Suppose you need to implement "web-mail" for an organization. The term "web" in web-mail implies that we have web interface for composing, sending, reading mails. We also perform certain maintenance task related to mails. Also maintaining contact list etc. is part of the system.

Model-View-Controller is an architectural practice that often followed in designing enterprise applications. In this lab we attempt implementing Model for proposed application. Model classes primarily implement business functionality, and have no dependency on any of the presentation logic/objects.

Class diagram on next page suggests some classes, and given below is a use case scenario of proposed mailing system using said objects.

In this lab you do following-
  (1) Provide Implement of necessary classes. Note that implemented system may not really be fully usable but it should able to simulate the whole scenario.
  (2) Add two additional methods, **save** and **read** to WebMail class. Save methods saves mail boxes of all users in a WebMail objects on disk. Read methods reads in from earlier file(s) created by save method.

```
public class WebMailSimulator {

    public static void main(String[] str) {

        WebMail webm = new WebMail();
        User user = webm.login("abc", "xyz");
        //login operation throws an exception if login is not successful
        MailBox mbox = user.getMailBox();


        Message m = null;
        //let us say you recieve a new message, and
        //store in m and add to InBox of the user
        mbox.addToInbox(m);


        m = new Message("abc@xyz.com");
        m.addRecipeint("a@b.com");
        m.appendBody("");
        //...
        mbox.send(m);
        mbox.addToDraft(m);
    }
}
```

Here are the classes that are to be used in Mailing System-

WebMail

users : ArrayList

addUser(usr : User) : void
login(usr_name : String,pwd : String) : User
deleteUser(usr : User) : void

Folder

messages : ArrayList

add(msg : Message) : void
remove(msg_id : long) : void

User

name : String
passwd : String
mbox : MailBox

<<create>> User(user_name : String,pwd : String)
getMailBox() : MailBox

Message

to : ArrayList
toCC : ArrayList
toBCC : ArrayList
attachments : ArrayList
sender : String
body : String
mesg_id : long
NEXT_MESG_ID : long

<<create>> Message(sendr : String)
addRecipeint(tostr : String) : void
addCC(tostr : String) : void
addBCC(tostr : String) : void
addAttachments(obj : MimeType) : void
appendBody(str : String) : void
equals(message : Object) : boolean

MailBox

inbox : ArrayList
draft : ArrayList
sent : ArrayList
folders : ArrayList

send(m : Message) : void
addToInbox(m : Message) : void
addToDraft(m : Message) : void
clone() : MailBox