Email Server Lap oop

Student Name	Student ID
Karim Fathy Abd-Elaziz	20011116
Ahmed Mohamed Hassan	20010174
Mohamed Ahmed Ali Ryad	20011457
FARIS MOHAMED ANWAR	20011065

Uml:


Controller
~ service : Service
+ searchContact(user : String, contactName : String) : ResponseEntity<String>
+ renameContact(user : String, newName : String, oldName : String) : void
+ deleteContact(user : String, contactName : String) : void
+ editContactinfo(user : String, json_editedContact : String) : ResponseEntity<String>
+ addContact(user : String, json_Contact : String) : ResponseEntity<String>
+ getContacts(mail : String) : ResponseEntity<String>
+ searchString(mailAddress : String, folderName : String, str : String, attribute : String) : ResponseEntity<String>
+ searchImportance(mailAddress : String, folderName : String, imp : long) : ResponseEntity<String>
+ searchDate(mailAddress : String, folderName : String, date : long) : ResponseEntity<String>
+ moveMessages(list : String, use : String, to : String, from : String) : ResponseEntity<String>
+ delMessages(json : String, use : String, fol : String) : ResponseEntity<String>
+ sortByFrom(mailAddress : String, folderName : String) : ResponseEntity<String>
+ sortByTo(mailAddress : String, folderName : String) : ResponseEntity<String>
+ sortByImportance(mailAddress : String, folderName : String) : ResponseEntity<String>
+ sortByDate(mailAddress : String, folderName : String) : ResponseEntity<String>
+ addMesDraft(message : String) : void
+ addMessages(message : String) : void
+ deleteFolder(mailAddress : String, name : String) : void
+ renameFolder(mailAddress : String, oldName : String, newName : String) : String
+ addFolder(mailAddress : String, folderName : String) : String
+ signUp(mailAddress : String, pass : String) : String
+ logIn(mailAddress : String, pass : String) : ResponseEntity<String>
+ reload(mail : String, foldName : String) : ResponseEntity<String>
+ print() : ResponseEntity<String>
+ save() : void
+ load() : void
+ Controller(service : Service)

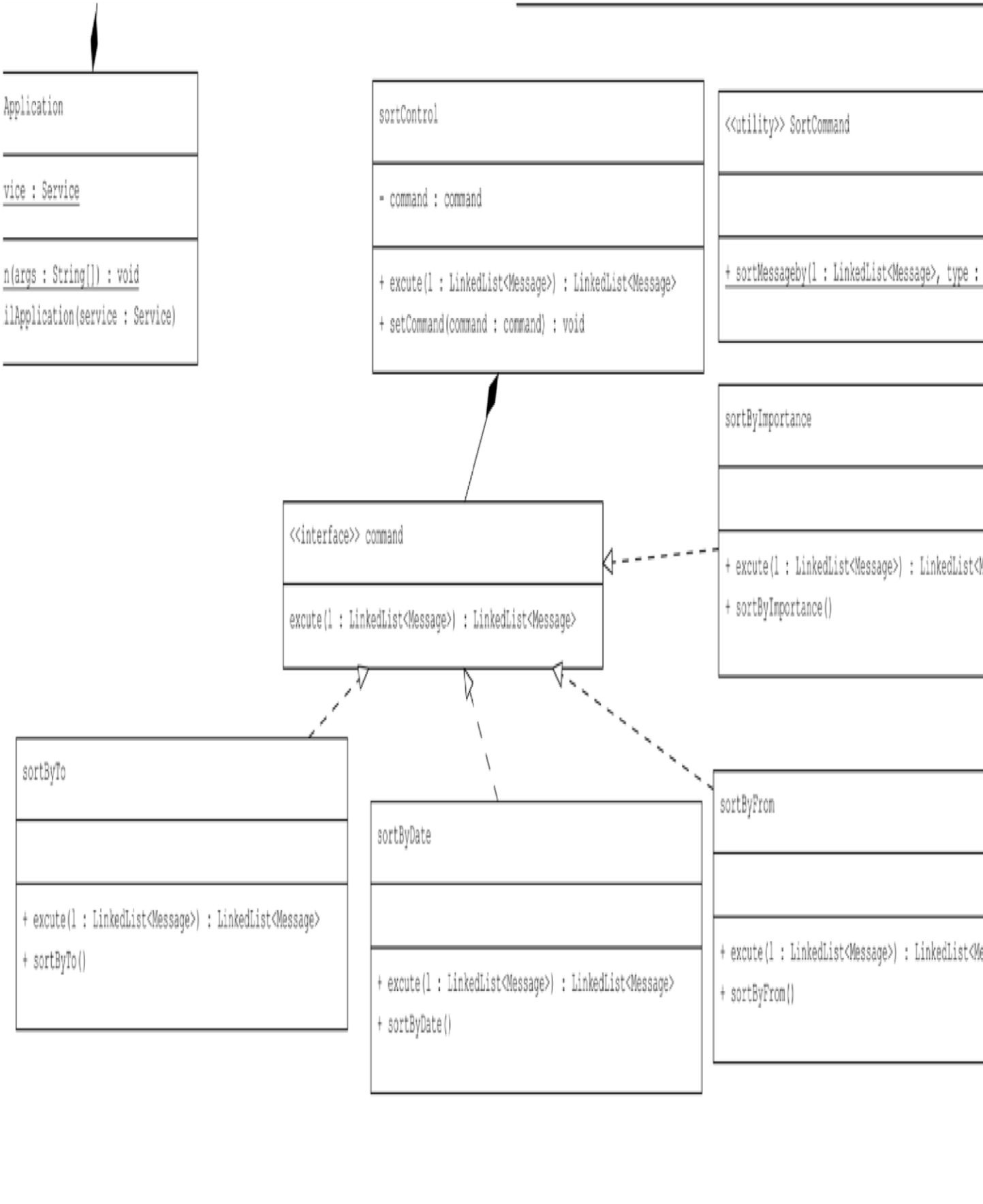
Service

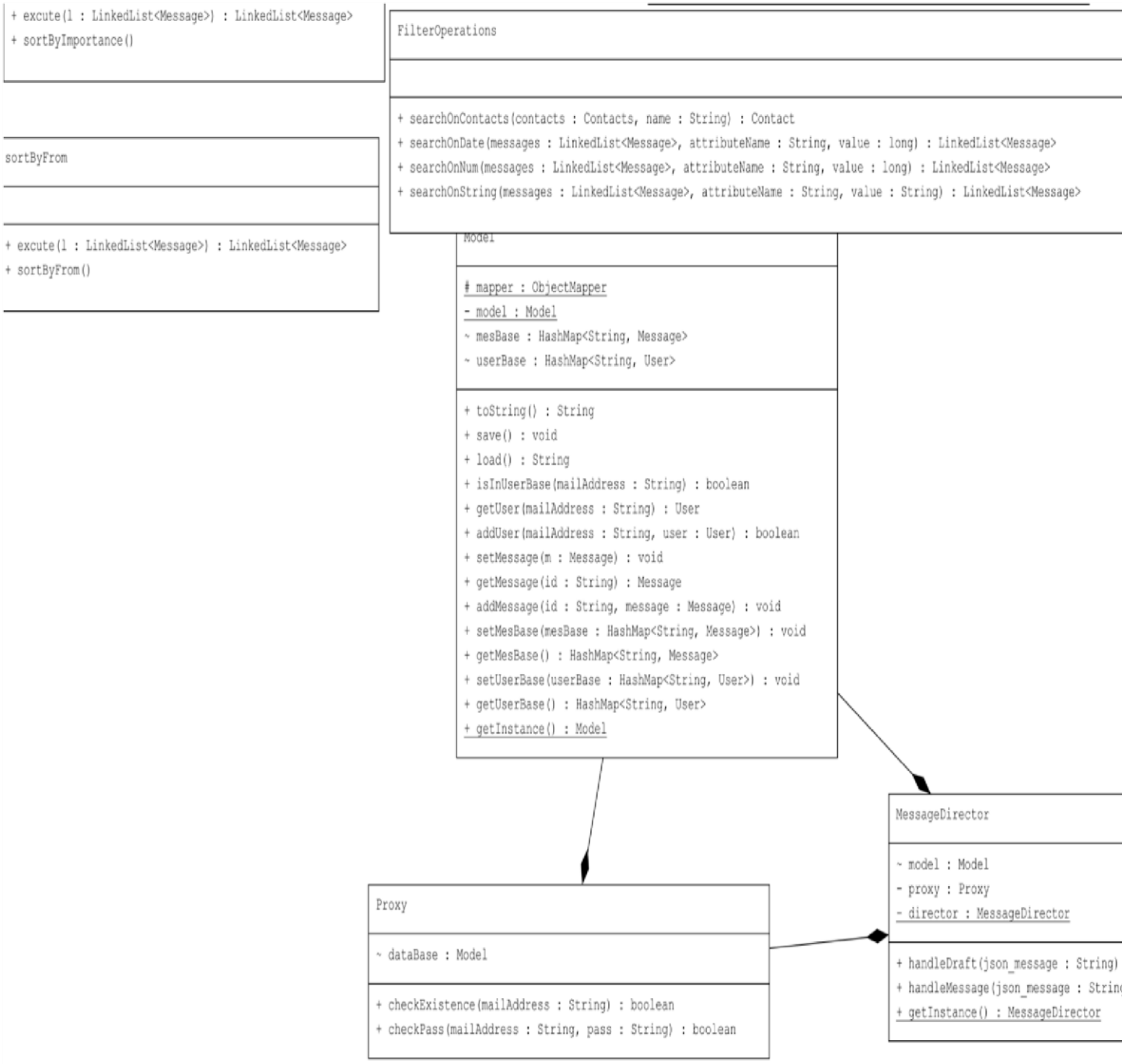
+ searchContact(user : String, contactName : String) : Contact
