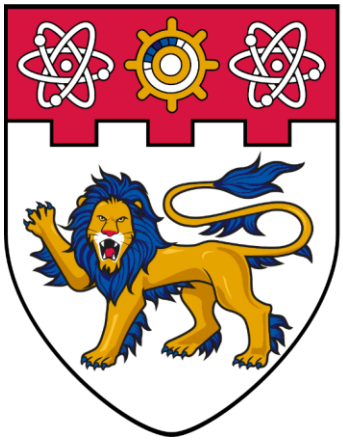

NANYANG TECHNOLOGICAL UNIVERSITY
SINGAPORE

CZ3005
Artificial Intelligence

Lab Exercise 4: Implementing a Talking Box with Prolog

DSAI

U1821610C

Liew Zhi Li (Sherna)

Contents

1 – Q3 Subway Sandwich Interactor (SSI)

The Prolog script offers different meal options, sandwich options, meat options, salad options, sauce options, top-up options, sides options etc. to create a customized list of person's choice. The options should be intelligently selected based on previous choices. For example, if the person chose a veggie meal, meat options should not be offered. If a person chose healthy meal, fatty sauces should not be offered. If a person chose vegan meal, cheese top-up should not be offered. If a person chose value meal, no top-up should be offered.

1.1 – Subway Talking Bot

In this assignment, I have implemented a talking box using Prolog called the Subway Talking Bot. The idea is that the bot will converse with the user in order to simulate the experience of placing an order at Subway. The bot will ask a series of questions similar to how an employee would ask the customer questions to create an order. An application of this could be a chat bot on the web browser.

1.2 – Design of Solution

The Prolog codes are contain in the Prolog file: LiewZhiLi_qn_3.pl.

In Line 4, I first defined 12 dynamic predicates using the keyword 'dynamic', that will be used to store the user's choice for an order. They are: meal, bread, size, main, cheese, veg, sauce, side, drink, cookie, dineOption and paymentMethod. All of these dynamic predicates have a '/1', meaning that they accept one parameter.

```
:- dynamic meal/1, bread/1, size/1, main/1, cheese/1, veg/1, sauce/1, side/1
, drink/1, cookie/1, dineOption/1, paymentMethod/1.
```

In Lines 9 – 23, I define the facts of all the list of menu items under each category. Although the order of definition of facts and rules is not important, a good rule of thumb is to define facts before rules.

```
meals([standard, vegetarian, healthy, value, salad_bowl]).
breads([multigrain, hearty_italian, italian_wheat, parmesan_oregano, honey_o
at, flat_bread]).
sizes([six_inch, foot_long]).
mains([chicken, ham, turkey, tuna, beef, meatball_marinara, egg_mayo, italia
n_bmt, veggie_delite, veggie_patty, tofu_patty]).
veg_mains([veggie_delite, veggie_patty, tofu_patty]).
cheeses([american, monterey, feta, mozzarella, cheddar, swiss, '0 (none)']).
vegs([cucumbers, tomatoes, lettuce, green_peppers, red_onions, jalapenos, bl
ack_olives, pickles, '0 (none)']).
sauces([chipotle, ranch, bbq, chilli, tomato, mayonnaise, vinegar, '0 (none)
']).
healthy_sauces([chipotle, ranch, tomato, vinegar, '0 (none)']).
sides([potato_chips, hashbrowns, yogurt, '0 (none)']).
drinks([coke, sprite, orange_fanta, mineral_water, orange_juice, green_tea,
hot_coffee, hot_tea, '0 (none)']).
healthy_drinks([diet_coke, diet_sprite, mineral_water, green_tea_no_sugar, h
ot_coffee_no_sugar, hot_tea_no_sugar, '0 (none)']).
```

```
cookies([choco_chip, double_choco_chip, peanut_butter, white_chip_macadamia_
nut, oatmeal_raisin, raspberry, '0 (none)']).
dineOptions([dine_in, takeaway, delivery]).
paymentMethods([cash, credit_card, nets]).
```

In Lines 27 – 41, I defined the picked/2 predicates for each menu item. For example, X is picked in meals if L is in meals and X is a member of L (meals). These picked predicates will be used later in pick_meal, pick_bread, pick_size, and so on.

```
picked(X, meals) :- meals(L), member(X, L).
picked(X, breads) :- breads(L), member(X, L).
picked(X, sizes) :- sizes(L), member(X, L).
picked(X, mains) :- mains(L), member(X, L).
picked(X, veg_mains) :- veg_mains(L), member(X, L).
picked(X, cheeses) :- cheeses(L), member(X, L).
picked(X, vegs) :- vegs(L), member(X, L).
picked(X, sauces) :- sauces(L), member(X, L).
picked(X, healthy_sauces) :- healthy_sauces(L), member(X, L).
picked(X, sides) :- sides(L), member(X, L).
picked(X, drinks) :- drinks(L), member(X, L).
picked(X, healthy_drinks) :- healthy_drinks(L), member(X, L).
picked(X, cookies) :- cookies(L), member(X, L).
picked(X, dineOptions) :- dineOptions(L), member(X, L).
picked(X, paymentMethods) :- paymentMethods(L), member(X, L).
```

