



BridgeLabz

Employability Delivered

AddressBook System

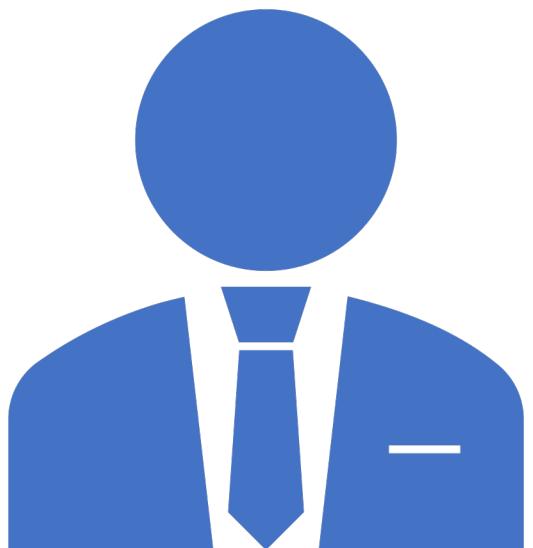
Outcome

1. Ability to work with File Handles
2. Ability to work with Database
3. Ability to save to Cloud Server
4. Ability to use Multi-Threading for IO or Network calls



START

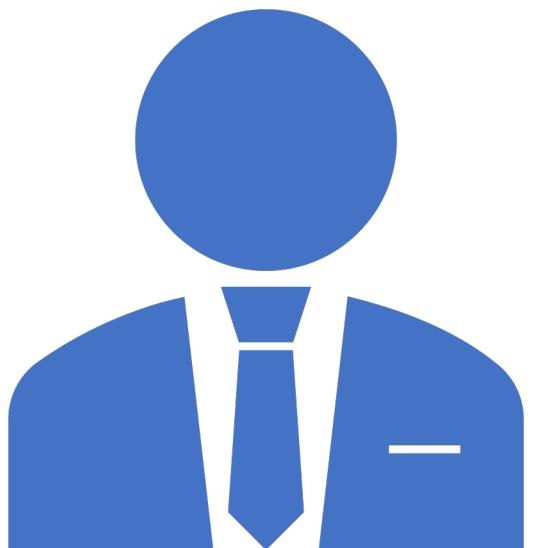
Start with Displaying
Welcome to Address Book
Program in
AddressBookMain class on
Master Branch



UC 1

Ability to create a Contacts in Address Book with first and last names, address, city, state, zip, phone number and email...

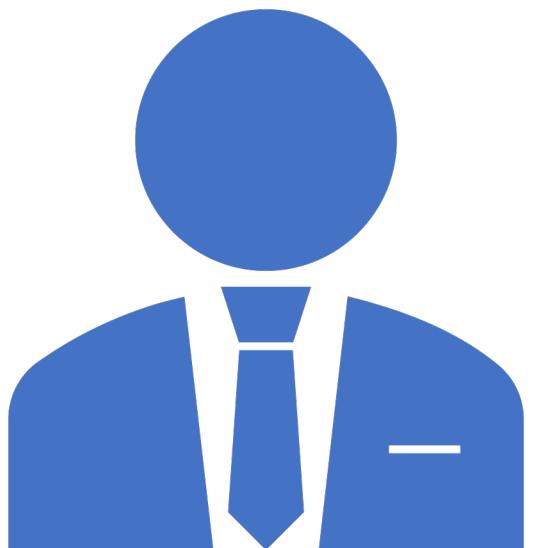
- Program is written using IDE like Visual Studio
- Every UC is in a separate Git Branch and then merged with main
- Naming Convention, Indentation, etc Code Hygiene will be checked during Review
- Git Check In Comments and Version History will be monitored



UC 2

Ability to add a new Contact to Address Book

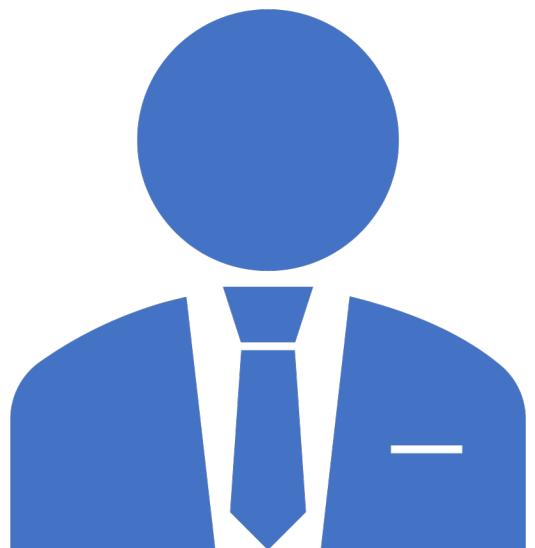
- Use Console to add person details from AddressBookMain class
- Use Object Oriented Concepts to manage relationship between AddressBook and Contact Person



