What to Eat ~ Application

Group: SegFaults

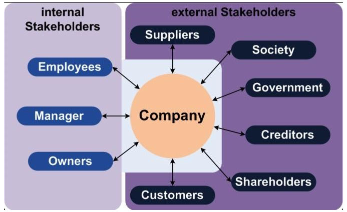
Members:

- Brian Lucero
- Matthew Padden
- Bilal El-haghassan
- Tommy Nguyen
- Berkeley C.

Course #: CPSC 362

Professor: Sara Ghadami

WTE ~ Introduction



Team Brainstorming

____

[x] - Stock Price Prediction App

[x] - Fitness App

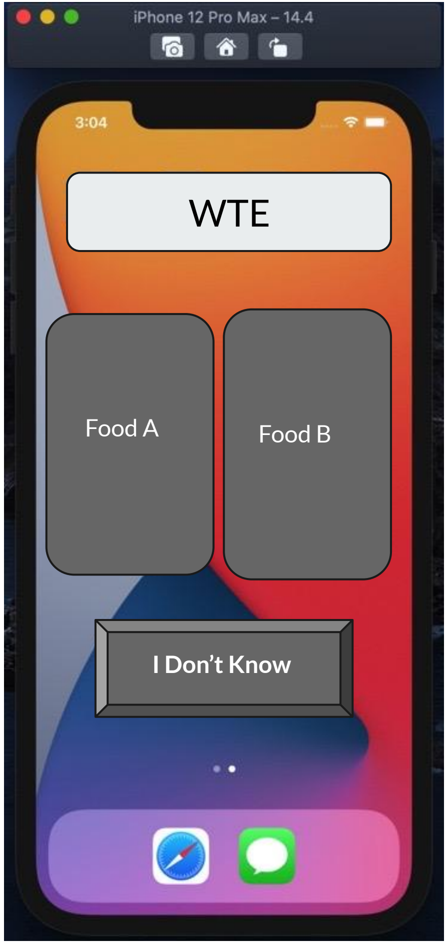
[x] - Cryptocurrency Trading App

[x] - Interplanetary Communication App

[X] - Mobile Game

[✅] - Food App

- **Simple GUI**
  - Comfortable First-Time User Experience
  - Less Clutter = Less Overwhelming
- **Short Food Survey**
  - I don't know button (randomization)
  - Aids user decision making
  - Give a recommendation reflecting Users choices
- **Streamlined option screens = less time wasted**
  - Maybe only 1-2 Question Pages (Canvas)
- **Programming Language:** Python
  - Kivy Framework
  - Multi-Platform



~ Cook at home ~

Minimum wage employee just wanting to go on break
Customer taking forever trying to decide what they want to eat

~ Dining out ~

WTE ~ User Stories

Functional:

1. *As a WTE user; I want to answer a set of questions; So that I can get a recommendation according to my tastes*

2. *As a WTE user; I want an I don't know button; So that I can have a randomized option chosen for me*

3. *As a WTE user; I want to be able to start over when I choose to; So that I can start over if I chose the wrong options*

4. *As a WTE user; I want to see images of what I am choosing; So that I better understand what my options are*

5. *As a WTE user; I want to choose to eat out or cook at home; So I can get recommendations for recipes or outside food recommendations*

Non-Functional:

1. *As a WTE user*
I want to be able to download this app on any mobile device
So that I don't have to use a different device to access it

2. *As a WTE userx*
I want the food suggestions to be accurate 95 percent of the time
So that I can receive useful suggestions when using the app

3. *As a WTE user*
I want the app to take less processable time to reach a conclusion
So that I can get to eat sooner

