# **Assignment 1: Part 1 of 3**

## A brief outline of the application

For this assignment, I have chosen to design a **Contact Book** application. The data structure and algorithm design are described in this document.

**Objective:**

Primary objective of that application is to provide convenient and efficient way to store, manage and display contact information. It will include names, surnames, phone numbers, birthday dates, e-mail and home addresses. As a neurodiverse person (ADHD), I find this tool being beneficial in everyday life, simply because of problems with remembering such details.

**Purpose:**

The purpose of the Contact Book is to keep contacts details organized in one place. To be more, make it easier for users to retrieve and manage their saved contacts. Whether it is for personal or professional use, the application aim is to enhance productivity and minimize hassle of asking for or searching contact details. That tool streamlines the process of staying connected, while ensuring that important information is always available on hand.

The application will be executing instructions such as *inserting new contacts*, *deleting*, *updating*, *sorting* and *searching* *through data* – with the Use Case diagram below (Figure 1). User can launch the application, select the action from main menu, provide the required input data and the application will process the input and display the results or confirmation. After that, the steps could be repeated for additional actions or exiting the application.

A diagram of a user

Description automatically generated

**Figure 1** - Use Case diagram

## Design of data structure and algorithms

**Attributes in Cases:**

* *Name, surname, phone number, birthday date, e-mail & address.*

**Actor:**

* **User:** A person that interacts with the Contact Book.

Cases: *insertion, deletion, update, sorting, or searching of record.*

A diagram of a flowchart

Description automatically generated

**Figure 2** - Insert Record

1. **Insert Record (Figure 2)**
   1. **Description:** The user adds a new contact by providing values for the following fields:

Name, Surname, Phone Number, Birthday Date, Email, and Address.

* 1. **Steps:**
* User fills out the required fields on the "Add Contact" form.
* User clicks "Save."
* The system validates the input and:

1. If the input is invalid, it displays an error message.
2. If the input is valid, it saves the contact to the database and shows success.

* After saving, the updated contact book is displayed.
  1. **Postconditions:** The contact is visible in the contact list.

A diagram of a flowchart

Description automatically generated

1. **Delete Record (Figure 3)**
   1. **Description:** User deletes the record from contact book by selecting one or more contacts from the list.
   2. **Steps:**

* User selects one or more contacts from the list.
* User selects ‘Delete’ button.
* The system confirms the deletion of the contact/s.
* After confirmation, the records are removed from the database.
* Updated contact book is displayed.
  1. **Postconditions:** The contact/s is/are deleted from the contact book.

**Figure 3** - Delete Record

**Figure 4** - Update Record

1. A diagram of a software flowchart

   Description automatically generated**Update Record (Figure 4)**

3.1. **Description:** User selects the record that they want to update, providing changes in any of the fields and saves it.

3.2. **Steps:**

* + User selects a contact to update from the list.
  + User changes the specified field.
  + User selects ‘Save’ button.
  + System confirms the update of the data.
  + After confirmation, the record is being updated in the database.
  + The updated contact book is being displayed.

3.3. **Postconditions:** The contact is updated with the new data.

**Figure 5** - Sort Records

1. A diagram of a software process

   Description automatically generated**Sort Record (Figure 5)**
   1. **Description:** User chooses to sort all the records by choosing the right sorting value, that will be alphabetically name or surname, or chronically by birthday date.
   2. **Steps:**
   * User selects the sorting value (alphabetically name or surname, chronically birthday date).
   * The system reorders the data based on chosen sort value.
   * Sorted contact book is displayed.
   1. **Postconditions:** The data was reordered based on the sort value order.

A diagram of a computer process

Description automatically generated

1. **Search Record (Figure 6)**
   1. **Description:** User searches for a record using a query, that will be a name, surname or e-mail address.
   2. **Steps:**
   * User enters the value such as name, surname or e-mail address in search bar.
   * The system filters the contacts based on the query.
   * All the matching contacts are displayed / if no matches found, system displaying “No Contacts Found”.
   1. **Postconditions:** Relevant contacts are being shown in the search results.

**Figure 6** - Search Record

## Test plan

Dummy data:

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| **CONTACT ID** | **NAME** | **SURNAME** | **PHONENUMBER** | **BIRTHDAYDATE** | **EMAIL** | **ADDRESS** |
| 1 | David | Smith | 07513476290 | 2000-01-01 | david.smith@example.co.uk | E12QQ |
| 2 | John | Doe | 07754321455 | 1995-05-24 | john.doe@example.co.uk | NW25TG |
| 3 | Angelina | Parker | 07534565432 | 2005-12-15 | angelina.parker@example.co.uk | SW12DT |

1. **Insert Record Test**

*Action:* **INSERT INTO**Contacts (Name, Surname, PhoneNumber, BirthdayDate, Email, Address)**VALUES**(‘Jessica', ‘Gordon’, '07343212344', '2000-12-12', 'jessica.gordon@example.co.uk', 'RM13TD');

*Expected Result:* Contact was added to database.

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| **CONTACT ID** | **NAME** | **SURNAME** | **PHONENUMBER** | **BIRTHDAYDATE** | **EMAIL** | **ADDRESS** |
| 4 | Jessica | Gordon | 07343212344 | 2000-12-12 | jessica.gordon@example.co.uk | RM13TD |

1. **Delete Record Test**

*Action:* **DELETE FROM**Contacts WHERE ContactID = 1;

*Expected Result:* Record with ContactID 1 (David Smith) is removed.

1. **Update Record Test**

*Action:* **UPDATE**Contacts **SET**PhoneNumber = '74231234564'**WHERE**ContactID = 2;

*Expected Result: John Doe’s phone number updates to 74231234564.*

1. **Filter Record Test**

*Action:* **SELECT**\* **FROM**Contacts**ORDER BY**Name**ASC;**

*Expected Result:* The contact book list is displayed in an ascending form from A-Z, based on the contact names.

1. **Search Record Test**

*Action:* **SELECT** \* **FROM** Contacts **WHERE** Name = 'Charlie';

*Expected Result:* No rows returned, with message saying, *"No Results".*