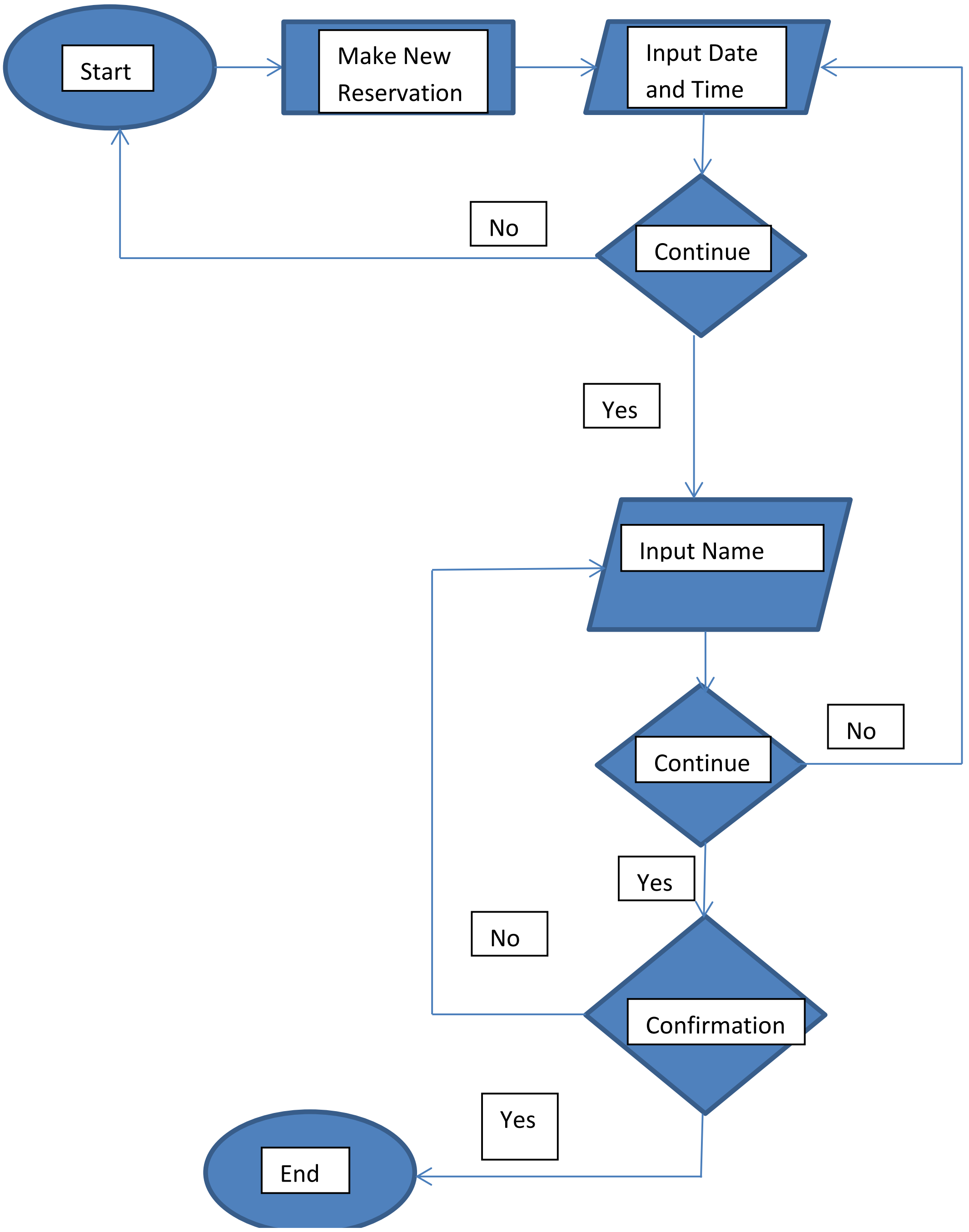
Criterion B: Design

Flowcharts

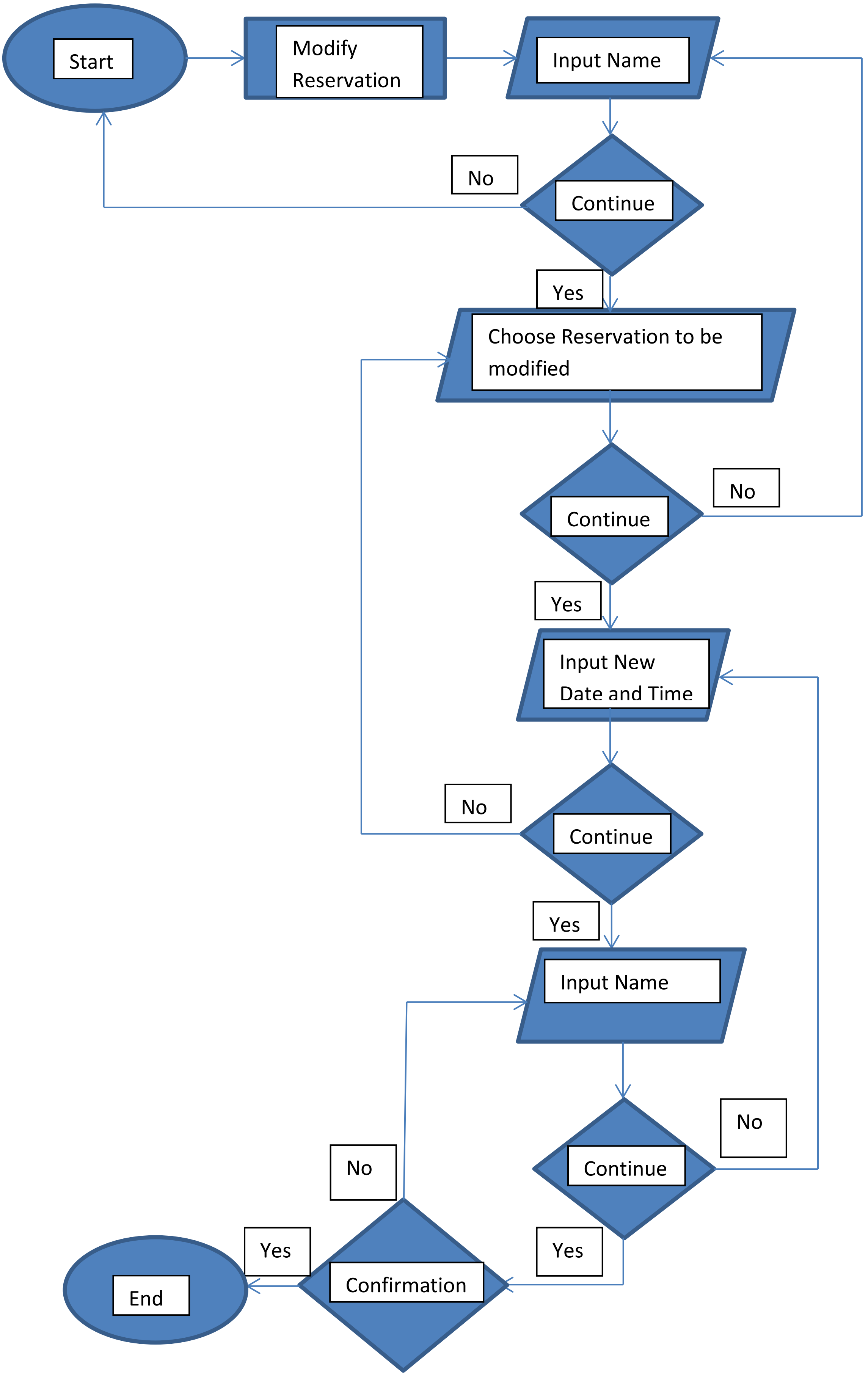
The two possible methods of making a new reservation are:

- Selecting a room and then receiving all the available times for the room.
- Selecting a Date/Time and receiving a list of the rooms available at the time.

