04 May 2020

1. Personal information

Uni-Canteen
Angeline Oktaviana Eka Pratiwi Jayanegara
706977
Aalto Bachelor’s Programme in Science and Technology – Data Science
Second year of studies

2. General description

Uni-Canteen is a scala application that gathers the daily menus of student restaurants in Otaniemi and displays them in a centralized manner (cf. Kanttiinit / ruokalistat.net). The work will include 5 restaurants (A-bloc, Alvari, Studio Kipsari, Täffä, and Väre).

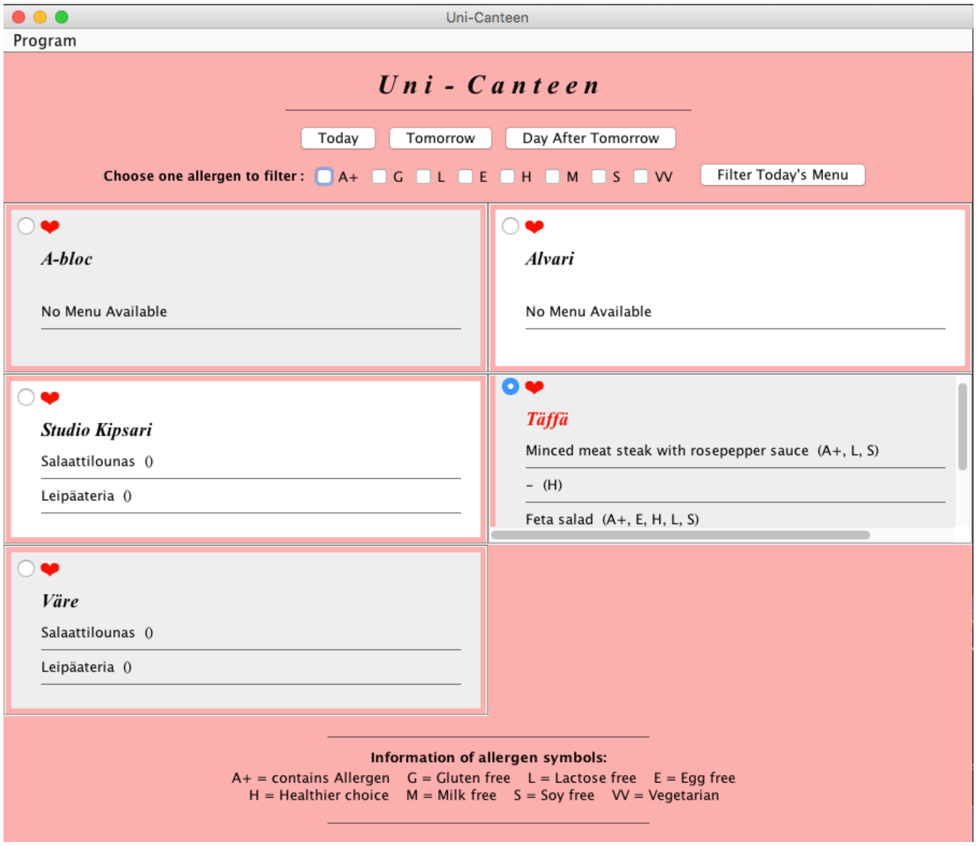
Additional features in the program:

- Support for choice of favourite restaurants.
- Filter menu items based on allergen
- Inform users about menu items that contain allergens (GUI)

In the end, there are only slight changes compared to the original general plan but overall it remains the same as the plan. The program has been implemented with GUI which is an intermediate requirement of the project.

3. User Interface

The user interface of Uni-Canteen is rather simple and user friendly. The application is run from the UniApp scala file. *MenuBar* is located on the left top as “Program” and it has one menu item “Quit” to exit the program. On top of the GUI there is the “Uni-Canteen” title. Below the title there are three buttons (*Today, Tomorrow,* and *Day After Tomorrow*). Every button has the same function to get the menu for each day (today, tomorrow, or day after tomorrow). If the button is clicked, the menu panels will be refreshed and menu panels will show the designated menu lists. Below the three buttons there are eight allergens checkboxes, and one *Filter Today’s Menu* button. Users can check one allergen from the checkboxes and then click the filter menu’s button to proceed with the feature. If users would like to change the allergen filter, uncheck the ticked checkbox then check the desired allergen checkbox. Radio button is shown on top of every restaurant’s name with a heart icon. It is to show if a user favorites the restaurant, the user can select the radio button and the restaurant’s name will be highlighted by color change (red - as shown in Figure 1).