In Lines 44 – 58, I defined all the predicates to display all the menu item options for each category. For example, when display_options(meals) is called if L is in meals, we write to output 'Meals: {' , and use predicate item_list(L).

```
display_options(meals):- meals(L), write('Meals: {'), item_list(L).
display_options(breads):- breads(L), write('Breads: {'), item_list(L).
display_options(sizes):- sizes(L), write('Sizes: {'), item_list(L).
display_options(mains):- mains(L), write('Mains: {'), item_list(L).
display_options(veg_mains):- veg_mains(L), write('Veg Mains: {'), item_list(
L).
display_options(cheeses):- cheeses(L), write('Cheeses: {'), item_list(L).
display_options(vegs):- vegs(L), write('Vegetables: {'), item_list(L).
display_options(sauces):- sauces(L), write('Sauces: {'), item_list(L).
display_options(healthy_sauces):- healthy_sauces(L), write('Healthy Sauces:
{'), item_list(L).
display_options(sides):- sides(L), write('Sides: {'), item_list(L).
display_options(drinks):- drinks(L), write('Drinks: {'), item_list(L).
display_options(healthy_drinks):- healthy_drinks(L), write('Healthy Drinks:
{'), item_list(L).
display_options(cookies):- cookies(L), write('Cookies: {'), item_list(L).
display_options(dineOptions):- dineOptions(L), write('Dine Options: {'), ite
m_list(L).
```

```
display_options(paymentMethods):- paymentMethods(L), write('Payment Methods:
 {'), item_list(L).
```

In Line 63 – 67, I defined the item_list predicates and rules to display the list item. The letter '!' is the cut operator which throws away choice points that are created since entering the predicate before it.

```
% item_list predicates and rules to display the list item
% item_list with param empty list
item_list([]).
% item_list with list, write to output each list element and newline
item_list([X]) :- write(X), write('}'), nl.
% item_list with H and T of list, write to output each sub list element
item_list([X|Y]) :- write(X), write(', '), item_list(Y), !.
```

In Lines 70 – 74, I defined the predicate clear/0, which is calling retractall/1 for all the 12 dynamic predicates to remove all the stored facts in the Knowledge-Based System (KBS). The underscore is used to signify anonymous variable, or a "don't-care" variable, which means that it will match anything.

```
clear :-
    retractall(meal(_)), retractall(bread(_)), retractall(size(_)),
    retractall(main(_)), retractall(cheese(_)), retractall(veg(_)),
    retractall(sauce(_)), retractall(side(_)), retractall(drink(_)),
    retractall(cookie(_)), retractall(dineOption(_)), retractall(paymentMeth
od(_)).
```

In Lines 77 – 89, I defined the retrieve_user_choice/12 rule to retrieve all the menu items that has been chosen by the user. It uses the predicate findall/3. For example, `findall(X, meal(X), Meals)`, it creates a list of the instantiations X gets successively on backtracking over each menu item meals(x) and finally unifies the results with Meals.

```
retrieve_user_choice(Meals, Breads, Sizes, Mains, Cheeses, Vegs, Sauces, Sid
es, Drinks, Cookies, DineOptions, PaymentMethods) :-
    findall(X, meal(X), Meals),
    findall(X, bread(X), Breads),
    findall(X, size(X), Sizes),
    findall(X, main(X), Mains),
    findall(X, cheese(X), Cheeses),
    findall(X, veg(X), Vegs),
    findall(X, sauce(X), Sauces),
    findall(X, side(X), Sides),
    findall(X, drink(X), Drinks),
    findall(X, cookie(X), Cookies),
    findall(X, dineOption(X), DineOptions),
    findall(X, paymentMethod(X), PaymentMethods).
```

In Lines 93 – 105, I defined the predicates to allow the user to choose each menu item in a sequence for each meal types.

There are 5 meal types:

1. **Standard**
    a. Ask the user to pick every menu item.
    b. Menu items are: bread, size of bread, mains, cheese, vegetables, sauce, sides, drinks, cookies, dining options and payment methods.
2. **Vegetarian**
    a. Instead of asking the user to pick meat-based mains, we ask vegetarian mains.
    b. Instead of picking fatty sauce, we ask healthy sauce.
    c. Do not ask user to pick cheese.
3. **Healthy**
    a. Ask the user to pick healthy sauce instead of fatty sauce.
    b. Ask the user to pick healthy drinks instead of unhealthy drinks.
    c. Do not ask the user to pick cookies as it is unhealthy due to high sugar level.
4. **Value**
    a. Do not ask user to pick cheese and sides.
5. **Salad Bowl**
    a. Do not ask user to pick bread and size of bread.

```
% for each meal types, ask each menu item in the following sequences
% ask_standard_meal - Ask everything.
ask_standard_meal :- ask_bread, ask_size, ask_mains, ask_cheese, ask_vegetab
les, ask_sauce, ask_sides, ask_drinks, ask_cookies, ask_dineOptions, ask_pay
mentMethods.

% ask_vegetarian_meal - Do not ask meat mains & cheese. Ask vegetarian mains
 instead.
ask_vegetarian_meal :- ask_bread, ask_size, ask_veg_mains, ask_vegetables, a
sk_healthy_sauce, ask_sides, ask_drinks, ask_cookies, ask_dineOptions, ask_p
aymentMethods.

% ask_healthy_meal - Ask healthy_sauce and healthy_drinks instead. Do not as
k cookies as it is unhealthy due to high sugar level.
ask_healthy_meal :- ask_bread, ask_size, ask_mains, ask_cheese, ask_vegetabl
es, ask_healthy_sauce, ask_sides, ask_healthy_drinks, ask_dineOptions, ask_p
aymentMethods.

% ask_value_meal - Do not ask cheese and sides
ask_value_meal :- ask_bread, ask_size, ask_mains, ask_vegetables, ask_sauce,
 ask_drinks, ask_dineOptions, ask_paymentMethods.

% ask_salad_bowl_meal - Do not ask bread and size of bread
ask_salad_bowl_meal :- ask_mains, ask_cheese, ask_vegetables, ask_sauce, ask
_sides, ask_drinks, ask_cookies, ask_dineOptions, ask_paymentMethods.
```