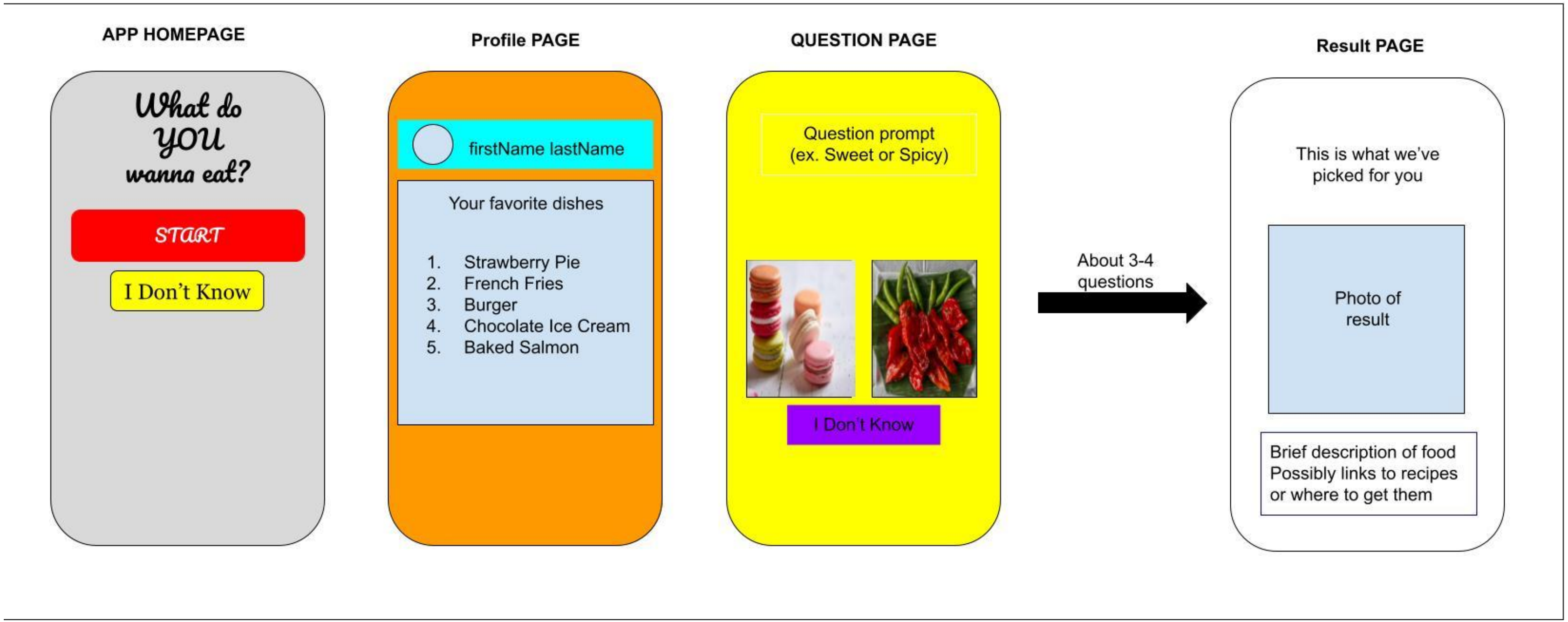
4. *As a WTE user*
I want the app to be available 24/7 99% of the time
So that I can access it at any time wherever I am

5. *As a WTE user*
I want scheduled maintenances to finish in less than 30 minutes
So that I can access it as soon as possible

WTE ~ GUI ~ initial design

Questions Asked:

1. ***Dining out or Cook at Home***
2. ***Sweet or Salty***
3. ***Healthy or Casual***
4. ***Light or Heavy***



APP HOMEPAGE

What do you wanna eat?

START

I Don't Know

Profile PAGE

firstName lastName

Your favorite dishes

1. Strawberry Pie
2. French Fries
3. Burger
4. Chocolate Ice Cream
5. Baked Salmon

QUESTION PAGE

Question prompt
(ex. Sweet or Spicy)