Uni-Canteen

Program

Uni - Canteen

Today Tomorrow Day After Tomorrow

Choose one allergen to filter : □ A+ □ G □ L □ E □ H □ M □ S □ VV Filter Today's Menu

○ ❤ *A-bloc*

No Menu Available

○ ❤ *Alvari*

No Menu Available

○ ❤ *Studio Kipsari*

Salaattilounas ()

Leipäateria ()

● ❤ *Täffä*

Minced meat steak with rosepepper sauce (A+, L, S)

– (H)

Feta salad (A+, E, H, L, S)

○ ❤ *Väre*

Salaattilounas ()

Leipäateria ()

Information of allergen symbols:

A+ = contains Allergen G = Gluten free L = Lactose free E = Egg free
H = Healthier choice M = Milk free S = Soy free VV = Vegetarian

Figure 1. Graphical User Interface of Uni-Canteen

4. Program Structure

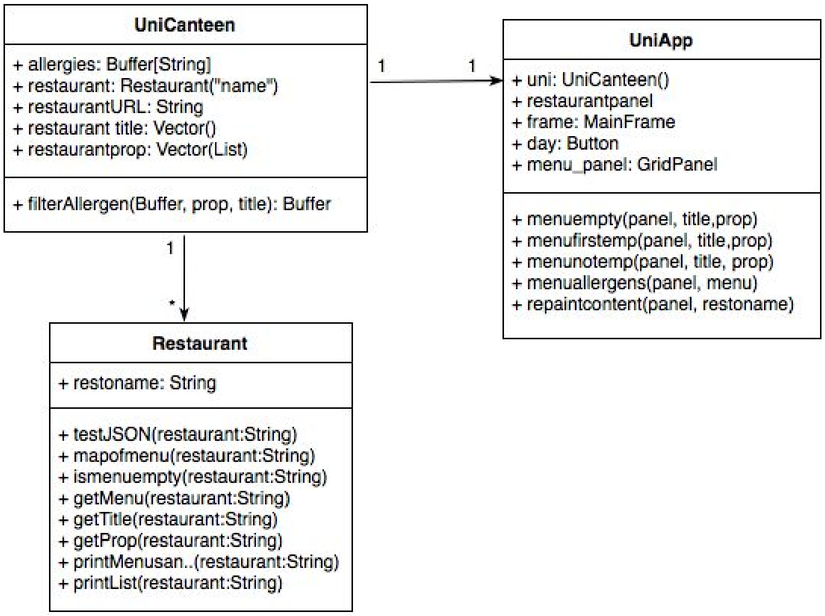

UniCanteen
+ allergies: Buffer[String]
+ restaurant: Restaurant("name")
+ restaurantURL: String
+ restaurant title: Vector()
+ restaurantprop: Vector(List)
+ filterAllergen(Buffer, prop, title): Buffer
1
1
UniApp
+ uni: UniCanteen()
+ restaurantpanel
+ frame: MainFrame
+ day: Button
+ menu_panel: GridPanel
+ menuempty(panel, title,prop)
+ menufirstemp(panel, title,prop)
+ menunotemp(panel, title, prop)
+ menuallergens(panel, menu)
+ repaintcontent(panel, restoname)
1
*
Restaurant
+ restoname: String
+ testJSON(restaurant:String)
+ mapofmenu(restaurant:String)
+ ismenuempty(restaurant:String)
+ getMenu(restaurant:String)
+ getTitle(restaurant:String)
+ getProp(restaurant:String)
+ printMenusan..(restaurant:String)
+ printList(restaurant:String)

Figure 2. UML of Uni-Canteen

There are 3 classes for this Uni-Canteen: *UniCanteen, Restaurant*, and *UniApp*.

UniCanteen is the main class of the program. *UniCanteen* class connects all the classes. There are all of the URLs needed from the restaurant website stored in variables, also each restaurants' menu titles and properties stored in variables. *FilterAllergen* function takes input from the allergen checkbox that is picked, also menu titles and properties for each restaurant to be filtered. This function returns a buffer of string.