UC 3

Ability to edit
existing contact
person using their
name

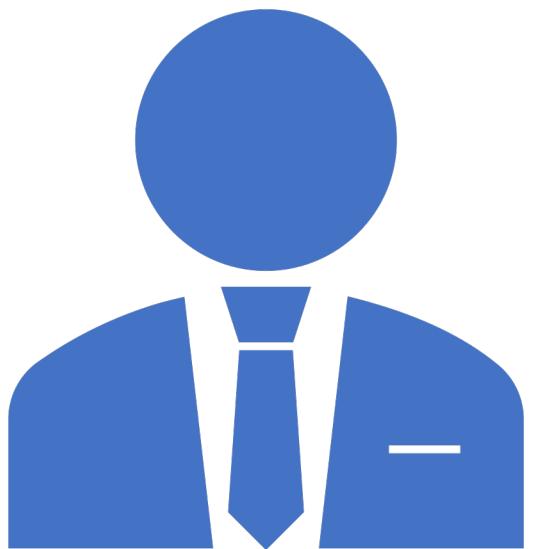
- Use Console to edit person details



UC 4

**Ability to delete a
person using
person's name**

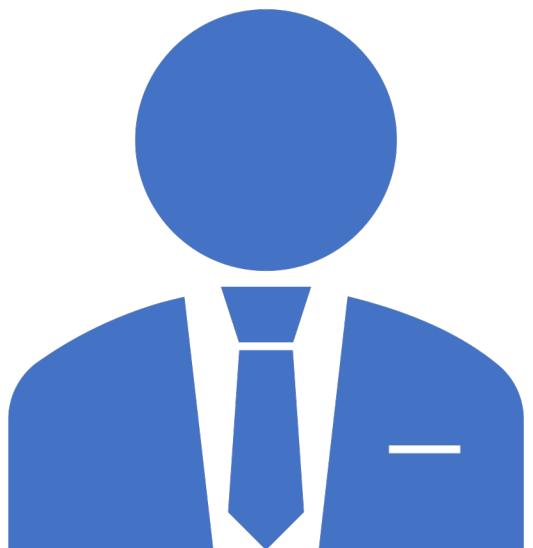
- Use Console to delete a person



UC 5

Ability to add multiple person to Address Book

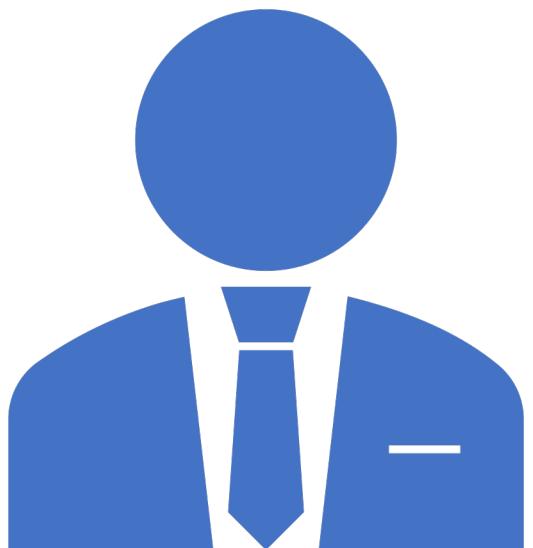
- Use Console to add person details one at a time
- Use Collection Class to maintain multiple contact persons in Address Book



UC 6

Refactor to add multiple Address Book to the System. Each Address Book has a unique Name