I Don't Know

About 3-4 questions

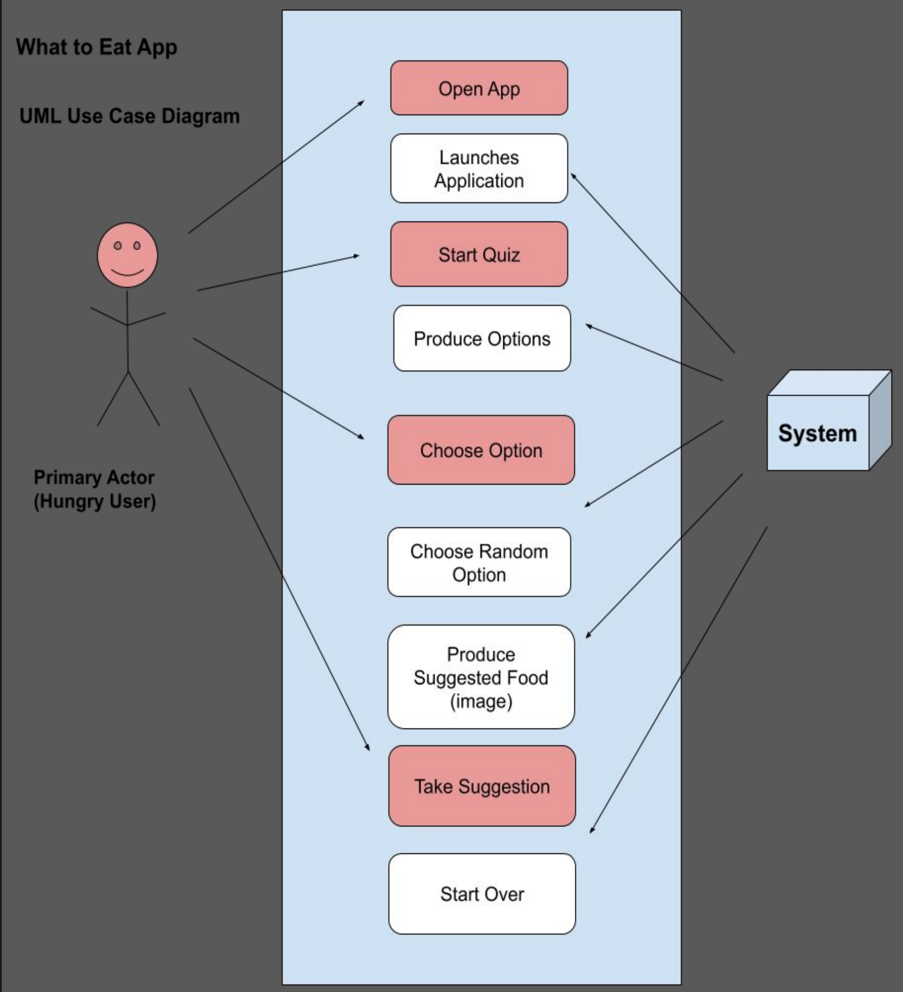
Result PAGE

This is what we've picked for you

Photo of result

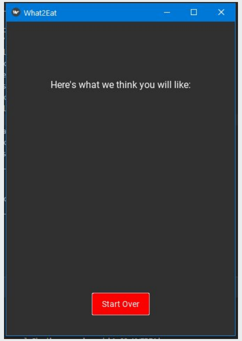
Brief description of food
Possibly links to recipes
or where to get them

NOTE: Conceptually, the integration of the initial Client requirements and also the User’s Functional requirements is observed to be well-defined. This makes it easier for our development team to build the application more accurately and much quicker.

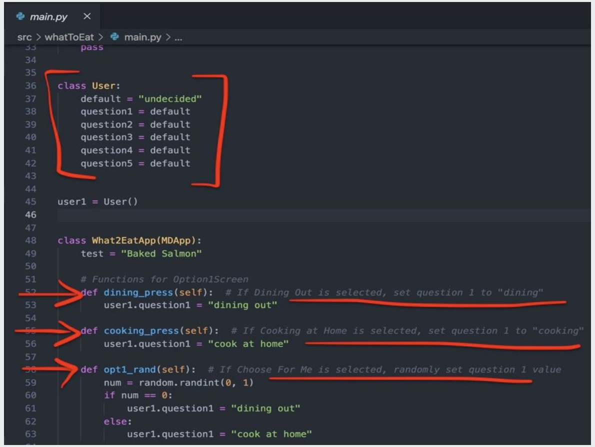


Functional:

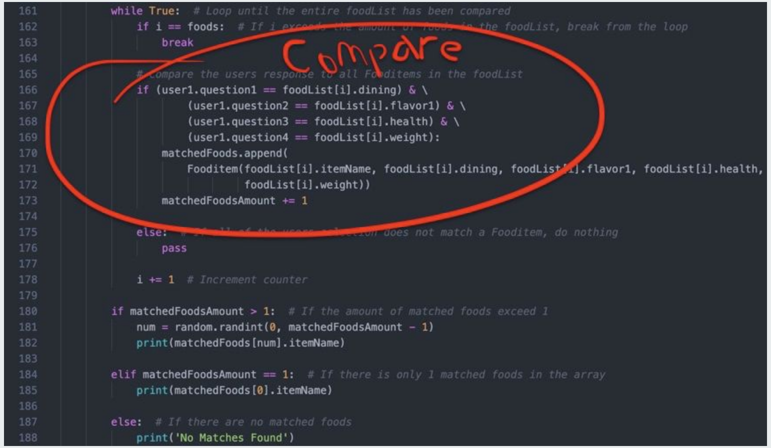
1. *As a WTE user; I want to answer a set of questions; So that I can get a recommendation according to my tastes*



Here's what we think you will like:



```python
        pass


class User:
    default = "undecided"
    question1 = default
    question2 = default
    question3 = default
    question4 = default
    question5 = default


user1 = User()


class What2EatApp(MDApp):
    test = "Baked Salmon"

    # Functions for Option1Screen
    def dining_press(self):  # If Dining Out is selected, set question 1 to "dining"
        user1.question1 = "dining out"

    def cooking_press(self):  # If Cooking at Home is selected, set question 1 to "cooking"
        user1.question1 = "cook at home"

    def opt1_rand(self):  # If Choose For Me is selected, randomly set question 1 value
        num = random.randint(0, 1)
        if num == 0:
            user1.question1 = "dining out"
        else:
            user1.question1 = "cook at home"
```



```python
        while True:  # Loop until the entire foodList has been compared
            if i == foods:  # If i exceeds the amount of foods in the foodList, break from the loop
                break

            # Compare the users response to all Fooditems in the foodList
            if (user1.question1 == foodList[i].dining) & \
                    (user1.question2 == foodList[i].flavor1) & \
                    (user1.question3 == foodList[i].health) & \
                    (user1.question4 == foodList[i].weight):
                matchedFoods.append(
                    Fooditem(foodList[i].itemName, foodList[i].dining, foodList[i].flavor1, foodList[i].health,
                             foodList[i].weight))
                matchedFoodsAmount += 1

            else:  # If all of the users selection does not match a Fooditem, do nothing
                pass

            i += 1  # Increment counter

        if matchedFoodsAmount > 1:  # If the amount of matched foods exceed 1
            num = random.randint(0, matchedFoodsAmount - 1)
            print(matchedFoods[num].itemName)

        elif matchedFoodsAmount == 1:  # If there is only 1 matched foods in the array
            print(matchedFoods[0].itemName)

        else:  # If there are no matched foods
            print('No Matches Found')
```

UC-01

Use Case: Receive food suggestions

Description: User wants to answer set of questions to get food suggestions

Primary Actor: Hungry User

Precondition: User has Application on Phone and it is open

Post Conditions:

Success end conditions:

User gets a food recommendation

Failure end conditions:

User gets bad recommendation

System could not find food to recommend

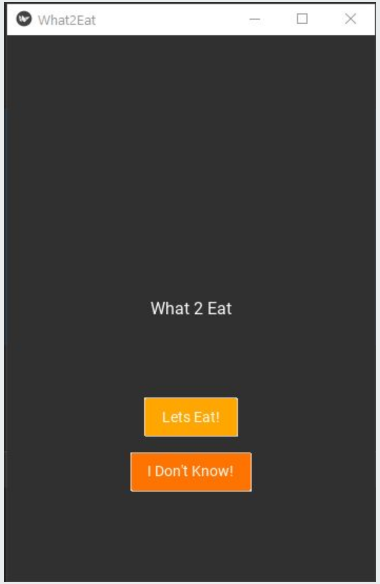
Trigger: User doesn't know what exactly they want to eat

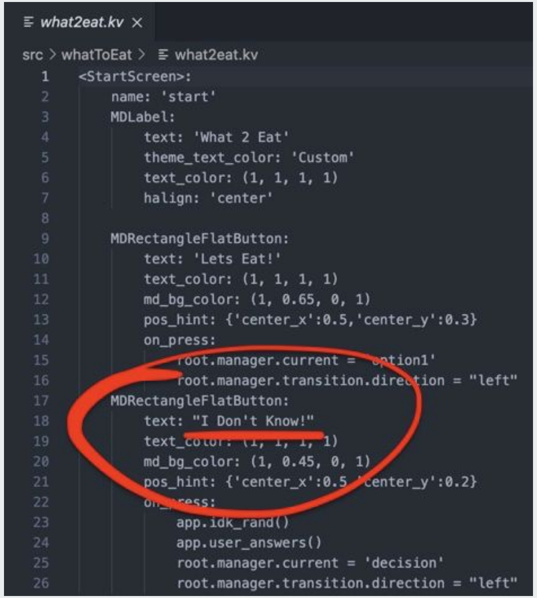
Main Success Scenario: capture user and system interactions

1. User launches application
2. System opens homepage
3. User Answers question and chooses option on home screen
4. System navigates to Option 1 page
5. User chooses between two options on screen two.
6. System navigates to Option 2 page
7. User chooses between two options on screen two.
8. System navigates to Option 3 page
9. User chooses between two options on screen two.
10. System navigates to Option 3 page

Functional:

2. *As a WTE user; I want an I don't know button; So that I can have a randomized option chosen for me*





```
<StartScreen>:
    name: 'start'
    MDLabel:
        text: 'What 2 Eat'
        theme_text_color: 'Custom'
        text_color: (1, 1, 1, 1)
        halign: 'center'

    MDRectangleFlatButton:
        text: 'Lets Eat!'
        text_color: (1, 1, 1, 1)
        md_bg_color: (1, 0.65, 0, 1)
        pos_hint: {'center_x':0.5,'center_y':0.3}
        on_press:
            root.manager.current = 'option1'
            root.manager.transition.direction = "left"
    MDRectangleFlatButton:
        text: "I Don't Know!"
        text_color: (1, 1, 1, 1)
        md_bg_color: (1, 0.45, 0, 1)
        pos_hint: {'center_x':0.5,'center_y':0.2}
        on_press:
            app.idk_rand()
            app.user_answers()
            root.manager.current = 'decision'
            root.manager.transition.direction = "left"
```

UC-02

Use Case: Push random selection button

ID: UC-02

Description: User can select an "I Don't Know" button which will randomly make a selection on all options for the user.

Primary Actor: Hungry User

Precondition:

User has a phone
WTE user has the application open

Post Conditions:

Success end conditions:

The user presses the button and the application will make random selections for the user. The application will now be able to make a random food selection for the user.

Failure end conditions:

The user presses the button and the application will not make random selections for the user. The application will not be able to make a random food selection for the user.