Restaurant class is where the data is processed and parsed according to needs. *Mapofmenu* function is where the data first parsed, the function returns a list of maps for menus which include day and courses. *IsMenuEmpty* function checks whether the menu list of the restaurant is empty. Next the data is parsed in the *getMenu* function that returns a list of maps of titles and properties. From here, there are two functions available for further data processing which are *getTitle* that returns vector of titles of menus and *getProp* which returns vector of lists of allergens. There are some functions that are available only for testing purposes (*testJSON* that test if the data is JSON*, printMenusAndProp* which prints the menu titles also properties and last function is *printList* which prints all components of the list).

UniApp class is GUI class of Uni-Canteen program. It is connected with the *UniCanteen* class. *UniApp* has variables of the restaurant panels that are included in the *MainFrame*. All needed contents and functions are located in the *MainFrame* of GUI including all the buttons, panels, checkboxes, and radiobuttons. The functions in *UniApp* class are mostly functions to reduce the repetition in the class.

5. Algorithms

The algorithm used for filtering the allergen is using predicate (input from the user) from the checkbox and filter button in GUI, then further matches allergen with the restaurant menus by looping the menu allergen properties. If the menu contains the allergen that the user ticked, then the menu is added to a buffer of *filteredmenu*. This algorithm might be implemented differently if there are more than one allergen chosen.

6. Data Structures

All of the data is obtained using JSON from the sources available for the restaurants menus. The allergens and filtered data of the lunch list are stored in *Mutable Buffers,* which is an easy structure for handling mutable collections. The normal lunch list titles and properties, which are immutable lists of units, are stored in *Vectors*.

7. Files and Internet Access

The program is implemented by accessing real time data of the restaurants website, thus it does not deal with any files. The data is obtained from *kanttiinit* website in form of JSON data.

8. Testing

First test is by reading the data from the URL in the restaurant class using *testJSON* function. This function will return true if the data is valid JSON data and it is able to be parsed. If the data has an empty list of courses (which is checked with *isMenuEmpty* function), the program will return "No Menu Available" on the restaurant panel in GUI. Project is also tested by comparing whether the data provided and the lunch list on the app are synchronised. The method *filterAllergen* should return the filtered list of menu which the user has chosen as a parameter, for example, if the user chooses lactose free, the method will then take all the food with symbol "L" that is a lactose free symbol and returns the list of that lactose free menu. The GUI is tested by trying to run the program and see if it works accordingly.

9. Known Bugs and Missing Features

There is no significant bug in the project that is known, however, if the user ticks more than one allergen checkboxes and filters the menu list, it might not work accordingly, as it is not how the program is implemented. The *Filter Today's Menu* feature is also applied only for today's menu, thus if the user chooses tomorrow or day after tomorrow menus and tries to filter the allergens, it will still give the user the today's filtered menu. If the project is developed further, the features could be refined or developed according to how the users desire the program to work.

10. Three Best Sides and Three Weaknesses

The best sides of the program are the design and simplicity of the GUI outcome. The GUI is simple and very user friendly. Users do not need to study the program to be able to use it and all of the information is provided and shown well in the GUI. The design of the GUI is also modern which stands out from typical basic GUI.

The weakness of the program might be the lack of features provided. The program is designed to be simple, however some users might want to have more features to play with. The code writing could be also improved to be more clean and neat.

11. Deviations from the plan, realized process and schedule

The project is implemented in 70+ hours. Due to unforeseen circumstances along the way while doing the project, one week extension was needed to complete this project. During the process time management is learned. Investing enough time for the project should have been done to finish the project according to the deadline.

12. Final Evaluation

Uni-Canteen is a simple and user-friendly app that provides five university restaurants' menu lists daily (today, tomorrow and day after tomorrow). The app also provides features to filter allergen according to user input. The allergens information is well shown on the GUI. More features could be implemented to refine the program. The more minimalistic JSON library could be used to work with the data easier. If the project could be started again, the code writing will be focused so that it is more neat and clear. Time will be also invested more so that the project is developed further.

