**Phone Book Programming Design**

Student’s Name

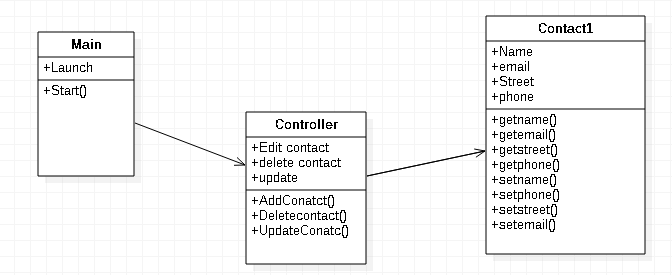
Professor’s Name

Course

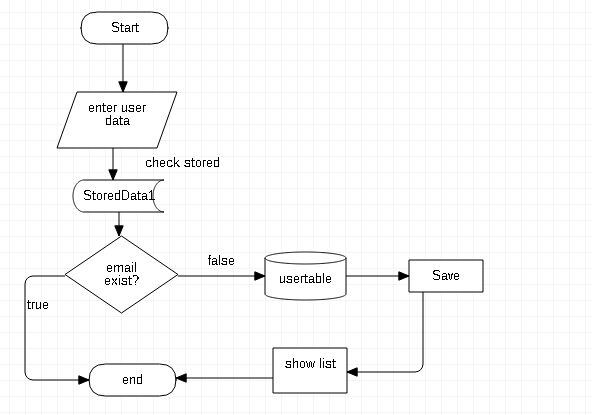
For this assignment I have selected phone book application . A phone book is a directory or a database containing a collection of names, phone numbers, and sometimes additional information such as addresses, email addresses, and other contact details. It serves as a reference or a tool for managing and organizing contact information for individuals, businesses, or organizations. The primary purpose of a phone book is to provide a centralized and convenient way to look up and retrieve contact information for individuals or businesses. It enables users to quickly find and communicate with people or organizations by searching for their names or phone numbers. With advancements in technology, traditional printed phone books have become less common, and online directories or digital contact management applications have gained popularity. These digital phone books provide more advanced search capabilities, the ability to sync and backup contacts, and other features to enhance contact management and communication efficiency.

the designed application focus on emulating these features of traditional phone book to digital phone where the persons can manage their contacts with easy. the application can serve as module for other application such as organizational system management with few modification. It stores the data in table giving allowing users to view data in category. Also makes it easy to obtain data for transferring from one person to another.

The application has three major function: deleting, updating and inserting. This are critical performance required in every data management system. Therefore users can add new details to the phone and edit them. since are persons information which prone to changes it is important to allow users to edit such information. this reduces repitition if for example person changes street, or email does not need to register again but just edit the information. this way it saves space and time. In other cases users wants to delete, to create, or if the information stored is no longer useful to them. The freedom of users for this application is highly depended on those three functionality.



The above diagrams shows the simple model design of the application, showing the classes, methods and attribute. The below diagram shows algorithm for the saving data.



**ArrayList Data Structures**

ArrayLists provide a versatile data structure for handling data from database tables in a phone book application. They offer dynamic sizing, random access, ordered collection, flexibility with data types, and easy iteration, making them an excellent choice for managing and manipulating the retrieved data efficiently. ArrayList data structures are suitable for obtaining data from database tables and the user table in a phone book application due to several reasons:

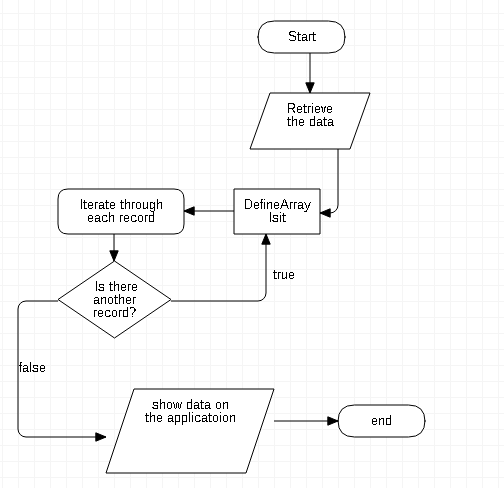
1. Dynamic Size: ArrayLists in most programming languages can dynamically grow and shrink as needed. This flexibility is beneficial when working with data from database tables, as the number of records can vary. You can add or remove elements from the ArrayList based on the retrieved data without worrying about the initial size allocation.