+ renameContact(user : String, newName : String, oldName : String) : void
+ editContactinfo(user : String, json_editedContact : String) : void
+ deleteContact(user : String, contactName : String) : void
+ addContact(user : String, json_Contact : String) : boolean
+ addMesDraft(json : String) : void
+ getPerson(name : String) : User
+ addMessages(json : String) : void
+ moveMessages(l : LinkedList<String>, use : String, to : String, from : String) : void
+ delMessages(l : LinkedList<String>, use : String, fol : String) : void
+ searchString(email : String, name : String, value : String, attributeName : String) : LinkedList<Message>
+ searchImportance(email : String, name : String, value : long) : LinkedList<Message>
+ searchDate(email : String, name : String, value : long) : LinkedList<Message>
+ sortByFrom(email : String, name : String) : LinkedList<Message>
+ sortByTo(email : String, name : String) : LinkedList<Message>
+ sortByImportance(email : String, name : String) : LinkedList<Message>
+ sortByDate(email : String, name : String) : LinkedList<Message>
+ deleteFolder(email : String, name : String) : void
+ renameFolder(email : String, oldF : String, newF : String) : boolean
+ addFolder(email : String, name : String) : boolean
+ printDataBase() : String
+ signUp(mailAddress : String, pass : String) : String
+ logIn(mailAddress : String, pass : String) : String
+ validateSignUP(mailAddress : String) : boolean
+ validateLogIn(mailAddress : String, pass : String) : boolean
+ saveDataBase() : void
+ loadDataBase() : void

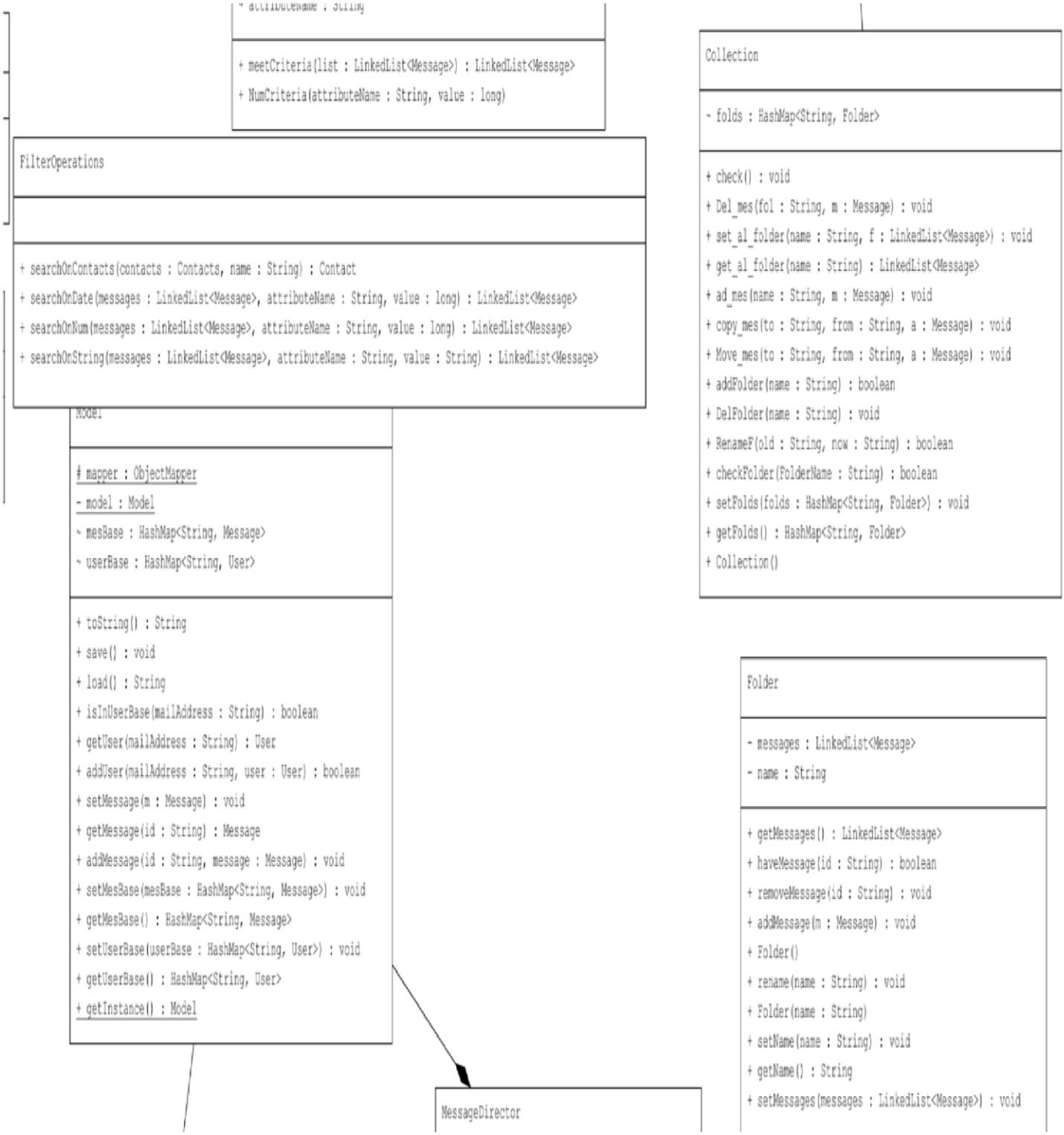
Message

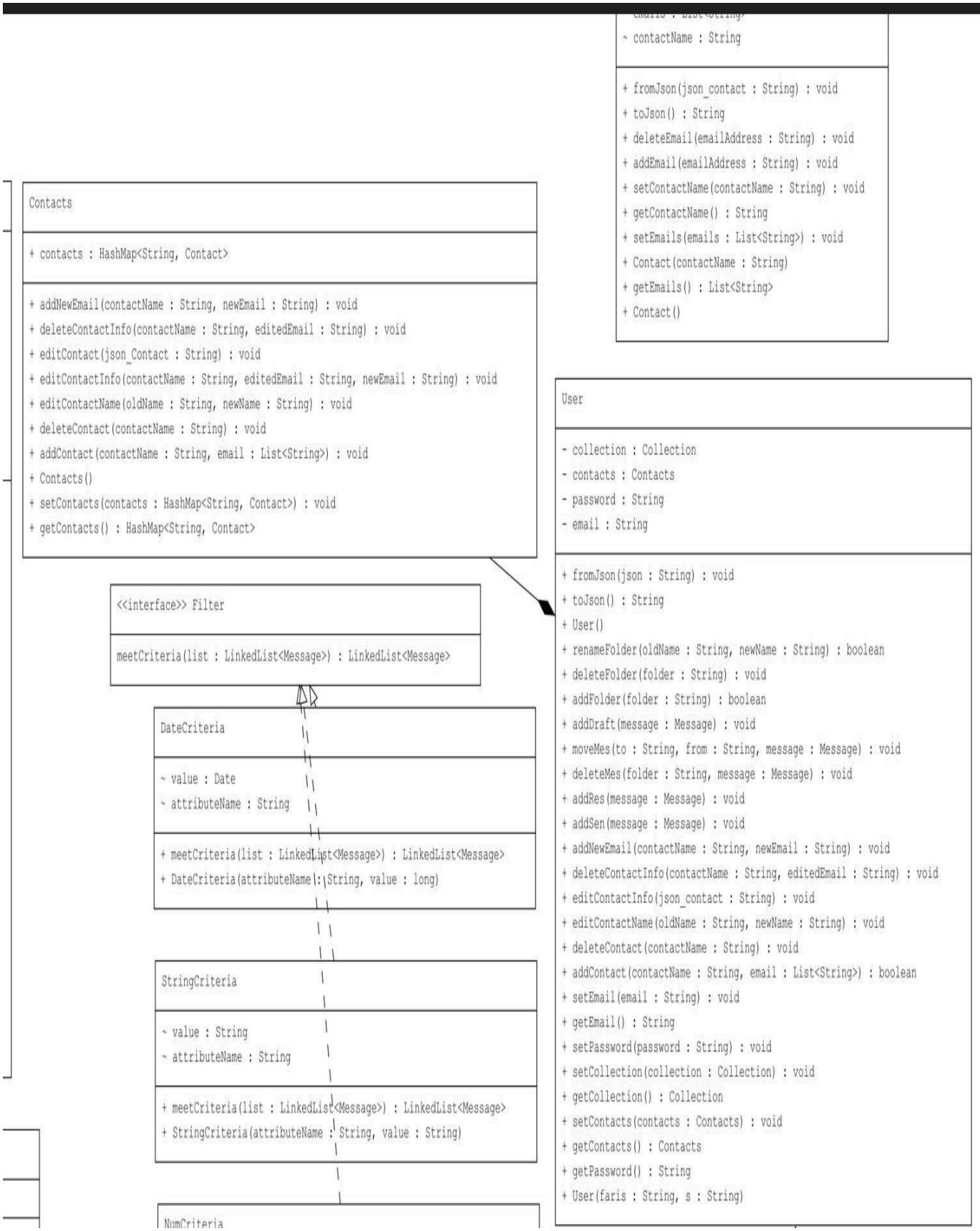
- attachment : String
- content : String
- subject : String
- delete : long
- date : long
- importance : long
- from : String
- to : String
- id : String

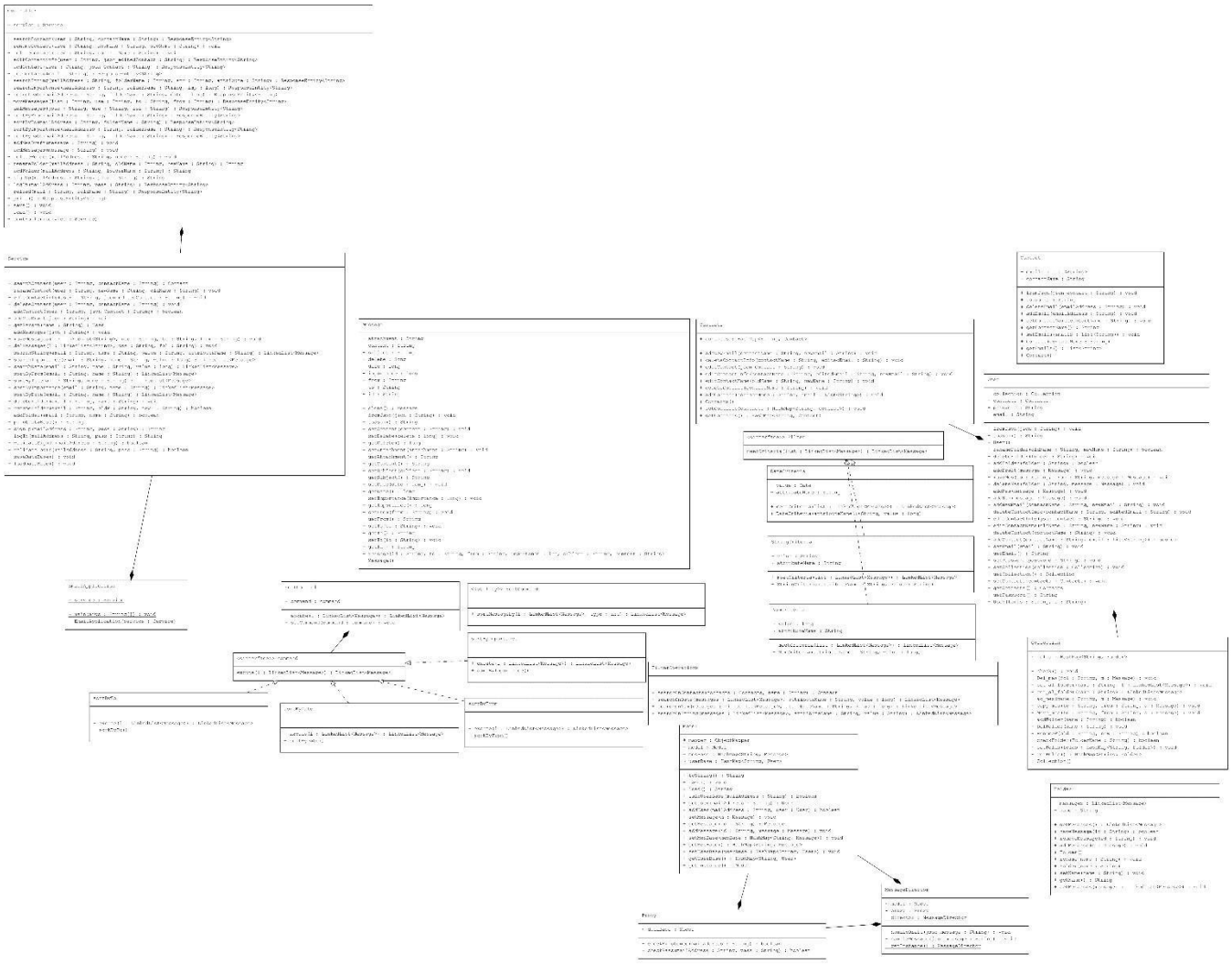
+ clone() : Message
+ fromJson(json : String) : void
+ toJson() : String
+ setContent(content : String) : void
+ setDelete(delete : long) : void
+ getDelete() : long
+ setAttachment(attachment : String) : void
+ getAttachment() : String
+ getContent() : String
+ setSubject(subject : String) : void
+ getSubject() : String
+ setDate(date : long) : void
+ getDate() : long