In Lines 108 – 181, I defined the predicates to ask the user to select an option for each of the following menu items. All of them follow the same structure, which is to first write an output statement, print a new line, display the options for each menu item, and call the pick predicate to allow user to pick menu item.

```
% ask user to select an option for the following menu items
ask_meal :-
    write('What kind of meal would you like? '), nl,
    display_options(meals),
    pick_meal.

ask_bread :-
    write('What kind of bread would you like? '), nl,
    display_options(breads),
    pick_bread.

 ask_size :-
    write('What size would you like? '), nl,
    display_options(sizes),
    pick_size.

ask_mains :-
    write('What mains would you like? '), nl,
    display_options(mains),
    pick_main.

ask_veg_mains :-
    write('What vegetarian mains would you like? '), nl,
    display_options(veg_mains),
    pick_main.

ask_cheese :-
    write('Which of these tasty cheese would you like? '), nl,
    display_options(cheeses),
    pick_cheese.

ask_vegetables :-
    write('Don\'t forget your vegetables! What vegetables would you like?'),
 nl,
    display_options(vegs),
    pick_vegetable.

ask_sauce :-
    write('Let\'s get saucy! What kind of sauce would you like? '), nl,
    display_options(sauces),
    pick_sauce.

ask_healthy_sauce :-
```

```
    write('Let\'s get saucy but stay healthy! What kind of sauce would you l
ike? '), nl,
    display_options(healthy_sauces),
    pick_healthy_sauce.

ask_sides :-
    write('What kind of sides would you like? '), nl,
    display_options(sides),
    pick_side.

ask_drinks :-
    write('Let\'s stay hydrated! What drinks would you like? '), nl,
    display_options(drinks),
    pick_drink.

ask_healthy_drinks :-
    write('Let\'s stay hydrated healthily! What drinks would you like? '), n
l,
    display_options(healthy_drinks),
    pick_healthy_drink.

ask_cookies :-
    write('Cookies! What cookies do you fancy having today? '), nl,
    display_options(cookies),
    pick_cookie.

ask_dineOptions :-
    write('Would you like to dine in, takeaway or delivery?'), nl,
    display_options(dineOptions),
    pick_dine_option.

ask_paymentMethods :-
    write('How would you like to pay?'), nl,
    display_options(paymentMethods),
    pick_payment_method.
```

In Lines 184 – 296, I defined the predicates to allow the user to pick each menu item and update the item to its dynamic predicate in KBS. All of them follow the same structure.

For example, in pick_meal, we first read the user's input in X, if X is picked in meals, then we call assertz(meal(X)) to update the value in dynamic predicate meal. The semicolon ';' is the or operator. If the X picked is not an option in meals, then we write to output that it is invalid, print a new line, and recursively call pick_meal.

