```python
class Node():
    def __init__(self, value):
        self.value = value
        self.next = None

class LinkedList():
    def __init__(self):
        self.head = None

    def append(self, value):
        if self.head is None:
            self.head = Node(value)
            return

        current_node = self.head
        new_node = Node(value)
        while (current_node.next):
            current_node = current_node.next
        current_node.next = new_node

    def to_list(self) -> list:
        l = []
        current_node = self.head
        while current_node:
            l.append(str(current_node.value))
            current_node = current_node.next
        return l

    def search(self, value):
        current_node = self.head
        while current_node:
            if current_node.value == value:
                return True
            current_node = current_node.next
        return False

    def delete(self, value) -> bool:
        temp_node = self.head

        # if head node is to be deleted
        if temp_node is not None:
            if temp_node.value == value:
                self.head = temp_node.next
                temp_node = None
                return True

        # search for value to delete
        # keep track of previous node
        while temp_node is not None:
            if temp_node.value == value:
                break
            previous_node = temp_node
            temp_node = temp_node.next

        # return false if node is not found
        if temp_node == None:
            return False

        # switch up pointers
        previous_node.next = temp_node.next

        # free the temp node
        temp_node = None
```

main.py

```python
from adt import LinkedList

def main():
    # Hardcoded users
    preexisting_users = ["Steven", "Faye", "Rangel", "Gea", "Oxy"]
    title = "Registration System!"
    registered = LinkedList()
    for user in preexisting_users:
        registered.append(user)

    print("=" * len(title))
    print(title)
    print("=" * len(title))
    while True:
        print("Current users registered: ", end ="")
        print(*registered.to_list(), sep=", ")
        action = input("Actions (register/delete/exit): ").strip().lower()
        if action == "register":
            registered_name = input("Enter your name: ").title()
            registered.append(registered_name)
        elif action == "delete":
            deleted_name = input("User to delete: ").strip().title()
            if registered.search(deleted_name) == True:
                registered.delete(deleted_name)
                print("User has been deleted!")
            else:
                print("User does not exist!")
        elif action == "exit":
            break



def get_string(prompt:str) -> str:
    try:
        return input(prompt)
    except ValueError:
        pass
if __name__ == "__main__":
    main()
```