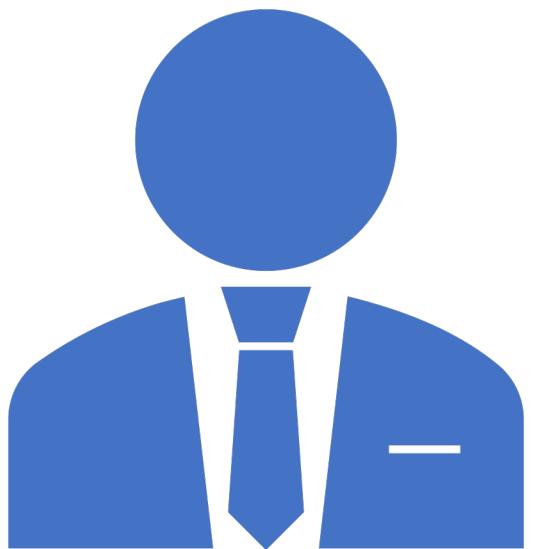
- Use Console to add new Address Book
- Maintain Dictionary of Address Book Name to Address Book



UC 7

Ability to ensure there is no Duplicate Entry of the same Person in a particular Address Book

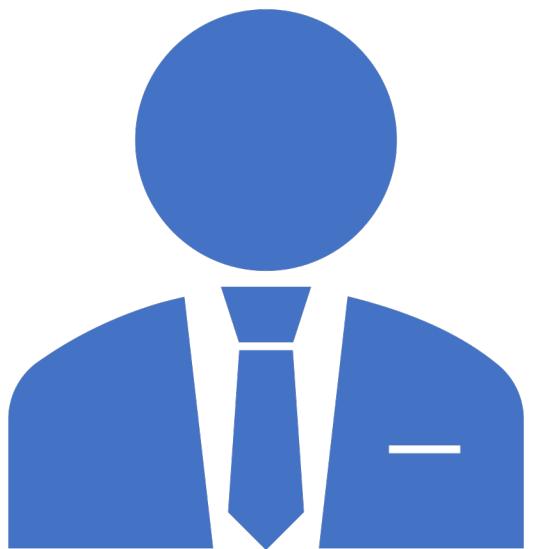
- Duplicate Check is done on Person Name while adding person to Address Book.
- Use Collection Methods to Search Person by Name for Duplicate Entry
- Override **equals** method to check for Duplicate



UC 8

**Ability to search Person
in a City or State across
the multiple Address
Book**

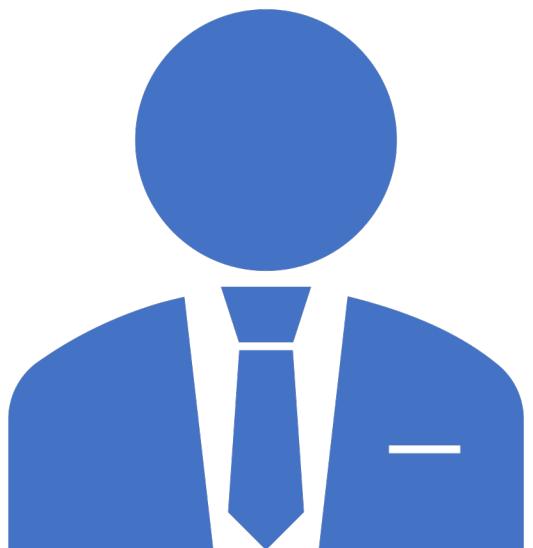
- Search Result can show multiple person in the city or state



UC 9

Ability to view Persons by City or State

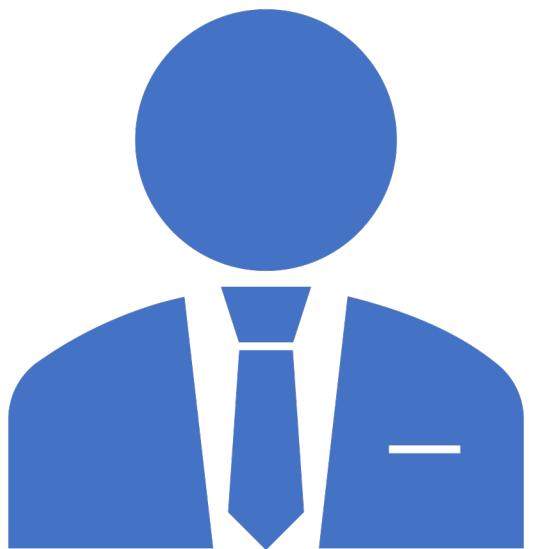
- Maintain Dictionary of City and Person as well as State and Person
- Use Collection Library to maintain Dictionary



