```
import numpy as np

def lift_alert_system(stuck_position, total_floors):
    floors = np.arange(total_floors + 1)  # Floors from 0 to total_floors

    # Calculate floor just below and just above the stuck position
    down_floor = int(np.floor(stuck_position))
    up_floor = int(np.ceil(stuck_position))

    # Calculate distances to each floor
    distances = np.abs(floors - stuck_position)

    # Get indices of two nearest floors
    nearest_two_indices = distances.argsort()[:2]
    nearest_two_floors = floors[nearest_two_indices]

    # Display alert info
    print(f"\n  Lift is stuck at position: {stuck_position}")
    print(f" Nearest two floors: {sorted(nearest_two_floors)}")
    print(f" Up floor: {up_floor if up_floor <= total_floors else 'N/A'}")
    print(f" Down floor: {down_floor if down_floor >= 0 else 'N/A'}")

    # Simulate sending alert
    print("\n Sending alerts to nearest floors:")
    for floor in sorted(nearest_two_floors):
        print(f"  - Alert sent to Floor {floor}")
    if down_floor >= 0 and down_floor not in nearest_two_floors:
        print(f"  - Alert sent to Down Floor {down_floor}")
    if up_floor <= total_floors and up_floor not in nearest_two_floors:
        print(f"  - Alert sent to Up Floor {up_floor}")


# Example usage
if __name__ == "__main__":
    stuck_position = 3.7
    total_floors = 10
    lift_alert_system(stuck_position, total_floors)
```