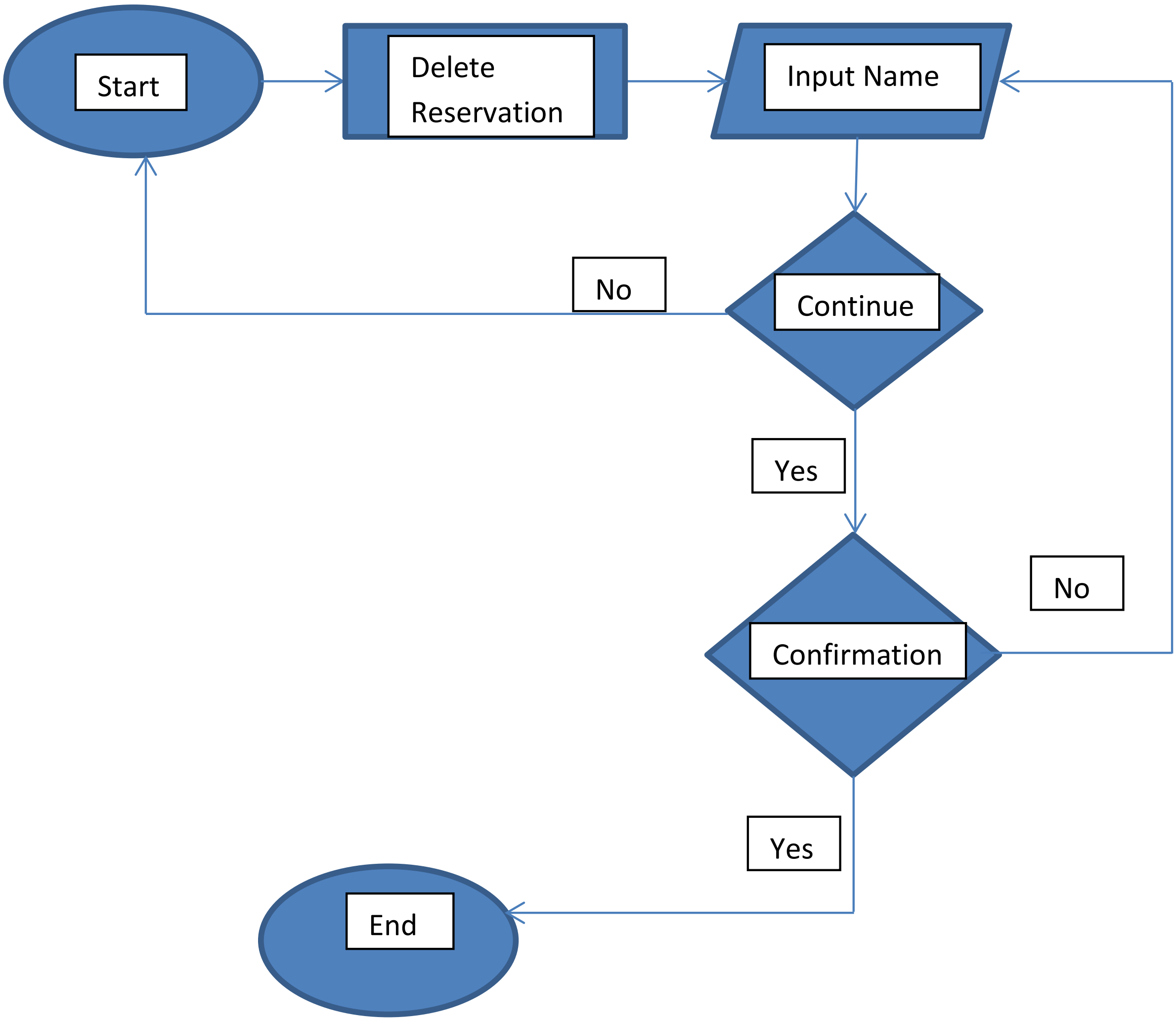
However, due to time constraints the complexity of the method, neither of these methods was chosen, and the program was created for the reservation of a single room. The flowchart below will outline the logic behind making a new reservation.



