Email Application Documentation

1. Overview

The **Email application** is a simple program that allows users to input their personal details, generate a secure random password, and store information related to their email account. It provides a structure to handle an email user's information, such as their name, email password, mailbox capacity, and alternate email.

2. Class Description

Email Class

The Email class represents an email user with basic attributes like first name, last name, email password, mailbox capacity, and alternate email. It provides the functionality to:

- Store user details.
- Generate a random password.
- Display user information.

3. Class Fields (Attributes)

1. **firstName (String)**:
    - Stores the user's first name.
2. **lastName (String)**:
    - Stores the user's last name.
3. **password (String)**:
    - Stores the user's email password.
4. **mailboxCapacity (int)**:
    - Represents the maximum size of the user's mailbox (in MB), initialized to 600.
5. **alternateEmail (String)**:
    - Stores an alternate email address for backup or recovery purposes.

4. Constructor

Email(String firstName, String lastName, String password, int mailboxCapacity, String alternateEmail)

- **Description**: Initializes an Email object with the given user details.
- **Parameters**:
    - firstName: The user's first name.
    - lastName: The user's last name.

- password: The user's email password.
- mailboxCapacity: The size of the user's mailbox in MB.
- alternateEmail: The user's alternate email address.

5. Methods

Getters and Setters

- **getFirstName()**: Returns the user's first name.
- **setFirstName(String firstName)**: Sets the user's first name.
- **getLastName()**: Returns the user's last name.
- **setLastName(String lastName)**: Sets the user's last name.
- **getPassword()**: Returns the user's email password.
- **setPassword(String password)**: Sets the user's email password.
- **getMailboxCapacity()**: Returns the user's mailbox capacity.
- **setMailboxCapacity(int mailboxCapacity)**: Sets the user's mailbox capacity.
- **getAlternateEmail()**: Returns the user's alternate email address.
- **setAlternateEmail(String alternateEmail)**: Sets the user's alternate email address.

generatePassword(int length)

- **Description**: Generates a random password of a given length using a predefined set of characters.
- **Parameters**:
  - length: The desired length of the password.
- **Returns**: A randomly generated password as a String.

info()

- **Description**: Returns a formatted string containing the user's details such as name, password, mailbox capacity, and alternate email.
- **Returns**: A formatted string containing user details.

toString()

- **Description**: Overrides the default toString() method to return the result of the info() method, providing a clean string representation of the user's details.
- **Returns**: A string containing the user's information.

6. Main Method

The main () method is the entry point of the program where the following actions occur:

1. **User Input**:
    - The program prompts the user to input their first name, last name, password, mailbox capacity, and alternate email.
2. **Password Generation**:
    - The program asks the user for a desired password length and generates a random password using the generate Password () method.
3. **Display Information**:
    - It displays the entered user details (name, password, mailbox capacity, and alternate email), as well as the generated password.

7. Example output

Sample Execution :

Enter your first name: sai

Enter your last name: charan

Enter your password: sai1234@

Enter your mailbox capacity: 600

Enter your alternate email: charan@example.com

User Details:

Enter your first name: sai

Enter your last name: charan

Enter your password: sai1234@

Enter your mailbox capacity: 600

Enter your alternate email: charan@example.com

8. Conclusion

This Email application provides a simple model for managing user details and generating secure passwords. It's designed for educational purposes and can be expanded with more advanced features like email sending, encryption, and additional security layers.