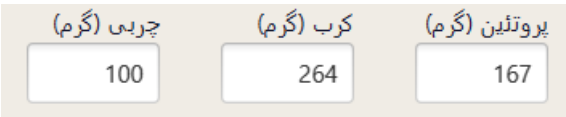
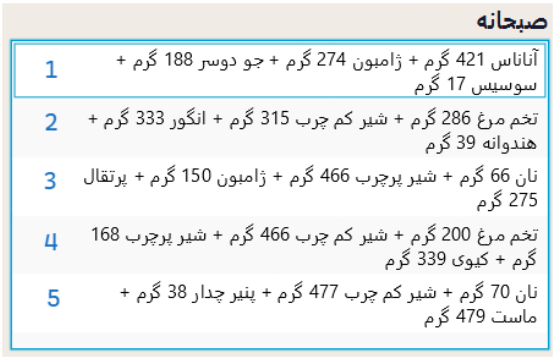
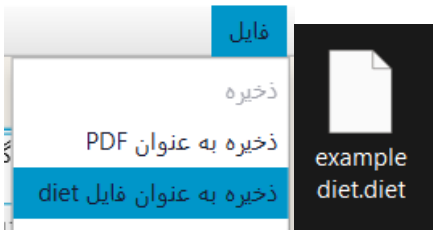
Criterion E: Evaluation

** Results obtained based on Unit, Integration, and Dry-Run testing. Throughout file, client comments from interview 3 (appendix) have been considered.*

= met/client satisfied *= partially met/satisfied* *= not met/satisfied*

No.	Success Criteria	Result
1	Input allowing user to choose food preferences (likes, dislikes) from list of aliments.	Clicking on preference inputs opens *alimentSelectorWindow* 
2	When choosing foods from the list, there should be search and filter options. User should be able to search for desired food or filter undesirable ones.	*alimentSelectorWindow* has search and meal filter options. 
3	input to manually enter calorie target or use built-in calculator to automatically calculate target.	*goalsCalculatorWindow* opens. Given input: Male, 25, 180 cm, 65 kg, 4-5 times weekly exercise, maintain weight Calorie output: 2,425 Online calculator output[1]: 2,425
4	input for Macronutrient in grams. calories of specified macronutrients must add up to calorie target.	Inputs exist on *dietCreationWindow*. Text input is not allowed. If macronutrient calories don't match specified target, warning is displayed 

[1] https://www.calculator.net/calorie-calculator.html

		
5	option allowing user to add new aliments to datafile	New aliments can be added using *addNewAlimentWindow*. Added aliment appears in *alimentSelectorWindow* search list.
6	option to remove aliments from datafile.	Right-clicking aliments in *alimentSelectorWindow* and pressing “delete” button removes it from datafile.
7	Output showing 5 meal options for each breakfast, lunch, snacks, and dinner.	5 meal options are displayed for each meal in *dietDisplayWindow* 
8	Ability to Modify foods in the generated meal options.	Right-clicking meal options in *dietDisplayWindow* and pressing “edit” opens *mealOptionEditorWindow* where user can add/remove aliments
9	Ability to view an aliment’s nutritional data.	Right-clicking aliment from *alimentSelectorWindow* and pressing “view information” button opens *viewAlimentInfoWindow* which displays the information.
10	Possibility to save generated diets for future use	Clicking “file” → “save as diet file” in *dietDisplayWindow* saves as diet file. 

11	possibility to access previously saved diets	In *startingWindow*, pressing “open diet file” opens file chooser where user can open diet file. It is displayed in *dietDisplayWindow*
12	Possibility to save diets as PDFs	Clicking “file” → “save as PDF” 
13	Diet should contain user preferences	generated diet largely follows user preference. Although it may occasionally include a disliked food.  
14	Diet calories and macronutrients should be close to the input amount	Some meal options are within calorie range. However, there are others completely out of range. E.g., in one test run, snack meal options had the following calories: 1. 763 2. 246 3. 232 4. 641 5. 230 green = in-range, red = out of range.

Recommendations for Further Development (with regards to Client comments in appendix interview 3)

Major improvements

Meal options out of range: As mentioned in success criteria, some meal options fall outside appropriate calorie range. This is because the genetic algorithm in rare cases doesn't converge and after 10,000 max iterations returns a bad meal option.

Repetition: some meal options contain repeated aliments and are like each other. This decreases variety within diet.

Insignificant aliments: sometimes meal options contain trivial aliments. E.g., 5 grams of apple. It would be better to not include those altogether.

Moderate improvements

Limited aliments: the datafile contains limited number of aliments. This can be improved by time as client adds more aliments.

Odd recommendations: client mentioned some meal option recommendations are odd, although they can be modified, it would be better to improve algorithm to generate better meal options.

Word count (*excluding headings and footers*): 521