CCD - Clean Code Development

1. **Readability :**
   The code is easy to read and understand because the variable and function names are self-explanatory. The functions purpose is communicated through meaningful names such as ‘new_booking’ and ‘cancel_booking’.

2. **Modularity:**
   The code is organized into functions called "new_booking," "cancel_booking," and "get_ground_name" which helps to facilitate the encapsulation of functionality and promote modularity.

3. **Comments:**
   While comments are minimal, they provide explanations where necessary, aiding understanding without cluttering the code.

4. **User Input Handling:**
   To increase robustness, the algorithm verifies user inputs for the kind and range of permissible values. Users are able to easily understand and learn from the input prompts.

5. **Encapsulation of External Dependencies:**
   External dependencies, such as email sending using SMTP, are encapsulated within functions, promoting separation of concerns.

6. **Error Handling:**
   The code includes error handling for scenarios where the user provides invalid inputs, preventing unexpected crashes.

7. **Sensible Break Conditions:**
   The use of `break` statements within loops helps exit the loop when necessary, improving control flow.

8. **Use of Meaningful Output:**
   The code provides meaningful output messages, such as "Booking successful" and "Cancellation successful," allowing users to easily understand the outcome of their actions.