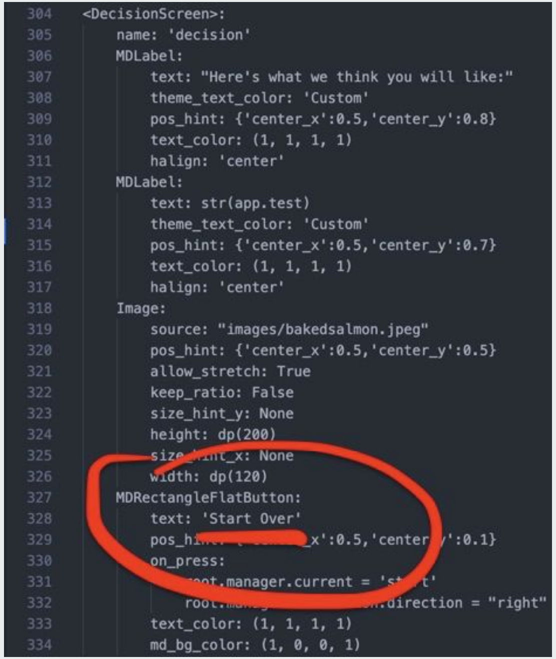
Trigger: User cannot decide between the two options

Main Success Scenario:

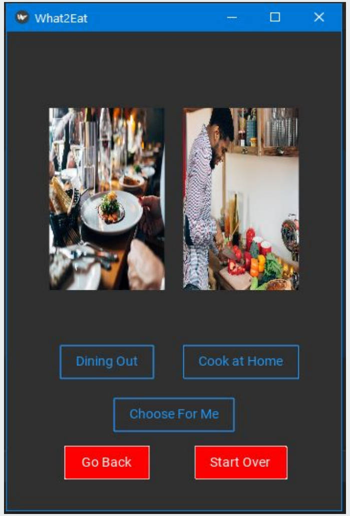
1. User will launch the app "What 2 Eat"
2. Application will show the home screen
3. User will press the "I Don't Know" button
4. Application will randomly select the options for the user

Functional:

3. *As a WTE user; I want to be able to start over when I choose to; So that I can start over if I chose the wrong options*



```
<DecisionScreen>:
    name: 'decision'
    MDLabel:
        text: "Here's what we think you will like:"
        theme_text_color: 'Custom'
        pos_hint: {'center_x':0.5,'center_y':0.8}
        text_color: (1, 1, 1, 1)
        halign: 'center'
    MDLabel:
        text: str(app.test)
        theme_text_color: 'Custom'
        pos_hint: {'center_x':0.5,'center_y':0.7}
        text_color: (1, 1, 1, 1)
        halign: 'center'
    Image:
        source: "images/bakedsalmon.jpeg"
        pos_hint: {'center_x':0.5,'center_y':0.5}
        allow_stretch: True
        keep_ratio: False
        size_hint_y: None
        height: dp(200)
        size_hint_x: None
        width: dp(120)
    MDRectangleFlatButton:
        text: 'Start Over'
        pos_hint: {'center_x':0.5,'center_y':0.1}
        on_press:
            root.manager.current = 'start'
            root.manager.transition.direction = "right"
        text_color: (1, 1, 1, 1)
        md_bg_color: (1, 0, 0, 1)
```



UC-03

Use Case: Start Survey Over

ID: UC-03

Description: User wants to start over when the wrong options were chosen

Primary Actor: Hungry User

Precondition: WTE Application is open

Post Conditions: User wants to start over

Success end conditions:

User presses start over button and is navigated to the start page

Failure end conditions:

User selects button and they are not directed to the start page(whether it be the wrong page or no actions happen at all)

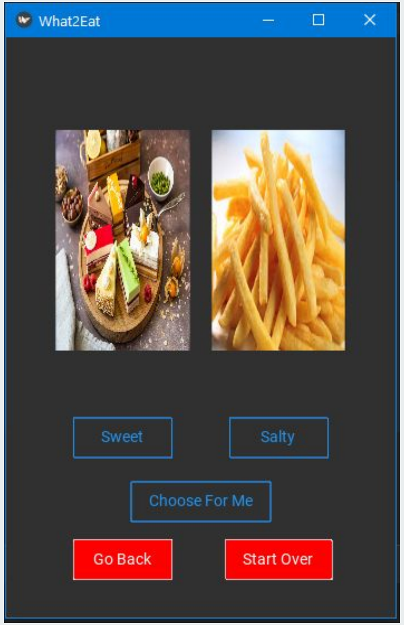
Trigger: User realizes they don't like the options they've chosen