In pick_cheese, we allow the user to make multiple cheese selection and we use 0 as the terminating index. We first read the user's input in X, and we have a condition that if X is not 0, we proceed with if X is picked in cheeses, then we write to output, print new line and update the value in dynamic predicate cheese. Otherwise, write to output that it is invalid, print new line, recursively call pick_cheese. True is used to always succeed.

```
% allow user to pick each menu item and update item to dynamic predicate in
kbs
pick_meal :-
    read(X),
    picked(X, meals) -> assertz(meal(X));
    write('Invalid meal. Please try again.'), nl,
    pick_meal.

pick_bread :-
    read(X),
    picked(X, breads) -> assertz(bread(X));
    write('Invalid bread. Please try again.'), nl,
    pick_bread.

pick_size :-
    read(X),
    picked(X, sizes) -> assertz(size(X));
    write('Invalid size. Please try again.'), nl,
    pick_size.

pick_main :-
    read(X),
    picked(X, mains) -> assertz(main(X));
    write('Invalid main. Please try again.'), nl,
    pick_main.

pick_veg_main :-
    read(X),
    picked(X, veg_mains) -> assertz(main(X));
    write('Invalid vegetarian main. Please try again.'), nl,
    pick_veg_main.

pick_cheese :-
    read(X),
    (not(X == 0) ->
```

```
        (picked(X, cheeses) -> write(' (Enter 0 to end cheese selection.)'),
 nl, assertz(cheese(X));
        write('Invalid cheese. Please try again.'), nl),
        pick_cheese;
        true
    ).

pick_vegetable :-
    read(X),
    (not(X == 0) ->
        (picked(X, vegs) -> write(' (Enter 0 to end vegetables selection.)')
, nl, assertz(veg(X));
        write('Invalid veg. Please try again.'), nl),
        pick_vegetable;
        true
    ).

pick_sauce :-
    read(X),
    (not(X == 0) ->
        (picked(X, sauces) -> write(' (Enter 0 to end sauces selection.)'),
nl, assertz(sauce(X));
        write('Invalid sauce. Please try again.'), nl),
        pick_sauce;
        true
    ).

pick_healthy_sauce :-
    read(X),
    (not(X == 0) ->
        (picked(X, healthy_sauces) -> write(' (Enter 0 to end healthy sauce
selection.)'), nl, assertz(sauce(X));
        write('Invalid healthy sauce. Please try again.'), nl),
        pick_healthy_sauce;
        true
    ).

pick_side :-
    read(X),
    (not(X == 0) ->
        (picked(X, sides) -> write(' (Enter 0 to end sides selection.)'), nl
, assertz(side(X));
        write('Invalid sides. Please try again.'), nl),
        pick_side;
        true
    ).

pick_drink :-
```

```
    read(X),
    (not(X == 0) ->
        (picked(X, drinks) -> write(' (Enter 0 to end drink selection.)'), n
l, assertz(drink(X));
        write('Invalid drinks. Please try again.'), nl),
        pick_drink;
        true
    ).

pick_healthy_drink :-
    read(X),
    (not(X == 0) ->
        (picked(X, healthy_drinks) -> write(' (Enter 0 to end healthy drink
selection.)'), nl, assertz(drink(X));
        write('Invalid healthy drink. Please try again.'), nl),
        pick_healthy_drink;
        true
    ).

pick_cookie :-
    read(X),
    (not(X == 0) ->
        (picked(X, cookies) -> write(' (Enter 0 to end cookie selection.)'),
 nl, assertz(cookie(X));
        write('Invalid cookies. Please try again.'), nl),
        pick_cookie;
        true
    ).

pick_dine_option :-
    read(X),
    picked(X, dineOptions) -> nl, assertz(dineOption(X));
    write('Invalid dine options. Please try again.'), nl,
    pick_dine_option.

pick_payment_method:-
    read(X),
    picked(X, paymentMethods) -> nl, assertz(paymentMethod(X));
    write('Invalid payment method. Please try again.'), nl,
    pick_payment_method.
```

In Lines 299 – 308, I defined a predicate current_date_time/6 which performs the action of retrieving and displaying the current date time. In the 3rd parameter of stamp_date_time, local is used instead of UTC. This predicate will be used later in the predicate print_receipt.

```
% retrieve and display the current date time
current_date_time(Day, Month, Year, Hour, Min, Sec) :-
    get_time(T),
    stamp_date_time(T,DateTime,'local'),
    arg(1, DateTime, Year),
    arg(2, DateTime, Month),
    arg(3, DateTime, Day),
    arg(4, DateTime, Hour),
    arg(5, DateTime, Min),
    arg(6, DateTime, Sec),
    format('~t ~w ~25|~w/~w/~w ~w:~w:~0f~n', ['Order Time:', Day, Month, Yea
r, Hour, Min, Sec]).
```

In Lines 312 – 319, I defined a predicate generate_price/1 to generate a random number from 5.1 to 11.9, for the price of his order as a placeholder to simulate the receipt. It calculates the price, GST, and subtotal and write to output. For simplicity, we assume the user has paid the exact amount, meaning that change is $0.

```
% generate random number for price as a placeholder
% assume user has paid exact amount
generate_price(Price):-
    random(5.1,11.9,Price),
    format('~t ~w ~25|$~2f~n', ['Total:', Price]),
    format('~t ~w ~25|~w%~n', ['GST:', 7]),
    product(Price, 1.07, SubTotal),
    format('~t ~w ~25|$~2f~n', ['Subtotal:', SubTotal]),
    format('~t ~w ~25|$~2f~n', ['Paid:', SubTotal]),
    format('~t ~w ~25|$~2f~n', ['Change:', 0]).
```

In Lines 323 – 334, I defined the predicate generate_price_delivery/1 which is similar to generate_price/1, however this includes a delivery surcharge of $3.50 in the subtotal price calculation. This predicate will be used if the user has chosen the dine option: delivery.

```
% generate random number for price with delivery as a placeholder
% assume user has paid exact amount
generate_price_delivery(Price):-
    random(5.1,11.9,Price),
    format('~t ~w ~25|$~2f~n', ['Total:', Price]),
    format('~t ~w ~25|~w%~n', ['GST:', 7]),
    product(Price, 0.07, GstTax),
    format('~t ~w ~25|$~2f~n', ['GST tax:', GstTax]),
    format('~t ~w ~25|$~2f~n', ['Delivery Surcharge:', 3.50]),
    product(Price, 1.07, TotalPrice),
    sum(TotalPrice, 3.5, SubTotal),
```

```
    format('~t ~w ~25|$~2f~n', ['Subtotal:', SubTotal]),
    format('~t ~w ~25|$~2f~n', ['Paid:', SubTotal]),
    format('~t ~w ~25|$~2f~n', ['Change:', 0]).
```