UC 10

Ability to get number
of contact persons
i.e. count by City or
State

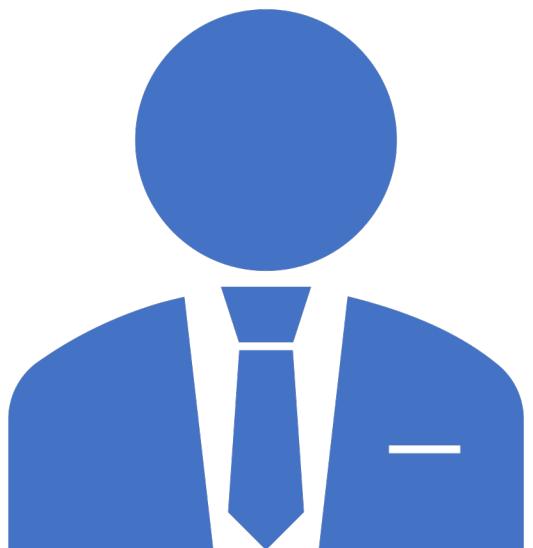
- Search Result will show count by city
and by state



UC 11

Ability to sort the entries in the address book alphabetically by Person's name

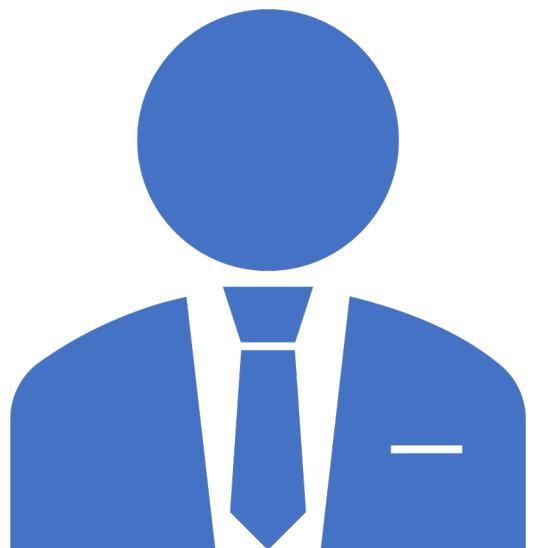
- Use Console to sort person details by name
- Use Collection Library for Sorting
- Override **toString** method to finally Print Person Entry in Concole



UC 12

**Ability to sort the entries
in the address book by
City, State, or Zip**

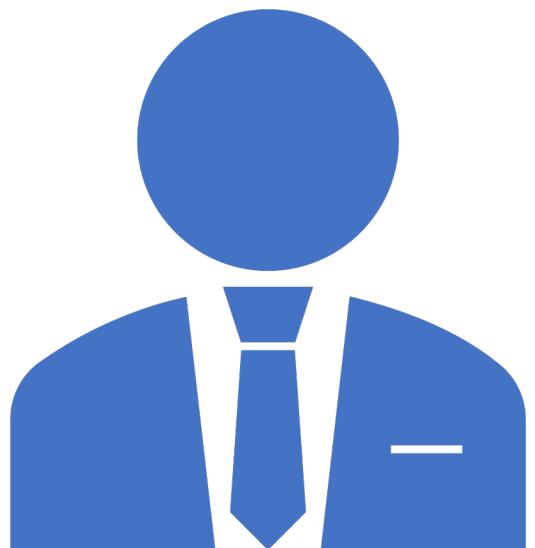
- Write functions to sort person by City, State or Zip
- Use Collection Library for Sorting