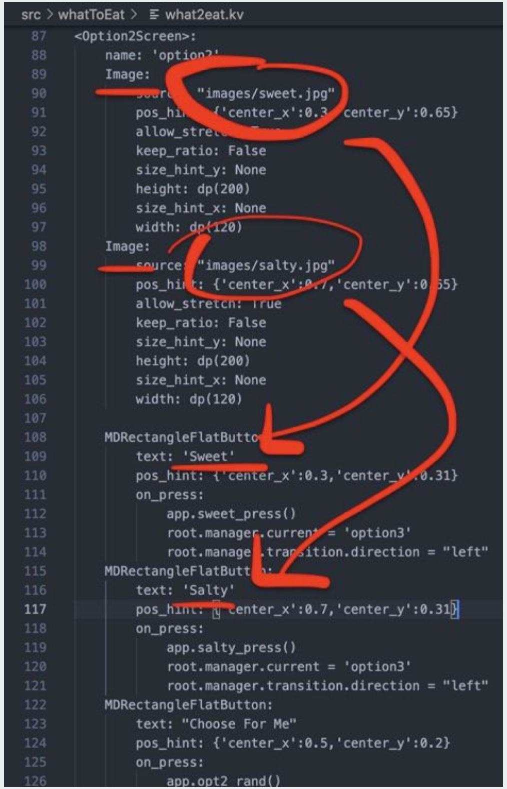
Main Success Scenario:

1. User opens application
2. User presses the start button
3. System navigates to the first question
4. The user picks out of the two options displayed
5. System navigates to the next question
6. Steps 4 and 5 are repeated until user realizes they dislike the options they have chosen
7. User presses start over button
8. System returns to the start screen

Functional:

4. *As a WTE user; I want to see images of what I am choosing; So that I better understand what my options are*




src > whatToEat > what2eat.kv

```
<Option2Screen>:
    name: 'option2'
    Image:
        source: "images/sweet.jpg"
        pos_hint: {'center_x':0.3,'center_y':0.65}
        allow_stretch: True
        keep_ratio: False
        size_hint_y: None
        height: dp(200)
        size_hint_x: None
        width: dp(120)
    Image:
        source: "images/salty.jpg"
        pos_hint: {'center_x':0.7,'center_y':0.65}
        allow_stretch: True
        keep_ratio: False
        size_hint_y: None
        height: dp(200)
        size_hint_x: None
        width: dp(120)

    MDRectangleFlatButton:
        text: 'Sweet'
        pos_hint: {'center_x':0.3,'center_y':0.31}
        on_press:
            app.sweet_press()
            root.manager.current = 'option3'
            root.manager.transition.direction = "left"
    MDRectangleFlatButton:
        text: 'Salty'
        pos_hint: {'center_x':0.7,'center_y':0.31}
        on_press:
            app.salty_press()
            root.manager.current = 'option3'
            root.manager.transition.direction = "left"
    MDRectangleFlatButton:
        text: "Choose For Me"
        pos_hint: {'center_x':0.5,'center_y':0.2}
        on_press:
            app.opt2_rand()
```

UC-04

Use Case: See Image Decisions

ID: UC-04

Description: User wants to see the images of the decision chosen

Primary Actor: Hungry User

Precondition: WTE Application is open

Post Conditions: Getting the image of the chosen decisions

Success end conditions:
The user is satisfied with the chosen images provided

Failure end conditions:
The user is unable to see the images of the provided decision