A similar flowchart was created to outline the logic behind modifying a reservation.



A final flowchart was created to outline the logic behind cancelling a reservation.



System Design (Classes with their important variables)

Classes – NewRes, NewRes3

These two classes represent the technique of making a new reservation. The user will need to enter the date and time of the reservation. If the room is available at the selected date and time, the user will enter his name, which will be saved along with the date and time selected after the reservation is confirmed.

Variable Name	Data Type	Description
ContinueButton	jButton	Go to next GUI screen
BackButton	jButton	Go to previous GUI screen
jDateChooser1	jDateChooser	Input date (not manually)
StartTimeText	jTextField	Enter start time
EndTimeText	jTextField	Enter end time
NameText	jTextField	Enter name of employee making reservation

Classes – ModRes, ModRes3, ModRes5

These three classes represent the technique of modifying an existing reservation. The user will need to enter his name, and his reservation will be obtained from the list of existing reservations. Then, similar to the “new reservation” process, the user will need to enter the new date and time of the reservation. If the room is available at the selected date and time, the user will enter his name, which will be saved along with the new date and time selected after the reservation is confirmed.

Variable Name	Data Type	Description
ContinueButton	jButton	Go to next GUI screen
BackButton	jButton	Go to previous GUI screen
jDateChooser1	jDateChooser	Input date (not manually)
StartTimeText	jTextField	Enter start time
EndTimeText	jTextField	Enter end time
NameText	jTextField	In ModRes – Enter name of employee who has made existing reservation in order to search for the existing reservation In ModRes4 - Enter new name (if modified) of employee making reservation