UC 13

**Ability to Read or Write
the Address Book with
Persons Contact into a
File using File IO**

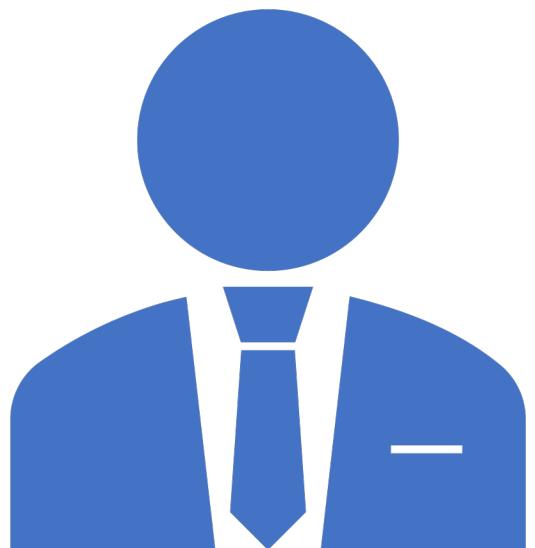
- Using C# File IO



UC 14

**Ability to Read/Write
the Address Book
with Persons Contact
as CSV File**

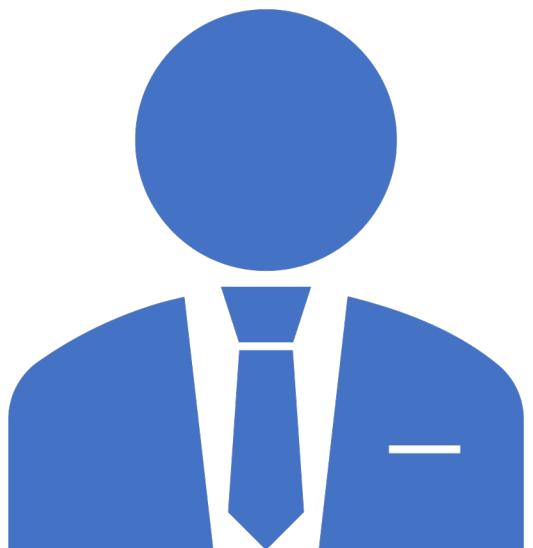
- Use OpenCSV Library



UC 15

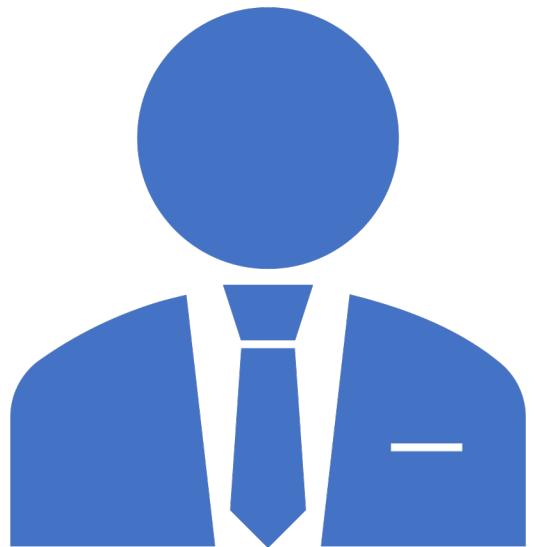
**Ability to Read or
Write the Address
Book with Persons
Contact as JSON File**

- Use GSON Library



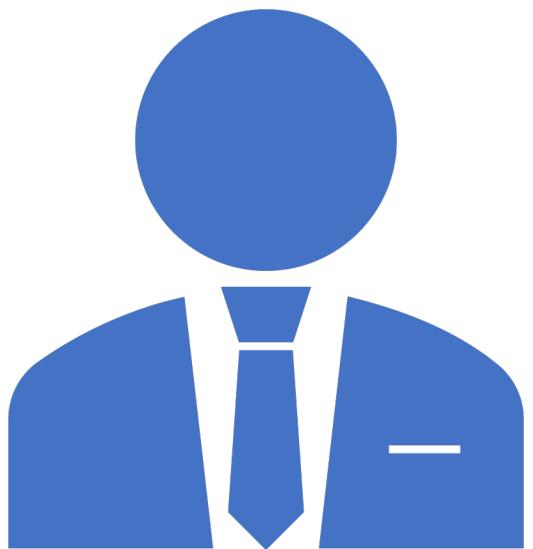
Ability to Read or Write
the Address Book with
Persons Contact to a
JSONServer

- Use RESTAssured.NET



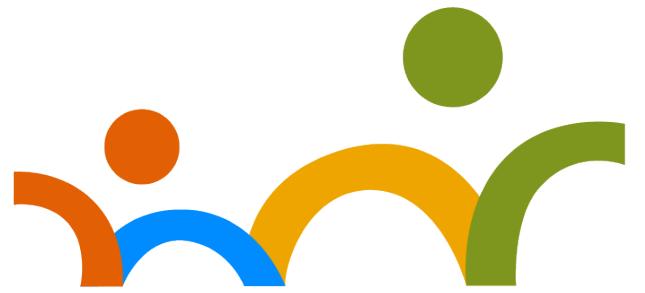
UC 17

Ensure IO Operation is not blocking the main thread while doing CURD operation on any of the three data source i.e. CSV File, JSON File, or JSONServer.



UC 18

Save the AddressBook to Database and Ensure Open Close Principle is not violated when new data source is added to already three data source i.e. CSV File, JSON File, or JSONServer.



BridgeLabz

Employability Delivered

Thank
You