+ setImportance(importance : long) : void
+ getImportance() : long
+ setFrom(from : String) : void
+ getFrom() : String
+ setTo(to : String) : void
+ getTo() : String
+ setId(id : String) : void
+ getId() : String
+ Message(id : String, to : String, from : String, importance : int, subject : String, content : String)
+ Message()


Application
vice : Service
n(args : String[]) : void
ilApplication(service : Service)
sortControl
- command : command
+ excute(l : LinkedList<Message>) : LinkedList<Message>
+ setCommand(command : command) : void
<<utility>> SortCommand
+ sortMessageby(l : LinkedList<Message>, type :
sortByImportance
+ excute(l : LinkedList<Message>) : LinkedList<
+ sortByImportance()
<<interface>> command
excute(l : LinkedList<Message>) : LinkedList<Message>
sortByTo
+ excute(l : LinkedList<Message>) : LinkedList<Message>
+ sortByTo()
sortByDate
+ excute(l : LinkedList<Message>) : LinkedList<Message>
+ sortByDate()
sortByFrom
+ excute(l : LinkedList<Message>) : LinkedList<Me
+ sortByFrom()


+ excute(l : LinkedList<Message>) : LinkedList<Message>
+ sortByImportance()
FilterOperations
+ searchOnContacts(contacts : Contacts, name : String) : Contact
+ searchOnDate(messages : LinkedList<Message>, attributeName : String, value : long) : LinkedList<Message>
+ searchOnNum(messages : LinkedList<Message>, attributeName : String, value : long) : LinkedList<Message>
+ searchOnString(messages : LinkedList<Message>, attributeName : String, value : String) : LinkedList<Message>
sortByFrom
+ excute(l : LinkedList<Message>) : LinkedList<Message>
+ sortByFrom()
Model
mapper : ObjectMapper
- model : Model
~ mesBase : HashMap<String, Message>
~ userBase : HashMap<String, User>
+ toString() : String
+ save() : void
+ load() : String
+ isInUserBase(mailAddress : String) : boolean
+ getUser(mailAddress : String) : User
+ addUser(mailAddress : String, user : User) : boolean
+ setMessage(m : Message) : void
+ getMessage(id : String) : Message
+ addMessage(id : String, message : Message) : void
+ setMesBase(mesBase : HashMap<String, Message>) : void
+ getMesBase() : HashMap<String, Message>
+ setUserBase(userBase : HashMap<String, User>) : void
