

## Code reviews feedbacks

The code reviewed was main\_menu.py written by Inkaphol S. and reviewed by Sira K. and Shine

The main feedback that was given were

1. The program only allows non duplicate names and ingredients.
2. Symbol inputs are not allowed.
3. Symbols not on default keyboards should also be validated.

```
201     # formats input text, used for steps
202     def format_text(self, in_string, line_length):
203         common_symbols = ['.', ',', '?', '!', ';', ':', '-', '"', "'", '(', ')', '[', ']',
204             '{', '}', '&', '@', '#', '$', '%', '^', '*', '+', '-', '=', '>', '<']
205
206         out_string = ""
207         curr_line = ""
208         line_num = 1
209
210         for i in range(len(in_string)):
211             if (in_string[i - 1].isalnum() or in_string[i - 1] in common_symbols) and in_string[i] == ' ':
212
213                 if int(len(curr_line) / line_length) >= line_num:
214                     out_string += curr_line + "\n"
215                     curr_line = ""
216                     line_num += 1
217                     continue
218
219             if (in_string[i].isalnum()) or (in_string[i] in common_symbols) or (in_string[i] == ' '):
220                 curr_line += in_string[i]
221
222             if curr_line:
223                 out_string += curr_line
224
225         return out_string
```

4. update\_suggestions updates the valid list of ingredients.

```
# updates the suggestion list according to the ingredient_name_in
def update_suggestions(self):

    in_string = self.ingredient_name_in.toPlainText()
    self.suggestion_list.clear()

    matching_suggestions = [suggestion for suggestion in self.valid_ingredients if in_string.lower() in suggestion.lower()]

    if matching_suggestions and self.ingredient_name_in.toPlainText() != "":
        for suggestion in matching_suggestions:
            self.suggestion_list.addItem(suggestion)
            self.suggestion_list.show()
    else:
        self.suggestion_list.hide()
```

5. There should be a character limit for recipe steps.

```
226         # adds the text within step_in into the all_steps
227     ✓     def add_to_step_list(self):
228         formatted_text = self.format_text(self.step_in.toPlainText(), 60)
229         if formatted_text:
230             self.step_list.addItem(formatted_text)
231             self.step_in.clear()
```