Project of DSA Calci

Presented by :
Sandip Gaywali
Siddhartha Basnet
Saugat Wasti
Niraj Bhattarai

Context

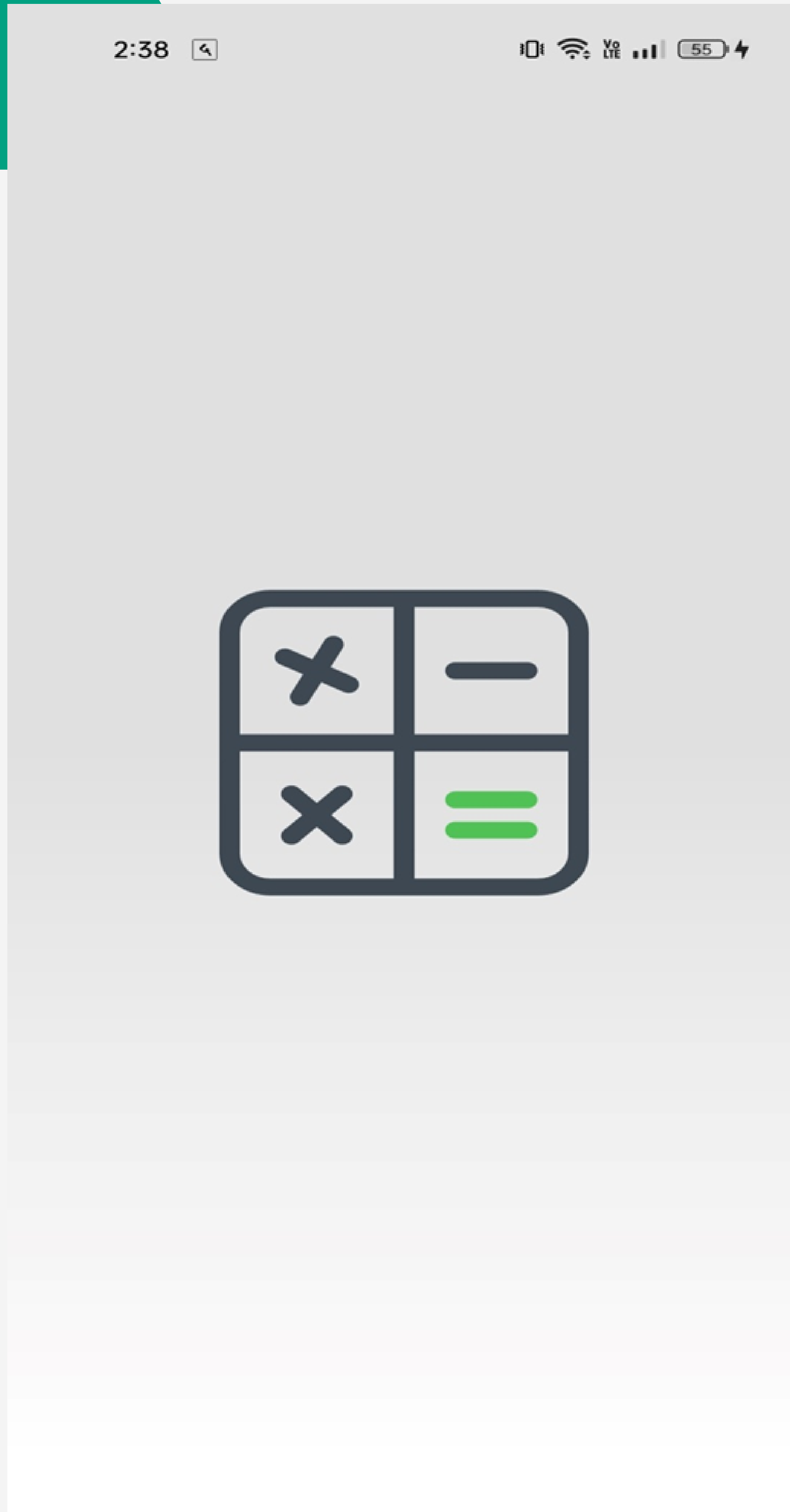
Application

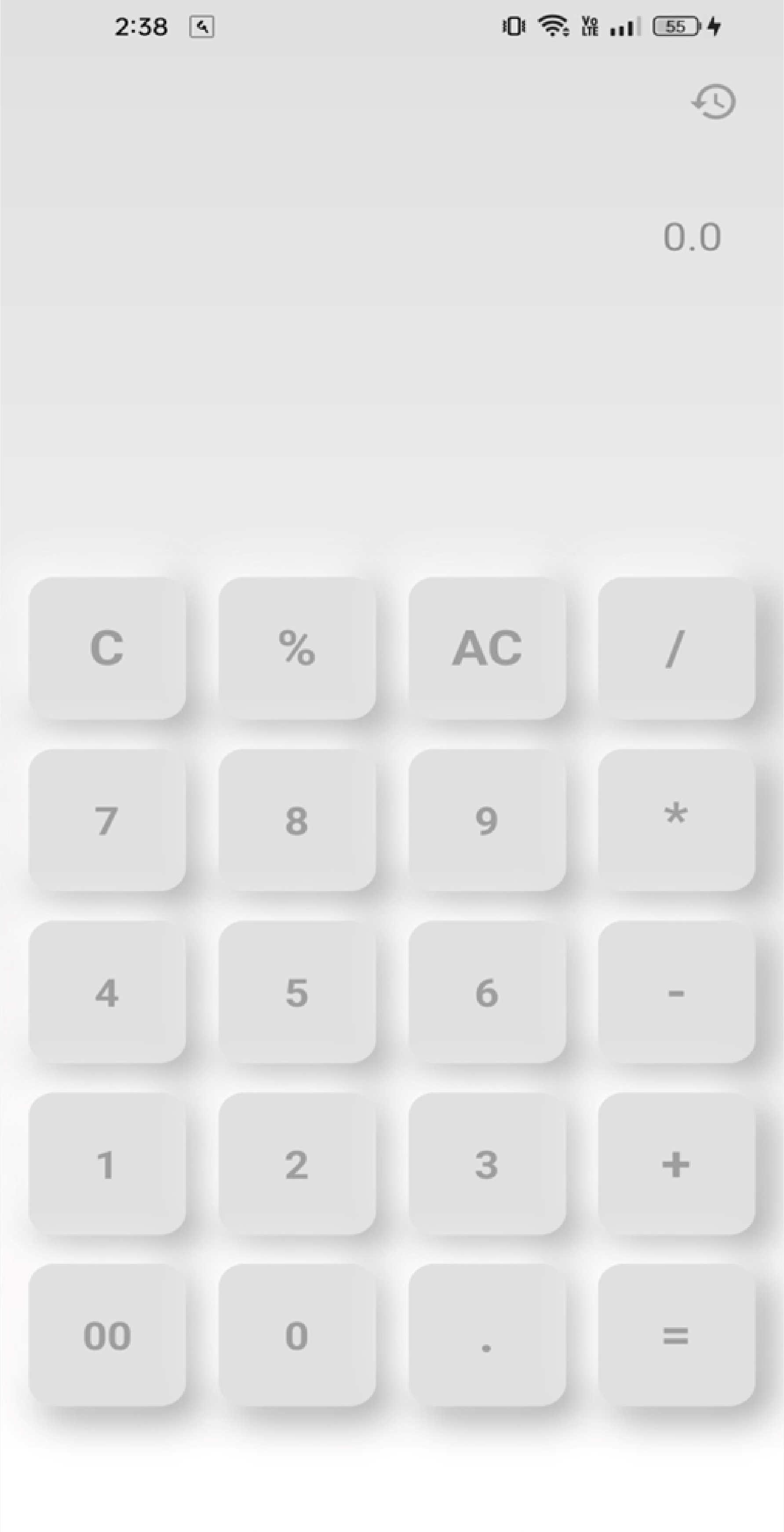
Stack

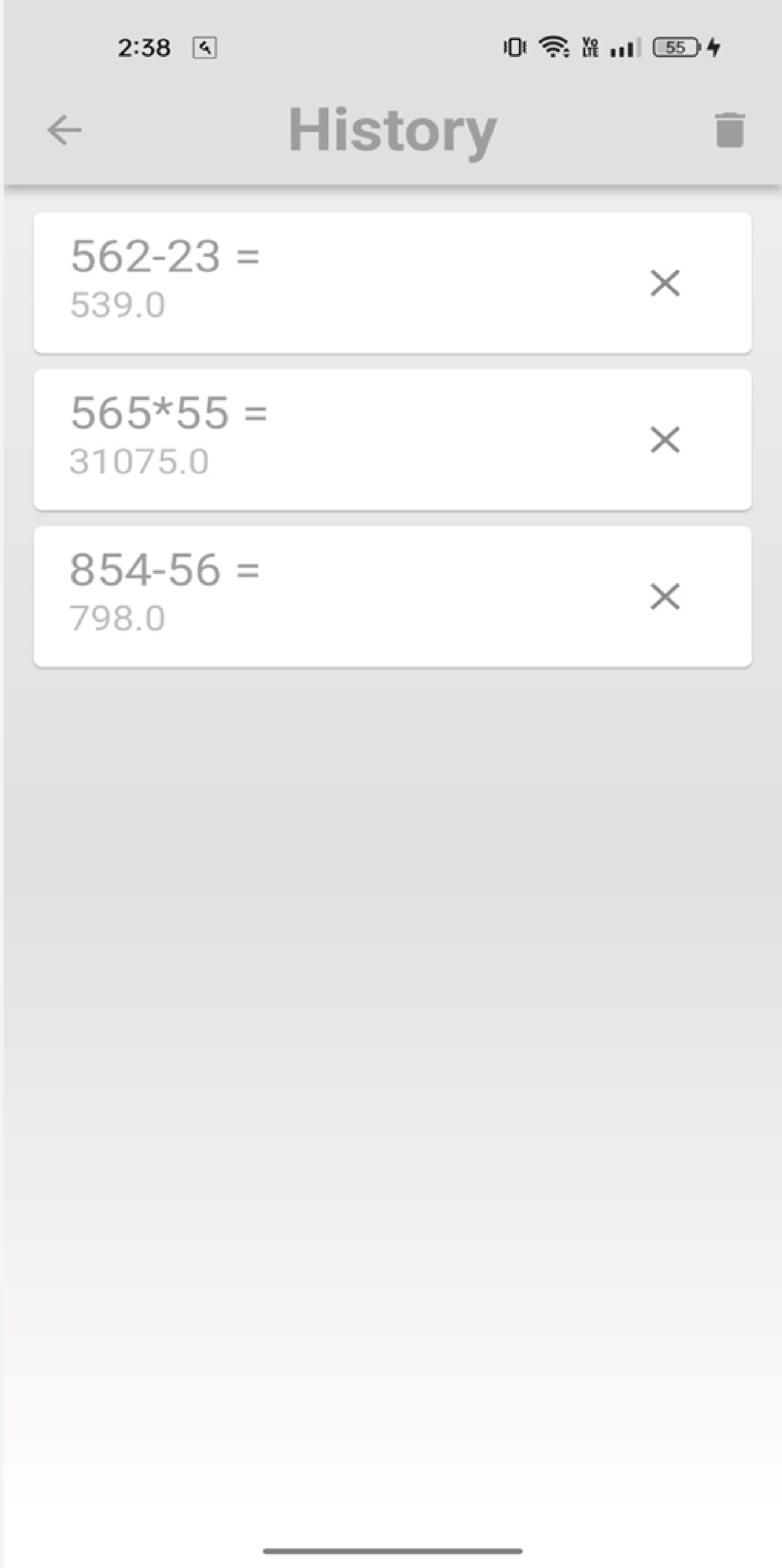
Array

List

Application Screen







History

562-23 =
539.0

565*55 =
31075.0

854-56 =
798.0

Stack

A stack is an abstract data type that represents a collection of elements, with two main operations:

- Push: which adds an element to the top of the stack.
- Pop: which removes the top element from the stack.

In addition to these, stacks typically include other operations such as:

- Peek: which returns the top element without removing it from the stack.
- Size: which returns the number of elements in the stack.
- IsEmpty: which checks if the stack is empty.

A stack follows the Last-In-First-Out (LIFO) principle, meaning that the most recently added element is the first one to be removed.