In Lines 338 and 341, I defined predicates sum/3 and product/3 to which is simple addition and multiplication function to facilitate the calculation of price.

```
% methods to calculate price
% sum - find the result of A + B
sum(A, B, Result) :- Result is A + B.

% product - find the result of A * B
product(A, B, Result) :- Result is A * B.
```

In Lines 345 – 364, I defined the predicate print_receipt to print a receipt that displays all the menu items picked by the user. I used atomic_list_concat to concatenate each menu item and write to output and print a new line. In line 362, if the DineOption is delivery, we use generate_price_delivery/1 and print the estimated delivery time. Otherwise, if we use the basic generate_price for dine in and takeaway orders.

```
% print_receipt to display all the picked menu items by user. Concatenate ea
ch menu item and write to output and newline.
% E.g. If Bread is not equals to '', then write out the picked Bread. Otherw
ise, write none.
print_receipt :-
    format('~`-t~50|~n'),
    format('~tSUBWAY~t~50|~n'),
    current_date_time(_Day, _Month, _Year, _Hour, _Min, _Sec),
    retrieve_user_choice(Meals, Breads, Sizes, Mains, Cheeses, Vegs, Sauces,
 Sides, Drinks, Cookies, DineOptions, PaymentMethods),
    atomic_list_concat(Meals, Meal), format('~t ~w ~25|~w~n', ['Meal:', Meal
]),
    atomic_list_concat(Breads, Bread), format('~t ~w ~25|', ['Bread:']), (no
t(Bread == '') -> write(Bread); write(none)), nl,
    atomic_list_concat(Sizes, Size), format('~t ~w ~25|', ['Size:']), (not(S
ize == '') -> write(Size); write(none)), nl,
    atomic_list_concat(Mains, Main), format('~t ~w ~25|', ['Main:']), (not(M
ain == '') -> write(Main); write(none)), nl,
    atomic_list_concat(Cheeses, ', ', Cheese), format('~t ~w ~25|', ['Cheese
s:']), (not(Cheese == '') -> foreach(member(X, Cheeses), format('~t~25|~w~n'
,[X])); write(none)), nl,
    atomic_list_concat(Vegs, ', ', Veg), format('~t ~w ~25|', ['Vegetables:'
]), (not(Veg == '') -> foreach(member(X, Vegs), format('~t~25|~w~n',[X])); w
rite(none)), nl,
    atomic_list_concat(Sauces, ', ', Sauce), format('~t ~w ~25|', ['Sauces:'
]), (not(Sauce == '') -> foreach(member(X, Sauces), format('~t~25|~w~n',[X])
); write(none)), nl,
```

```
    atomic_list_concat(Sides, ', ', Side), format('~t ~w ~25|', ['Sides:']),
 (not(Side == '') -> foreach(member(X, Sides), format('~t~25|~w~n',[X])); wr
ite(none)), nl,
    atomic_list_concat(Drinks, ', ', Drink), format('~t ~w ~25|', ['Drinks:'
]), (not(Drink == '') -> foreach(member(X, Drinks), format('~t~25|~w~n',[X])
); write(none)), nl,
    atomic_list_concat(Cookies, ', ', Cookie), format('~t ~w ~25|', ['Cookie
s:']), (not(Cookie == '') -> foreach(member(X, Cookies), format('~t~25|~w~n'
,[X])); write(none)), nl,
    atomic_list_concat(DineOptions, ', ', DineOption), format('~t ~w ~25|~w~
n', ['Dine Options:', DineOption]),
    atomic_list_concat(PaymentMethods, PaymentMethod), format('~t ~w ~25|~w~
n', ['Payment Methods:', PaymentMethod]),
    ((DineOption = 'delivery') -> generate_price_delivery(_Price), format('E
stimated delivery time: 30 minutes~n'); generate_price(_Price)),
    format('~t~w~40|~n',['Thank you! Enjoy your meal!']),
    format('~`-t~50|~n').
```

Lastly, in lines 372 – 382, I defined the predicate new to allow the user to start a new subway order. This is the entry point of the application. We ask the user which meal type, and use the corresponding ask predicates, after the order is done, we print the receipt and use clear to remove all stored dynamic predicate values to allow for new orders.

```
% Entry point of application
% Start a new subway order by asking user what type of meal he wants.
% If input is vegetarian, use ask_vegetarian_meal
% If input is healthy, use ask_healthy_meal
% If input is value, use ask_value_meal
% If input is salad_bowl, use ask_salad_bowl_meal
new :-
    format('Hello! Welcome to Subway! ~nMay I take your order please?~n'),
    ask_meal, meal(Meal),
    ((Meal == vegetarian) -> ask_vegetarian_meal;
    (Meal == healthy) -> ask_healthy_meal;
    (Meal == value) -> ask_value_meal;
    (Meal == salad_bowl) -> ask_salad_bowl_meal;
    ask_standard_meal),
    format('Ok! All done! Here is your receipt:~n'),
    print_receipt,
    clear.
```

1.3 – How to Run

To run the Prolog script, simply use the command '**new.**' to start a new Subway order.

In the following screenshots, I will demonstrate the user choosing each of the 5 meal types, various menu items, different dining options and payment methods.

