**Architecture Plan**

**Components and Their Responsibilities**

1. **main.py (Central Controller)**
   * Displays an interactive menu for user input.
   * Calls functions from other modules based on user choices.
   * Acts as the entry point for the application.
2. **Modules (Feature-Specific Files)**
   * **addContacts.py**:
     + Adds a new contact.
     + Validates user input of phone number.
     + Prevents duplicate phone numbers.
   * **viewContacts.py**:
     + Displays all saved contacts in a certain format that is easy to understand.
   * **deleteContact.py**:
     + Deletes a contact based on the user's input (name).
   * **searchContact.py**:
     + Searches for contacts based on specific criteria (e.g., name, email, phone).
   * **saveContacts.py**:
     + Saves updated contact data to a file (contacts.txt).
   * **loadContacts.py**:
     + Loads existing contact data from the file into memory when the program starts.
3. **Data Storage**
   * **contacts.txt**:
     + Stores contact data persistently in a text file in a certain format.

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

**Interaction Flow**

1. **Start (User Launches main.py)**
   * The program loads contacts from contacts.txt using loadContacts.py.
   * Displays a menu with options:
     + Add a Contact
     + View Contacts
     + Delete a Contact
     + Search for a Contact
     + Exit
2. **User Actions**:
   * Based on the menu choice:
     + Calls the corresponding module.
     + Updates or retrieves contact data.
3. **Data Storage**:
   * Contact data is read from and written to contacts.txt for persistence.

### ****Diagram****

Here’s a visual presentation of the architecture (design\_plan.png):

### ****F:\Yusuf\pyPrograms\Ostad\contactBookManagement_myCopy\design_plan.png****

### ****File Structure****

### 