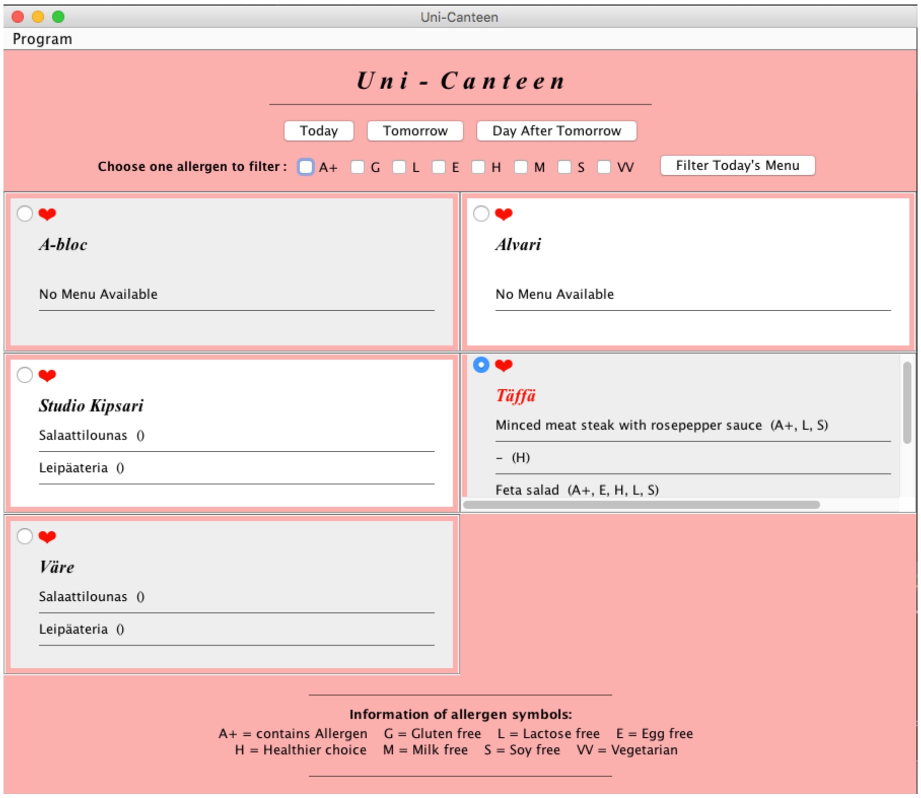
13. References

- Alvinalexander.com. (2020). *How to use the 'filter' method to filter a Scala collection | alvinalexander.com*. [online] Available at:

https://alvinalexander.com/scala/how-to-use-filter-method-scala-collections-cookbook [Accessed 14 Feb. 2020].

- Anon, (2020). [ebook] Available at: https://cordis.europa.eu/docs/projects/cnect/5/215455/080/deliverables/ROADIDEA-D3-1-Data-filtering-methods-V1-1.pdf [Accessed 14 Feb. 2020].
- Kanttiinit.fi. (2020). *Kanttiinit*. [online] Available at: https://kanttiinit.fi [Accessed 14 Feb. 2020].
- Otfried.org. 2020. *Scala Documentation -- GUI Programming*. [online] Available at: <http://otfried.org/scala/gui.html> [Accessed 4 May 2020].
- YouTube. 2020. *Mark Lewis*. [online] Available at: <https://www.youtube.com/channel/UCEvjiWkK2BoIH819T-buioQ> [Accessed 4 May 2020].
- Cs.helsinki.fi. 2020. *Code Examples For Programming In Scala*. [online] Available at: <https://www.cs.helsinki.fi/u/wikla/OTS/Sisalto/examples/html/ch32.html> [Accessed 4 May 2020]
- App.diagrams.net. 2020. *Flowchart Maker & Online Diagram Software*. [online] Available at: <https://app.diagrams.net> [Accessed 4 May 2020].

14. Appendixes



Choose one allergen to filter : □ A+ □ G □ L □ E □ H □ M □ S □ VV

○ ❤ *A-bloc*

No Menu Available

○ ❤ *Alvari*

No Menu Available

○ ❤ *Studio Kipsari*

Salaattilounas ()

Leipäateria ()

● ❤ *Täffä*

Minced meat steak with rosepepper sauce (A+, L, S)

– (H)

Feta salad (A+, E, H, L, S)