2 – Screenshots of Subway Talking Bot

```
?-
% c:/Users/sherna/OneDrive - Nanyang Technological University/AY2021-S1/CZ3005 - AI/Lab/Lab 4 - Implementing a Talk
ing Box with Prolog/LiewZhiLi-U1821610C-Lab4/LiewZhiLi_qn_3.pl compiled 0.00 sec, 92 clauses
?- new.
Hello! Welcome to Subway!
May I take your order please?
What kind of meal would you like?
Meals: {standard, vegetarian, healthy, value, salad_bowl}
|: standard.
What kind of bread would you like?
Breads: {multigrain, hearty_italian, italian_wheat, parmesan_oregano, honey_oat, flat_bread}
|: multigrain.
What size would you like?
Sizes: {six_inch, foot_long}
|: six_inch.
What mains would you like?
Mains: {chicken, ham, turkey, tuna, beef, meatball_marinara, egg_mayo, italian_bmt, veggie_delite, veggie_patty, to
fu_patty}
|: chicken.
Which of these tasty cheese would you like?
Cheeses: {american, monterey, feta, mozzarella, cheddar, swiss, 0 (none)}
|: american.
 (Enter 0 to end cheese selection.)
|: monterey.
 (Enter 0 to end cheese selection.)
|: 0.
Don't forget your vegetables! What vegetables would you like?
Vegetables: {cucumbers, tomatoes, lettuce, green_peppers, red_onions, jalapenos, black_olives, pickles, 0 (none)}
|: lettuce.
 (Enter 0 to end vegetables selection.)
|: green_peppers.
 (Enter 0 to end vegetables selection.)
|: red_onions.
 (Enter 0 to end vegetables selection.)
|: 0.
Let's get saucy! What kind of sauce would you like?
Sauces: {chipotle, ranch, bbq, chilli, tomato, mayonnaise, vinegar, 0 (none)}
|: chipotle.
 (Enter 0 to end sauces selection.)
|: bbq.
 (Enter 0 to end sauces selection.)
|: 0.
What kind of sides would you like?
Sides: {potato_chips, hashbrowns, yogurt, 0 (none)}
|: hashbrowns.
 (Enter 0 to end sides selection.)
|: 0.
Let's stay hydrated! What drinks would you like?
Drinks: {coke, sprite, orange_fanta, mineral_water, orange_juice, green_tea, hot_coffee, hot_tea, 0 (none)}
|: coke.
 (Enter 0 to end drink selection.)
|: 0.
Cookies! What cookies do you fancy having today?
Cookies: {choco_chip, double_choco_chip, peanut_butter, white_chip_macadamia_nut, oatmeal_raisin, raspberry, 0 (non
e)}
|: double_choco_chip.
 (Enter 0 to end cookie selection.)
|: peanut_butter.
 (Enter 0 to end cookie selection.)
|: 0.
Would you like to dine in, takeaway or delivery?
Dine Options: {dine_in, takeaway, delivery}
|: dine_in.

How would you like to pay?
Payment Methods: {cash, credit_card, nets}
|: cash.

Ok! All done! Here is your receipt:
------------------------------------------------------
                    SUBWAY
             Order Time: 19/11/2020 14:48:5
                   Meal: standard
                  Bread: multigrain
                   Size: six_inch
                   Main: chicken
                Cheeses: american
                         monterey

             Vegetables: lettuce
                         green_peppers
                         red_onions

                 Sauces: chipotle
                         bbq

                  Sides: hashbrowns

                 Drinks: coke

                Cookies: double_choco_chip
                         peanut_butter

          Dine Options: dine_in
        Payment Methods: cash
                  Total: $7.17
                    GST: 7%
               Subtotal: $7.67
                   Paid: $7.67
                 Change: $0.00
             Thank you! Enjoy your meal!
------------------------------------------------------
true.

?-
```

```
?- new.
Hello! Welcome to Subway!
May I take your order please?
What kind of meal would you like?
Meals: {standard, vegetarian, healthy, value, salad_bowl}
|: vegetarian.
What kind of bread would you like?
Breads: {multigrain, hearty_italian, italian_wheat, parmesan_oregano, honey_oat, flat_bread}
|: hearty_italian.
What size would you like?
Sizes: {six_inch, foot_long}
|: six_inch.
What vegetarian mains would you like?
Veg Mains: {veggie_delite, veggie_patty, tofu_patty}
|: veggie_delite.
Don't forget your vegetables! What vegetables would you like?
Vegetables: {cucumbers, tomatoes, lettuce, green_peppers, red_onions, jalapenos, black_olives, pickles, 0 (none)}
|: cucumbers.
 (Enter 0 to end vegetables selection.)
|: tomatoes.
 (Enter 0 to end vegetables selection.)
|: lettuce.
 (Enter 0 to end vegetables selection.)
|: jalapenos.
 (Enter 0 to end vegetables selection.)
|: pickles.
 (Enter 0 to end vegetables selection.)
|: 0.
Let's get saucy but stay healthy! What kind of sauce would you like?
Healthy Sauces: {chipotle, ranch, tomato, vinegar, 0 (none)}
|: vinegar.
 (Enter 0 to end healthy sauce selection.)
|: chipotle.
 (Enter 0 to end healthy sauce selection.)
|: 0.
What kind of sides would you like?
Sides: {potato_chips, hashbrowns, yogurt, 0 (none)}
|: yogurt.
 (Enter 0 to end sides selection.)
|: 0.
Let's stay hydrated! What drinks would you like?
Drinks: {coke, sprite, orange_fanta, mineral_water, orange_juice, green_tea, hot_coffee, hot_tea, 0 (none)}