Classes – CanRes, CanRes3

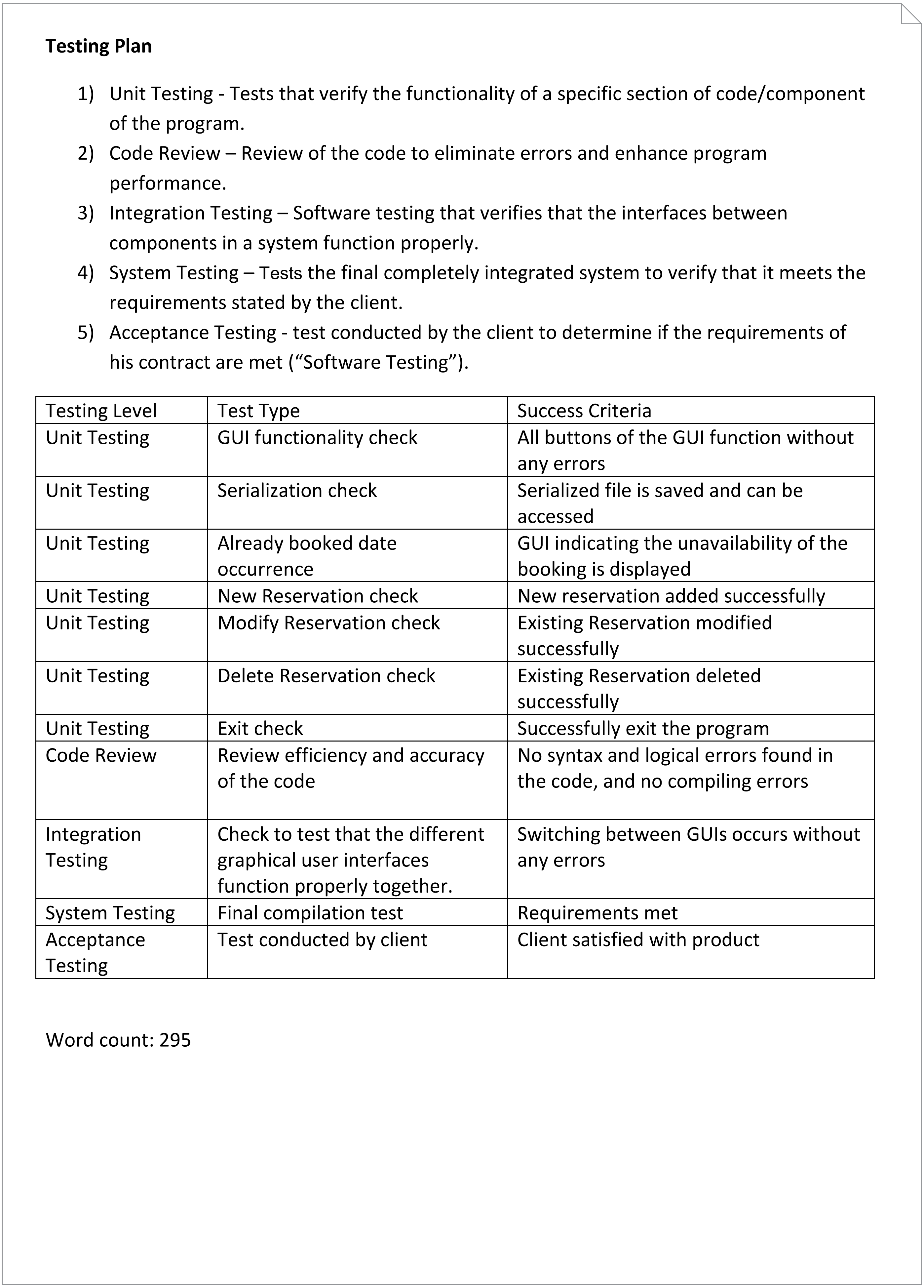
These two classes represent the technique of deleting an existing reservation. The user will need to enter his name, and his reservation will be obtained from the list of existing reservations. Then, after confirmation from the user, the reservation will be deleted.

Variable Name	Data Type	Description
ContinueButton	jButton	Go to next GUI screen
BackButton	jButton	Go to previous GUI screen
NameText	jTextField	Enter name of employee who has made existing reservation in order to search for the existing reservation

Class – NR

This class is used to perform the serialization process.

Variable Name	Data Type	Description
strField	String	To serialize Date
strField2	String	To serialize Name of User
intField	int	To serialize Start Time
intField2	int	To serialize End Time
hours	ArrayList	To store times of all reservations
dates	ArrayList	To store dates of all reservations



Testing Plan

1) Unit Testing - Tests that verify the functionality of a specific section of code/component of the program.
2) Code Review – Review of the code to eliminate errors and enhance program performance.
3) Integration Testing – Software testing that verifies that the interfaces between components in a system function properly.
4) System Testing – Tests the final completely integrated system to verify that it meets the requirements stated by the client.
5) Acceptance Testing - test conducted by the client to determine if the requirements of his contract are met (“Software Testing”).

Testing Level	Test Type	Success Criteria
Unit Testing	GUI functionality check	All buttons of the GUI function without any errors
Unit Testing	Serialization check	Serialized file is saved and can be accessed
Unit Testing	Already booked date occurrence	GUI indicating the unavailability of the booking is displayed
Unit Testing	New Reservation check	New reservation added successfully
Unit Testing	Modify Reservation check	Existing Reservation modified successfully
Unit Testing	Delete Reservation check	Existing Reservation deleted successfully
Unit Testing	Exit check	Successfully exit the program
Code Review	Review efficiency and accuracy of the code	No syntax and logical errors found in the code, and no compiling errors
Integration Testing	Check to test that the different graphical user interfaces function properly together.	Switching between GUIs occurs without any errors
System Testing	Final compilation test	Requirements met
Acceptance Testing	Test conducted by client	Client satisfied with product

Word count: 295