Ticket Reservation System Code

This document contains the Python code for the Ticket Reservation System with the following features:
1. Add Reservation
2. Process Reservation
3. Cancel Reservation
4. View Queue
5. Clear Queue
6. Search Reservation
7. Save Queue
8. Load Queue
9. Reservation Time

```
import tkinter as tk
from tkinter import messagebox
import time
from datetime import datetime

# Initialize an empty queue for ticket reservations
ticket_queue = []

# Function to add a reservation to the queue
def add_reservation():
    name = entry_name.get()
    if name.strip():
        timestamp = datetime.now().strftime("%Y-%m-%d %H:%M:%S")
        ticket_queue.append((name, timestamp))
        entry_name.delete(0, tk.END)
        update_queue_display()
        messagebox.showinfo("Success", f"Reservation for '{name}' added!")
    else:
        messagebox.showwarning("Input Error", "Please enter a name!")

# Function to process the next reservation
def process_reservation():
    if ticket_queue:
        processed_name, _ = ticket_queue.pop(0)
        update_queue_display()
        messagebox.showinfo("Processed", f"Reservation for '{processed_name}'
```

```
processed!")
    else:
        messagebox.showwarning("Queue Empty", "No reservations to process!")

# Function to cancel a specific reservation
def cancel_reservation():
    name = entry_name.get()
    for reservation in ticket_queue:
        if reservation[0] == name:
            ticket_queue.remove(reservation)
            entry_name.delete(0, tk.END)
            update_queue_display()
            messagebox.showinfo("Cancelled", f"Reservation for '{name}' cancelled!")
            return
    messagebox.showwarning("Not Found", f"Reservation for '{name}' not found!")

# Function to clear all reservations
def clear_queue():
    global ticket_queue
    ticket_queue = []
    update_queue_display()
    messagebox.showinfo("Cleared", "All reservations cleared!")

# Function to search for a reservation
def search_reservation():
    name = entry_name.get()
    for reservation in ticket_queue:
        if reservation[0] == name:
            messagebox.showinfo("Found", f"Reservation for '{name}' exists!")
            return
    messagebox.showwarning("Not Found", f"Reservation for '{name}' not found!")

# Function to save the queue to a file
def save_queue():
    if ticket_queue:
        with open("ticket_reservations.txt", "w") as file:
            for name, timestamp in ticket_queue:
                file.write(f"{name},{timestamp}\n")
        messagebox.showinfo("Saved", "Queue saved to 'ticket_reservations.txt'")
    else:
        messagebox.showwarning("Queue Empty", "No reservations to save!")

# Function to load the queue from a file
```

```
def load_queue():
    global ticket_queue
    try:
        with open("ticket_reservations.txt", "r") as file:
            ticket_queue = [tuple(line.strip().split(',')) for line in file]
        update_queue_display()
        messagebox.showinfo("Loaded", "Queue loaded from 'ticket_reservations.txt'")
    except FileNotFoundError:
        messagebox.showerror("Error", "File 'ticket_reservations.txt' not found!")

# Function to update the queue display
def update_queue_display():
    queue_display.delete(0, tk.END)
    for idx, (name, timestamp) in enumerate(ticket_queue, start=1):
        queue_display.insert(tk.END, f"{idx}. {name} ({timestamp})")

# Create the GUI
root = tk.Tk()
root.title("Ticket Reservation System")

# Input field for name
label_name = tk.Label(root, text="Customer Name:")
label_name.pack(pady=5)
entry_name = tk.Entry(root, width=30)
entry_name.pack(pady=5)

# Buttons for various operations
btn_add = tk.Button(root, text="Add Reservation", command=add_reservation)
btn_add.pack(pady=5)

btn_process = tk.Button(root, text="Process Reservation",
command=process_reservation)
btn_process.pack(pady=5)

btn_cancel = tk.Button(root, text="Cancel Reservation", command=cancel_reservation)
btn_cancel.pack(pady=5)

btn_search = tk.Button(root, text="Search Reservation", command=search_reservation)
btn_search.pack(pady=5)

btn_clear = tk.Button(root, text="Clear All Reservations", command=clear_queue)
btn_clear.pack(pady=5)
```

```
btn_save = tk.Button(root, text="Save Queue", command=save_queue)
btn_save.pack(pady=5)

btn_load = tk.Button(root, text="Load Queue", command=load_queue)
btn_load.pack(pady=5)

# Listbox to display the queue
label_queue = tk.Label(root, text="Current Reservations:")
label_queue.pack(pady=5)
queue_display = tk.Listbox(root, width=50, height=10)
queue_display.pack(pady=5)

# Run the GUI
root.mainloop()
```