|: hot_tea.
 (Enter 0 to end drink selection.)
|: mineral_water.
 (Enter 0 to end drink selection.)
|: 0.
Cookies! What cookies do you fancy having today?
Cookies: {choco_chip, double_choco_chip, peanut_butter, white_chip_macadamia_nut, oatmeal_raisin, raspberry, 0 (non
e)}
|: raspberry.
 (Enter 0 to end cookie selection.)
|: 0.
Would you like to dine in, takeaway or delivery?
Dine Options: {dine_in, takeaway, delivery}
|: takeaway.

How would you like to pay?
Payment Methods: {cash, credit_card, nets}
|: credit_card.

Ok! All done! Here is your receipt:
------------------------------------------------------
                     SUBWAY
             Order Time: 19/11/2020 15:0:21
                   Meal: vegetarian
                  Bread: hearty_italian
                   Size: six_inch
                   Main: veggie_delite
              Cheeses: none
             Vegetables: cucumbers
                         tomatoes
                         lettuce
                         jalapenos
                         pickles

                 Sauces: vinegar
                         chipotle

                  Sides: yogurt

                 Drinks: hot_tea
                         mineral_water

                Cookies: raspberry

          Dine Options: takeaway
        Payment Methods: credit_card
                  Total: $9.02
                    GST: 7%
               Subtotal: $9.65
                   Paid: $9.65
                 Change: $0.00
             Thank you! Enjoy your meal!
------------------------------------------------------
true.

?-
```

```
?- new.
Hello! Welcome to Subway!
May I take your order please?
What kind of meal would you like?
Meals: {standard, vegetarian, healthy, value, salad_bowl}
|: healthy.
What kind of bread would you like?
Breads: {multigrain, hearty_italian, italian_wheat, parmesan_oregano, honey_oat, flat_bread}
|: italian_wheat.
What size would you like?
Sizes: {six_inch, foot_long}
|: foot_long.
What mains would you like?
Mains: {chicken, ham, turkey, tuna, beef, meatball_marinara, egg_mayo, italian_bmt, veggie_delite, veggie_patty, to
fu_patty}
|: italian_bmt.
Which of these tasty cheese would you like?
Cheeses: {american, monterey, feta, mozzarella, cheddar, swiss, 0 (none)}
|: swiss.
 (Enter 0 to end cheese selection.)
|: 0.
Don't forget your vegetables! What vegetables would you like?
Vegetables: {cucumbers, tomatoes, lettuce, green_peppers, red_onions, jalapenos, black_olives, pickles, 0 (none)}
|: green_peppers.
 (Enter 0 to end vegetables selection.)
|: lettuce.
 (Enter 0 to end vegetables selection.)
|: cucumbers.
 (Enter 0 to end vegetables selection.)
|: pickles.
 (Enter 0 to end vegetables selection.)
|: red_onions.
 (Enter 0 to end vegetables selection.)
|: black_olives.
 (Enter 0 to end vegetables selection.)
|: 0.
Let's get saucy but stay healthy! What kind of sauce would you like?
Healthy Sauces: {chipotle, ranch, tomato, vinegar, 0 (none)}
|: ranch.
 (Enter 0 to end healthy sauce selection.)
|: tomato.
 (Enter 0 to end healthy sauce selection.)
|: 0.
What kind of sides would you like?
Sides: {potato_chips, hashbrowns, yogurt, 0 (none)}
|: yogurt.
 (Enter 0 to end sides selection.)
|: 0.
Let's stay hydrated healthily! What drinks would you like?
Healthy Drinks: {diet_coke, diet_sprite, mineral_water, green_tea_no_sugar, hot_coffee_no_sugar, hot_tea_no_sugar,
0 (none)}
|: diet_coke.
 (Enter 0 to end healthy drink selection.)
|: 0.
Would you like to dine in, takeaway or delivery?
Dine Options: {dine_in, takeaway, delivery}
|: delivery.

How would you like to pay?
Payment Methods: {cash, credit_card, nets}
|: credit_card.

Ok! All done! Here is your receipt:
------------------------------------------------------
                      SUBWAY
              Order Time: 19/11/2020 15:8:37
                    Meal: healthy
                   Bread: italian_wheat
                    Size: foot_long
                    Main: italian_bmt
                Cheeses: swiss

              Vegetables: green_peppers
                          lettuce
                          cucumbers
                          pickles
                          red_onions
                          black_olives

                  Sauces: ranch
                          tomato

                   Sides: yogurt

                  Drinks: diet_coke

                 Cookies: none
             Dine Options: delivery
          Payment Methods: credit_card
                    Total: $6.55
                      GST: 7%
                  GST tax: $0.46
      Delivery Surcharge: $3.50
                 Subtotal: $10.51
                     Paid: $10.51
                   Change: $0.00
Estimated delivery time: 30 minutes
              Thank you! Enjoy your meal!
------------------------------------------------------
true.

?-
```

```
?- new.
Hello! Welcome to Subway!
May I take your order please?
