

Design and Testing for Question 2

Design Description

The program is designed to stimulate a waiting list management system for an event ticket selling company which will be only used by the company staff. The program allows to maximum ten events have a waiting list at the same time. Furthermore, each waiting list can contain no more than ten people.

When a staff starts the program, the system will ask the staff to input maximum ten names for ten events which need a waiting list. Then the program will provide the main menu for the staff to choose: option 1 to print out a list of the events which have a waiting list and the current number of people in each waiting list. Option 2 is to quit the program.

If choosing the option 1, after the list of events have been printed out, the staff is prompt to choose a specific event to manage its waiting list. After the staff have chosen a particular event, the program will display the second menu with a list of operations that the staff can operate to manage the waiting list for this chosen event. The operations include the functionalities to add a new person into the waiting list, delete the first person from the list, check a person's position in the list, reset the list to empty and return to the main menu. The program allows the staff to keep choosing different options to operate unless the staff choose to quit the program.

Design Rationale

The designer of the program decided to use two main classes, customer class and event class. Customer class is simply used for storing a person's surname and phone number who will be put into the waiting list of the event class later as an object. In the customer class, private members include a name and a phone number with corresponding set and get functions for the private members.

The event class is the main class which is used to perform various operations mentioned in the design description. In the private member, the name for an event is stored and a waiting list is created for this event. The designer used the template queue (a dynamically allocated array) which is provided by the lecturer to act as a waiting list. Public functions include the operations to manipulate the waiting list, including add, delete, search and reset. The designer also includes a function to keep tracking on the number of element in the waiting list so that it can be presented as part of information when the staff chooses to print out the list of events with waiting lists.

Additional functions include showing the main (first) menu, the second menu and a function named 'operations for event' to arrange and call the specific functions from the event class in order to improve the code reusability and reduce repetition of blocks of codes.

Testing Strategy

The testing strategy the designer applied is to check if the system will give the results the designer expected in every possible path by inputting various data and make different choices to try out all the paths. The testing strategy also takes potential error handling into consideration. So at the testing stage, the designer will deliberately enter some wrong format input, choose something which is not presenting in the menu, add a new person into a full waiting list, delete a person record from an empty list or search someone who is not in the list. By deliberately making these kind of mistakes, the designer can test how the program will cope with these to ensure it works robustly. Additionally, the designer has tested the results for each block of codes to ensure it works before putting each section of codes together as a completed program.

Test Results

Testing different paths:

- **Operation – choosing from the first menu**
 - Choosing to show all the waiting lists
 - E.g. scenario 1: showing the lists with only ten event names being initialised, the result looks like this:
The following events have waiting lists:

1. event1	0 people in the list.
2. event2	0 people in the list.
3. event3	0 people in the list.
4. event4	0 people in the list.
5. eventt5	0 people in the list.
6. event6	0 people in the list.
7. event7	0 people in the list.
8. event8	0 people in the list.
9. event9	0 people in the list.
10. event10	0 people in the list.

Then the system displayed the message “Please enter the corresponding number to get access to the waiting list for the specific event.” to prompt the user to choose a list to work on.

- E.g. scenario 2: showing the lists with some updated data and event names after some management operations performed on data in the waiting lists:
The following events have waiting lists:

1. event1	1 people in the list.
2. PinkFloyd	3 people in the list.
3. event3	0 people in the list.
4. event4	0 people in the list.
5. event5	0 people in the list.
6. event6	0 people in the list.

- 7. event7 0 people in the list.
- 8. event8 0 people in the list.
- 9. event9 0 people in the list.
- 10. event10 0 people in the list.

Similarly, the system then displayed the same message to prompt the user to choose a list to work on.

- Choosing to quit the program
 - E.g. when I pressed 2, the system displayed “Thank you for using the program” to notify me about this successful operation.
- Choosing something is not in the menu
 - E.g. when I pressed 3, 6 or even a letter, the program displayed the message “Sorry, the system cannot recognise your choice. Try again!” to prompt me about this wrong input and allow me to try again until I make a proper choice.
- **Operation – choosing from the second menu (once the user chose which particular waiting list to work on)**
 - Choosing option 1
 - This will redirect to the operation to add a new record for this event list.
 - Choosing option 2
 - This will redirect to the operation to remove the first person in the list.
 - Choosing option 3
 - This will redirect to the operation to search for a person’s position in the list.
 - Choosing option 4
 - This will redirect to the operation to reset the list
 - Choosing option 5
 - This will redirect to the operation to return to the first menu
 - Choosing option 6
 - This will redirect to the operation to quit the program with the message “Thank you for using the program.”
 - Choosing something is not on the menu
 - Similar error handling method as for the first menu, any undefined option entered by me triggered this message “Sorry, the system cannot recognise your choice. Try again!”
- **Operation – adding a new record**
 - Adding in when a waiting list is not full
 - E.g. adding data ‘person1’ ‘111’ into the list and it displayed “A new person has been added in successfully”
 - Adding in when the list is full
 - I deliberately initialised 10 customer objects for adding into an event list by using a for loop iterating 10 times and then trying to perform this operation. The program displayed “Sorry the waiting list for this event is already full. You cannot add any more to the list.”

- **Operation – removing the first person from the list**
 - Removing the first person from a list when there are some records in the list
 - E.g. the first person 'person 1' '111' which had been added before will be removed from the list. The program displayed "A ticket now is available for person 1 whose phone number is: 111. So this person has been removed from the waiting list" when this operation had been completed.
 - Removing the first person from a list when it is an empty list
 - I first chose the operation 4 to reset the list to empty and then choose this operation. The message "Sorry, this is an empty waiting list." displayed.
 - Similarly, I restarted the program with an empty list and then choose to remove a record, the same message displayed.

- **Operation – search for a person and return their position in the queue**
 - A person has been found
 - E.g. I added five people in the list and search for the second person with his surname 'person2'. This was what returned: "person2 position 2 of the waiting list."
 - A person's name is not in the list
 - E.g. I entered a random name 'nobody' to search for his position, this message returned: "Sorry, no record is found. The person you want to search is not in the waiting list."
 - E.g. I also tried to enter a random name when the list has been reset to an empty list, the same message displayed on the screen.

- **Operation – reset the waiting list to empty**
 - Successful operation
 - E.g. When choosing this operation, this message would appear first "The waiting list is about to reset, please enter a new name of the event to create a new waiting list." Once a new event name has been inserted, the program will show "The waiting list has been reset."
 - And then I tried the operation of removing a record, the program also displayed "Sorry, this is an empty waiting list." This confirmed the operation of reset worked.
 - If not success
 - E.g. if failed, this message will appear "Something went wrong and the operation failed."

- **Operation – repeat the second menu**
 - The second menu displays again after one specific operation completed
 - E.g. after I added a new record into the waiting list for the 'event1', the second menu appears again to allow me to do more work on the waiting list of the chosen event until I choose to return to the first menu with updated data of the chosen event or quit the program.