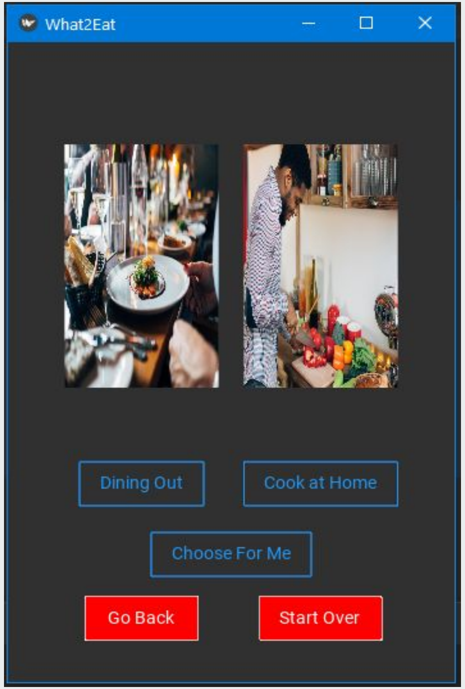
Trigger: User wants a better definition of the option they are choosing

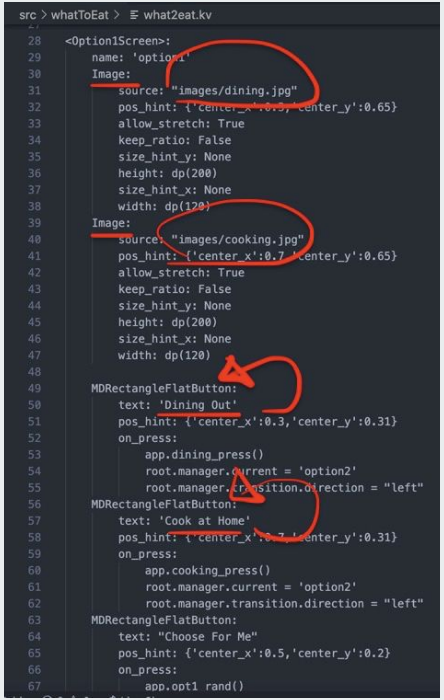
Main Success Scenario:

1. User opens the application
2. User presses the start button
3. System provides the images provided with decisions
4. User clicks on the decisions that is desired
5. System provides the next set of images of the decisions chosen
6. User finishes choosing the decisions
7. System shows images chosen from set of decisions chosen by user
8. System returns to start screen

Functional:

5. *As a WTE user; I want to choose to eat out or cook at home; So I can get recommendations for recipes or outside food recommendations*





src > whatToEat > what2eat.kv

```
<Option1Screen>:
    name: 'option1'
    Image:
        source: "images/dining.jpg"
        pos_hint: {'center_x':0.3,'center_y':0.65}
        allow_stretch: True
        keep_ratio: False
        size_hint_y: None
        height: dp(200)
        size_hint_x: None
        width: dp(120)
    Image:
        source: "images/cooking.jpg"
        pos_hint: {'center_x':0.7,'center_y':0.65}
        allow_stretch: True
        keep_ratio: False
        size_hint_y: None
        height: dp(200)
        size_hint_x: None
        width: dp(120)

    MDRectangleFlatButton:
        text: 'Dining Out'
        pos_hint: {'center_x':0.3,'center_y':0.31}
        on_press:
            app.dining_press()
            root.manager.current = 'option2'
            root.manager.transition.direction = "left"
    MDRectangleFlatButton:
        text: 'Cook at Home'
        pos_hint: {'center_x':0.7,'center_y':0.31}
        on_press:
            app.cooking_press()
            root.manager.current = 'option2'
            root.manager.transition.direction = "left"
    MDRectangleFlatButton:
        text: "Choose For Me"
        pos_hint: {'center_x':0.5,'center_y':0.2}
        on_press:
            app.opt1_rand()
```

UC-05

Use Case: Choose eating at home or dining out

ID: UC-05

Description: User wants to be able to eat at home

Primary Actor: Hungry User

Precondition: WTE Application is open

Post Conditions: The user gets to pick a food option that they can make at home

Success end conditions:
The user chooses an at home option

Failure end conditions:
The user is not able to choose an option to eat at home

Trigger: User wants to eat at home and be given recipes OR user does not want to cook at home and be given suggestions from restaurants

Main Success Scenario: capture user and system interactions

1. User opens the application
2. System opens up to the home page
3. User presses the cooking at home option button
4. System provides the cooking at home options