What kind of meal would you like?
Meals: {standard, vegetarian, healthy, value, salad_bowl}
|: value.
What kind of bread would you like?
Breads: {multigrain, hearty_italian, italian_wheat, parmesan_oregano, honey_oat, flat_bread}
|: honey_oat.
What size would you like?
Sizes: {six_inch, foot_long}
|: six_inch.
What mains would you like?
Mains: {chicken, ham, turkey, tuna, beef, meatball_marinara, egg_mayo, italian_bmt, veggie_delite, veggie_patty, to
fu_patty}
|: egg_mayo.
Don't forget your vegetables! What vegetables would you like?
Vegetables: {cucumbers, tomatoes, lettuce, green_peppers, red_onions, jalapenos, black_olives, pickles, 0 (none)}
|: lettuce.
 (Enter 0 to end vegetables selection.)
|: cucumbers.
 (Enter 0 to end vegetables selection.)
|: 0.
Let's get saucy! What kind of sauce would you like?
Sauces: {chipotle, ranch, bbq, chilli, tomato, mayonnaise, vinegar, 0 (none)}
|: chilli.
 (Enter 0 to end sauces selection.)
|: 0.
Let's stay hydrated! What drinks would you like?
Drinks: {coke, sprite, orange_fanta, mineral_water, orange_juice, green_tea, hot_coffee, hot_tea, 0 (none)}
|: 0.
Would you like to dine in, takeaway or delivery?
Dine Options: {dine_in, takeaway, delivery}
|: delivery.

How would you like to pay?
Payment Methods: {cash, credit_card, nets}
|: credit_card.

Ok! All done! Here is your receipt:
------------------------------------------------------
                     SUBWAY
             Order Time: 19/11/2020 15:10:48
                   Meal: value
                  Bread: honey_oat
                   Size: six_inch
                   Main: egg_mayo
              Cheeses: none
             Vegetables: lettuce
                         cucumbers

                 Sauces: chilli

                  Sides: none
                 Drinks: none
                Cookies: none
           Dine Options: delivery
        Payment Methods: credit_card
                  Total: $5.45
                    GST: 7%
                GST tax: $0.38
     Delivery Surcharge: $3.50
                Subtotal: $9.33
                   Paid: $9.33
                 Change: $0.00
Estimated delivery time: 30 minutes
             Thank you! Enjoy your meal!
------------------------------------------------------
true.

?-
```

```
?- new.
Hello! Welcome to Subway!
May I take your order please?
What kind of meal would you like?
Meals: {standard, vegetarian, healthy, value, salad_bowl}
|: salad_bowl.
What mains would you like?
Mains: {chicken, ham, turkey, tuna, beef, meatball_marinara, egg_mayo, italian_bmt, veggie_delite, veggie_patty, to
fu_patty}
|: turkey.
Which of these tasty cheese would you like?
Cheeses: {american, monterey, feta, mozzarella, cheddar, swiss, 0 (none)}
|: 0.
Don't forget your vegetables! What vegetables would you like?
Vegetables: {cucumbers, tomatoes, lettuce, green_peppers, red_onions, jalapenos, black_olives, pickles, 0 (none)}
|: cucumbers.
 (Enter 0 to end vegetables selection.)
|: tomatoes.
 (Enter 0 to end vegetables selection.)
|: lettuce.
 (Enter 0 to end vegetables selection.)
|: green_peppers.
 (Enter 0 to end vegetables selection.)
|: 0.
Let's get saucy! What kind of sauce would you like?
Sauces: {chipotle, ranch, bbq, chilli, tomato, mayonnaise, vinegar, 0 (none)}
|: 0.
What kind of sides would you like?
Sides: {potato_chips, hashbrowns, yogurt, 0 (none)}
|: 0.
Let's stay hydrated! What drinks would you like?
Drinks: {coke, sprite, orange_fanta, mineral_water, orange_juice, green_tea, hot_coffee, hot_tea, 0 (none)}
|: hot_tea.
 (Enter 0 to end drink selection.)
|: 0.
Cookies! What cookies do you fancy having today?
Cookies: {choco_chip, double_choco_chip, peanut_butter, white_chip_macadamia_nut, oatmeal_raisin, raspberry, 0 (non
e)}
|: peanut_butter.
 (Enter 0 to end cookie selection.)
|: 0.
Would you like to dine in, takeaway or delivery?
Dine Options: {dine_in, takeaway, delivery}
|: dine_in.

How would you like to pay?
Payment Methods: {cash, credit_card, nets}
|: nets.

Ok! All done! Here is your receipt:
------------------------------------------------------
                    SUBWAY
             Order Time: 19/11/2020 15:12:23
                   Meal: salad_bowl
                  Bread: none
                   Size: none
                   Main: turkey
               Cheeses: none
             Vegetables: cucumbers
                         tomatoes
                         lettuce
                         green_peppers

                 Sauces: none
                  Sides: none
                 Drinks: hot_tea

                Cookies: peanut_butter

          Dine Options: dine_in
        Payment Methods: nets
                  Total: $5.84
                    GST: 7%
               Subtotal: $6.25
                   Paid: $6.25
                 Change: $0.00
             Thank you! Enjoy your meal!
------------------------------------------------------
true.

?- 
```