**Input Validation**

**Purpose:**  
Ensure users enter only correct and expected data into the system.

**Implementation Strategies:**

* **Email Validation** using Regular Expressions.
* **Phone Number Validation** to check for exactly 10 digits.
* **Employee ID Uniqueness**: Prevent creating duplicate employee records.
* Trimming input to remove unnecessary spaces.

Sample Code

if (!email.matches("^[A-Za-z0-9+\_.-]+@(.+)$")) {

System.out.println("Invalid email format. Please try again.");

return;

}

if (!contact.matches("\\d{10}")) {

System.out.println("Contact must be exactly 10 digits.");

return;

}

**Accuracy of Output**

**Purpose:**Make sure the information shown to users is consistent and reliable.

**Implementation Strategies:**

* Always read back the file after update to confirm the change.
* Uniform format in all display outputs.
* Use meaningful and structured confirmation messages.

**Error Messages and Feedback**

* **Purpose:**
* Help users understand what went wrong and how to fix it.
* **Implementation Strategies:**
* Use specific catch blocks for different exceptions (e.g., FileNotFoundException, IOException).
* Show messages in plain language with suggestions.
* Include relevant info like employee ID or name for clarity.

**EXAMPLE:**

catch (FileNotFoundException e) {

System.out.println("Employee record not found. Please check the ID and try again.");

}

catch (IOException e) {

System.out.println("An error occurred while accessing the file. Please try again later.");

}