**Launching into Computer Science March 2022 Assignment 1 Part 1**

1. **Purpose**

This phone book application was created to allow the user to store people’s contact information and details.

1. **Technical specification**

This application was written using Python 3 and Django 4.

1. **Features**

* The application allows the user to add a new person with the following information
  + First Name
  + Last Name
  + Phone Number
  + Email
* The application allows the user to remove any person from their list
* The application allows the user to sort people both in ascending and descending order. The user may sort the data with the following fields
  + First Name
  + Last Name
  + Phone Number
  + Email
* The application allows the user to search for any person with the following fields
  + First Name
  + Last Name
  + Phone Number
  + Email

1. **Executing the application**

To execute and run the application follow the steps below

* 1. Clone the project using Git
     1. <https://github.com/mustafa-sibai-essex/LCS_PCOM7E_A1_P1>
  2. Download and install Python 3
  3. Download and install Django by running the following command
     1. python -m pip install Django
  4. Navigate to the project directory and run the following command
     1. Python manage.py runserver
  5. Open chrome web browser and enter the following URL
     1. <http://127.0.0.1:8000/>

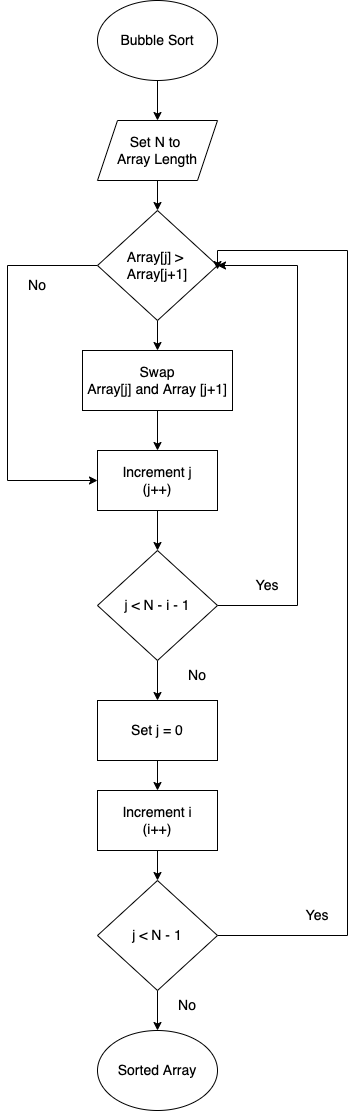
1. **Usage**

* **Add a new user**
  + Click on “Add new user” button to add new user.
  + Enter user information like “First Name”, “Last Name”, “Phone Number”, “Email” and click on submit.
* **Removing a user**
  + Click on the “Remove” button under the “Action” column to remove a user.
* **Sort table**
  + Click on “First Name”, “Last Name”, “Phone Number”, or “Email” to sort the data by that field in ascending order.
  + Click on the same button again to sort in descending order.
* **Search for user**
  + In the search input field, enter a “First Name” or “Last Name” or “Phone Number” or “Email” to search.
  + Click on the “Search” button to being searching.

1. **Algorithms and data structures**

The application uses a simple dynamic array (better known as a List in Python) to store the user data. A list is a very fast data structure because its elements are contiguous (stored next to each other) in memory. The CPU loves when data are close to each other due to how fetching data from RAM, the cache, and the cache line work.

**Bubble Sort Algorithm**



**Search Algorithm**

**Diagram

Description automatically generated**

**Sudo Code Bubble Sort**

Set N to Array Length

[STEP 2] if Array[j] > Array[j + 1]

if not continue to step 4

Swap Array[j] and Array[j+1]

[STEP 4] Increment j (j++)

If j < N - i – 1

if TRUE - continue to step 2

Set j = 0

Increment I by 1

If i < N – 1

if TRUE continue to step 2

1. **Test plan**

Test driven development is a fantastic way to create, maintain, and update a piece of software. It allows the development team to add new features while making sure that the old features are still working. This approach might increase development time, however in the long run it will save on time and prevent a lot of headaches.

The approach is very simple. The development team writes a set of tests with a set of inputs and an expected output. After running the test, if the output equals the expected output, then the test passes. If not, the test fails.

**Test 1:** Inserting an element into a list

**Description:** Insert the below input element into a list

Input:

User: { id=0, first\_name=”Mustafa”, last\_name=”Sibai”, phone\_number: “0557716033”, email=“contact@m-sibai.com” }

Output:

[ User: { id=0, first\_name=”Mustafa”, last\_name=”Sibai”, phone\_number: “0557716033”, email=“contact@m-sibai.com” } ]

**Test 2:** Removing an element from a list

**Description:** Remove element at index 1 from the list

Input:

[

User: { id=0, first\_name=”Anna”, last\_name=”Tookey”, phone\_number: “0557788099”, email=“contact@a-tookey.com” },

User: { id=1, first\_name=”Mustafa”, last\_name=”Sibai”, phone\_number: “0557716033”, email=“contact@m-sibai.com” },

User: { id=2, first\_name=”David”, last\_name=”John”, phone\_number: “05577998899”, email=“contact@d-john.com” }

]

Output:

[

User: { id=0, first\_name=”Anna”, last\_name=”Tookey”, phone\_number: “0557788099”, email=“contact@a-tookey.com” },

User: { id=2, first\_name=”David”, last\_name=”John”, phone\_number: “05577998899”, email=“contact@d-john.com” }

]

**Test 3:** Sort list in ascending order

**Description:** Sort a list of User in ascending order by first\_name

Input:

[

User: { id=0, first\_name=”Anna”, last\_name=”Tookey”, phone\_number: “0557788099”, email=“contact@a-tookey.com” },

User: { id=1, first\_name=”Mustafa”, last\_name=”Sibai”, phone\_number: “0557716033”, email=“contact@m-sibai.com” },

User: { id=2, first\_name=”David”, last\_name=”John”, phone\_number: “05577998899”, email=“contact@d-john.com” }

]

Output:

[

User: { id=0, first\_name=”Anna”, last\_name=”Tookey”, phone\_number: “0557788099”, email=“contact@a-tookey.com” },

User: { id=2, first\_name=”David”, last\_name=”John”, phone\_number: “05577998899”, email=“contact@d-john.com” },

User: { id=1, first\_name=”Mustafa”, last\_name=”Sibai”, phone\_number: “0557716033”, email=“contact@m-sibai.com” }

]

**Test 4:** Search for a User in a list

**Description:** Search for a User in a list by first\_name

Data:

[

User: { id=0, first\_name=”Anna”, last\_name=”Tookey”, phone\_number: “0557788099”, email=“contact@a-tookey.com” },

User: { id=2, first\_name=”David”, last\_name=”John”, phone\_number: “05577998899”, email=“contact@d-john.com” },

User: { id=1, first\_name=”Mustafa”, last\_name=”Sibai”, phone\_number: “0557716033”, email=“contact@m-sibai.com” }

]

Input:

{ search\_term=”David”}

Output:

[

User: { id=2, first\_name=”David”, last\_name=”John”, phone\_number: “05577998899”, email=“contact@d-john.com” }

]