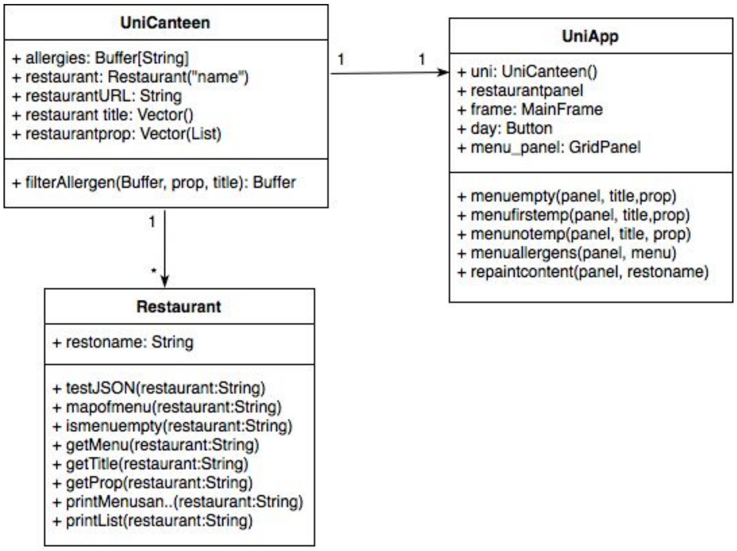
○ ❤ *Väre*

Salaattilounas ()

Leipäateria ()

Information of allergen symbols:
A+ = contains Allergen G = Gluten free L = Lactose free E = Egg free
H = Healthier choice M = Milk free S = Soy free VV = Vegetarian

Figure 1. Graphical User Interface of Uni-Canteen with Täffä as favorite restaurant



UniCanteen

+ allergies: Buffer[String]
+ restaurant: Restaurant("name")
+ restaurantURL: String
+ restaurant title: Vector()
+ restaurantprop: Vector(List)

+ filterAllergen(Buffer, prop, title): Buffer

UniApp

+ uni: UniCanteen()
+ restaurantpanel
+ frame: MainFrame
+ day: Button
+ menu_panel: GridPanel

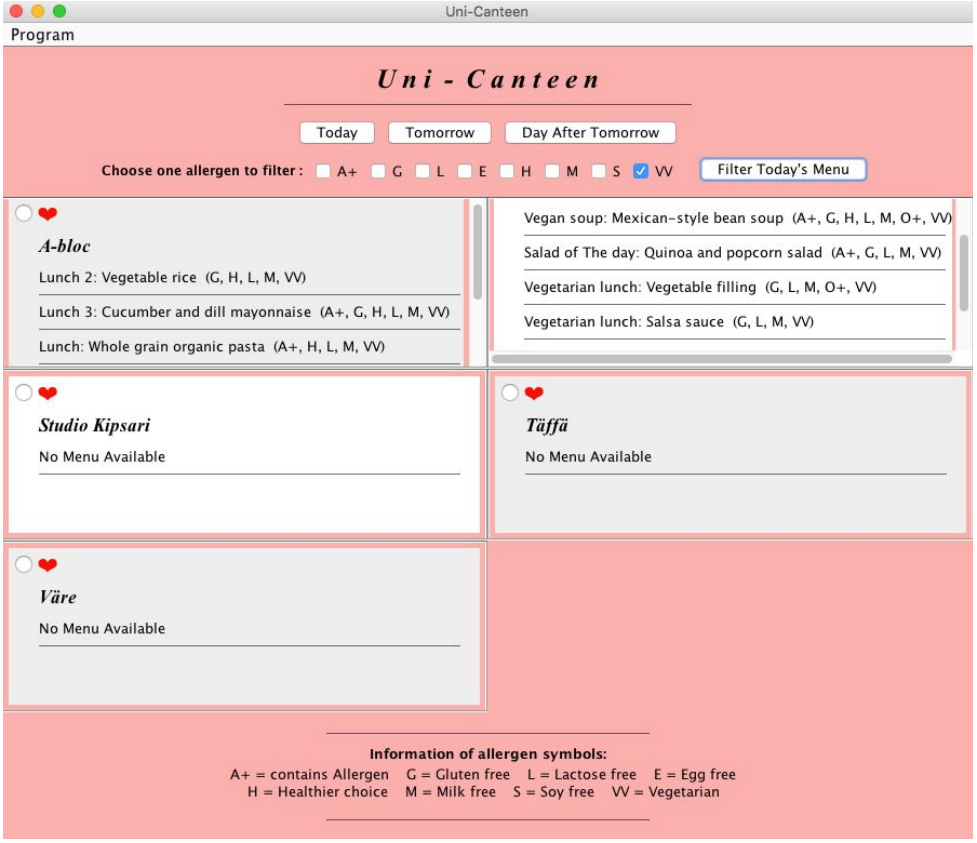
+ menuempty(panel, title,prop)
+ menufirstemp(panel, title,prop)
+ menunotemp(panel, title, prop)
+ menuallergens(panel, menu)
+ repaintcontent(panel, restoname)