2. Random Access: ArrayLists provide efficient random access to elements. When retrieving data from database tables, you might need to access specific records based on indexes or perform search operations. ArrayLists allow direct access to elements based on their index, making it easy to retrieve and manipulate data.

3. Ordered Collection: When obtaining data from database tables, the order of records might be important, especially for phone book applications. ArrayLists maintain the order of elements as they are added, allowing you to preserve the desired sequence. This ensures that the retrieved data can be displayed or manipulated in the intended order, such as alphabetically by contact names.

4. Flexibility with Data Types: ArrayLists can store objects of any type, making them suitable for storing data retrieved from different columns of a database table. For example, you can have an ArrayList of custom objects representing contacts, with each object containing properties like name, phone number, and email address. This flexibility allows you to map the database table structure to the ArrayList structure seamlessly.

5. Easy Iteration: ArrayLists provide straightforward iteration mechanisms. When retrieving data from a database table, you often need to process each record individually. ArrayLists allow you to iterate through the elements using simple loops or iterator methods, making it convenient to perform operations on each data entry.

\

Test Plan for Phonebook Application Development:

1. Checkpoint 1: User Registration

- Test Scenario 1: User registration with valid credentials

- Test Scenario 2: User registration with invalid or duplicate credentials

- Test Scenario 3: User registration with missing or incomplete information

2. Checkpoint 2: Contact Creation

- Test Scenario 1: Creating a new contact with valid information

- Test Scenario 2: Creating a contact with missing or incomplete information

- Test Scenario 3: Creating a contact with duplicate information

3. Checkpoint 3: Contact Searching

- Test Scenario 1: Searching for a contact by name

- Test Scenario 2: Searching for a contact by phone number

- Test Scenario 3: Searching for a contact with non-existent or partial information

4. Checkpoint 4: Contact Editing

- Test Scenario 1: Editing an existing contact's information

- Test Scenario 2: Editing a contact with missing or incomplete information

- Test Scenario 3: Editing a non-existent contact

5. Checkpoint 5: Contact Deletion

- Test Scenario 1: Deleting an existing contact

- Test Scenario 2: Deleting a non-existent contact

6. Checkpoint 6: Contact Sorting

- Test Scenario 1: Sorting contacts by name in ascending order

- Test Scenario 2: Sorting contacts by name in descending order

- Test Scenario 3: Sorting contacts by phone number in ascending order

7. Checkpoint 7: Data Backup and Restore

- Test Scenario 1: Backing up the phonebook data

- Test Scenario 2: Restoring the phonebook data from a backup

Range of Data for Testing:

For each checkpoint, it is recommended to test with the following range of data:

- Small data set: 10-20 contacts

- Medium data set: 100-200 contacts

- Large data set: 1000+ contacts

Expected Test Results:

1. User Registration:

- Test Scenario 1: Successful registration

- Test Scenario 2: Error message indicating invalid or duplicate credentials

- Test Scenario 3: Error message indicating missing or incomplete information

2. Contact Creation:

- Test Scenario 1: Successful creation of a new contact

- Test Scenario 2: Error message indicating missing or incomplete information

- Test Scenario 3: Error message indicating duplicate information

3. Contact Searching:

- Test Scenario 1: Successful search and retrieval of a contact by name or phone number

- Test Scenario 2: No results found for the search query

- Test Scenario 3: Partial or fuzzy search results returned

4. Contact Editing:

- Test Scenario 1: Successful editing of an existing contact's information

- Test Scenario 2: Error message indicating missing or incomplete information

- Test Scenario 3: Error message indicating the contact does not exist

5. Contact Deletion:

- Test Scenario 1: Successful deletion of an existing contact

- Test Scenario 2: Error message indicating the contact does not exist

6. Contact Sorting:

- Test Scenario 1: Contacts sorted by name in ascending order

- Test Scenario 2: Contacts sorted by name in descending order

- Test Scenario 3: Contacts sorted by phone number in ascending order

7. Data Backup and Restore:

- Test Scenario 1: Successful backup creation and restoration of data

- Test Scenario 2: Error message indicating a failure in data backup or restoration