Stack

Stacks can be implemented using an array or a linked list. The implementation using an array is usually more efficient in terms of time and space complexity, but it is limited by a fixed size. On the other hand, the implementation using a linked list is more flexible in terms of size, but it requires additional memory to store the pointers to the next node.

Stacks are used in a variety of applications, such as:
Evaluating expressions, such as in a calculator.
Implementing recursive algorithms.
Parsing and syntax analysis.
Maintaining call stacks in programming languages.

Array

- An array is a collection of elements of the same data type that are stored sequentially in memory.
- The size of an array is fixed and cannot be changed once it is created.
- Elements in an array are accessed using an index, which represents their position in the array.
- The index of the first element in an array is typically 0.
- Arrays can be used for fast access to elements using their index and are often used to implement data structures such as stacks, queues, and hash tables.

Common operations performed on arrays include searching, sorting, and iterating.

List

A list is a collection of elements of the same or different data types that are stored in a linked sequence of nodes. The size of a list can change dynamically as elements are added or removed. Elements in a list are accessed by traversing the linked sequence of nodes. A list can be implemented using a singly-linked list, doubly-linked list, or circular linked list.

Lists can be used to implement various data structures such as stacks, queues, and trees. Common operations performed on lists include insertion, deletion, searching, and iterating

Thank you!