Restaurant

+ restoname: String

+ testJSON(restaurant:String)
+ mapofmenu(restaurant:String)
+ ismenuempty(restaurant:String)
+ getMenu(restaurant:String)
+ getTitle(restaurant:String)
+ getProp(restaurant:String)
+ printMenusan..(restaurant:String)
+ printList(restaurant:String)

Figure 2. UML of Uni-Canteen



Choose one allergen to filter : □ A+ □ G □ L □ E □ H □ M □ S ☑ VV

○ ❤ *A-bloc*

Lunch 2: Vegetable rice (G, H, L, M, VV)

Lunch 3: Cucumber and dill mayonnaise (A+, G, H, L, M, VV)

Lunch: Whole grain organic pasta (A+, H, L, M, VV)

Vegan soup: Mexican-style bean soup (A+, G, H, L, M, O+, VV)

Salad of The day: Quinoa and popcorn salad (A+, G, L, M, VV)

Vegetarian lunch: Vegetable filling (G, L, M, O+, VV)

Vegetarian lunch: Salsa sauce (G, L, M, VV)

○ ❤ *Studio Kipsari*

No Menu Available

○ ❤ *Täffä*

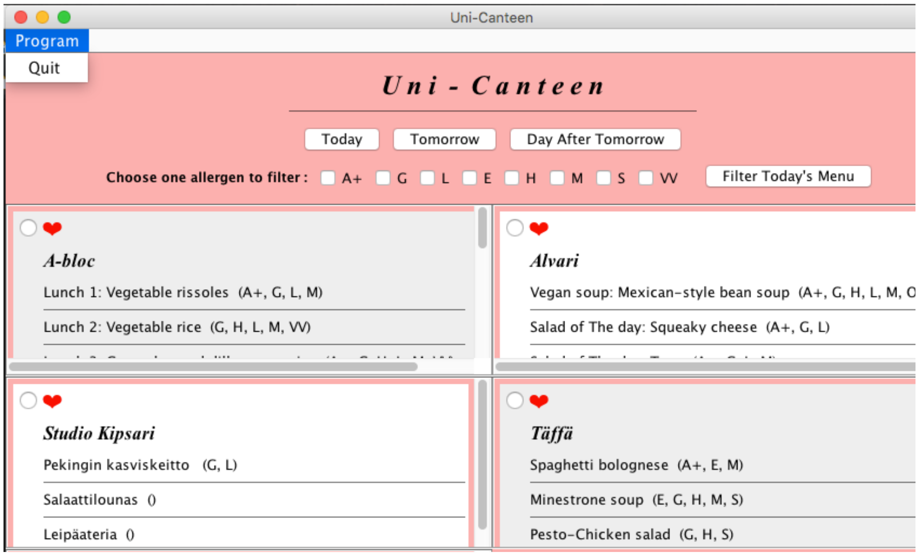
No Menu Available

○ ❤ *Väre*

No Menu Available

Information of allergen symbols:
A+ = contains Allergen G = Gluten free L = Lactose free E = Egg free
H = Healthier choice M = Milk free S = Soy free VV = Vegetarian

Figure 3. Graphical User Interface of Uni-Canteen filters VV (Vegetarian) menu items.



Choose one allergen to filter : □ A+ □ G □ L □ E □ H □ M □ S □ VV

○ ❤ *A-bloc*

Lunch 1: Vegetable rissoles (A+, G, L, M)

Lunch 2: Vegetable rice (G, H, L, M, VV)

○ ❤ *Alvari*

Vegan soup: Mexican-style bean soup (A+, G, H, L, M, O

Salad of The day: Squeaky cheese (A+, G, L)

○ ❤ *Studio Kipsari*

Pekingin kasviskeitto (G, L)

Salaattilounas ()

Leipäateria ()

○ ❤ *Täffä*

Spaghetti bolognese (A+, E, M)

Minestrone soup (E, G, H, M, S)

Pesto-Chicken salad (G, H, S)

Figure 4. Graphical User Interface of Uni-Canteen showing Menu item Quit to exit the program