+ getUserBase() : HashMap<String, User>
+ getInstance() : Model
MessageDirector
~ model : Model
- proxy : Proxy
- director : MessageDirector
+ handleDraft(json_message : String)
+ handleMessage(json_message : Strin
+ getInstance() : MessageDirector
Proxy
~ dataBase : Model
+ checkExistence(mailAddress : String) : boolean
+ checkPass(mailAddress : String, pass : String) : boolean

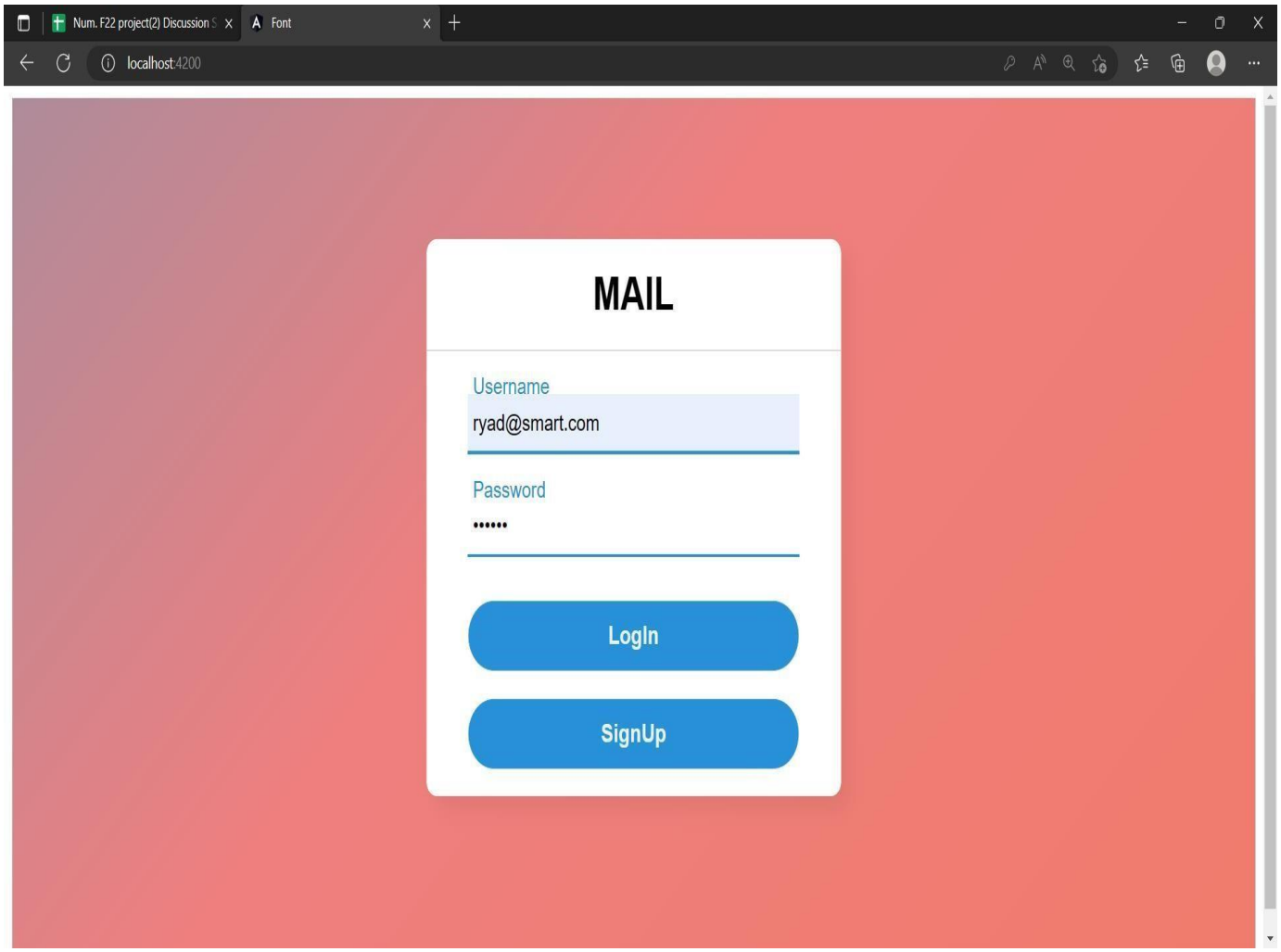

~ attributeName : String
+ meetCriteria(list : LinkedList<Message>) : LinkedList<Message>
+ NumCriteria(attributeName : String, value : long)
FilterOperations
+ searchOnContacts(contacts : Contacts, name : String) : Contact
+ searchOnDate(messages : LinkedList<Message>, attributeName : String, value : long) : LinkedList<Message>
+ searchOnNum(messages : LinkedList<Message>, attributeName : String, value : long) : LinkedList<Message>
+ searchOnString(messages : LinkedList<Message>, attributeName : String, value : String) : LinkedList<Message>
Collection
~ folds : HashMap<String, Folder>
+ check() : void
+ Del_mes(fol : String, m : Message) : void
+ set_al_folder(name : String, f : LinkedList<Message>) : void
+ get_al_folder(name : String) : LinkedList<Message>
+ ad_mes(name : String, m : Message) : void
+ copy_mes(to : String, from : String, a : Message) : void
+ Move_mes(to : String, from : String, a : Message) : void
+ addFolder(name : String) : boolean
+ DelFolder(name : String) : void
+ RenameF(old : String, now : String) : boolean
+ checkFolder(FolderName : String) : boolean
+ setFolds(folds : HashMap<String, Folder>) : void
+ getFolds() : HashMap<String, Folder>
+ Collection()
Model
mapper : ObjectMapper
- model : Model
~ mesBase : HashMap<String, Message>
~ userBase : HashMap<String, User>
+ toString() : String
+ save() : void
+ load() : String
+ isInUserBase(mailAddress : String) : boolean
+ getUser(mailAddress : String) : User
+ addUser(mailAddress : String, user : User) : boolean
+ setMessage(m : Message) : void
+ getMessage(id : String) : Message
+ addMessage(id : String, message : Message) : void
+ setMesBase(mesBase : HashMap<String, Message>) : void
+ getMesBase() : HashMap<String, Message>
+ setUserBase(userBase : HashMap<String, User>) : void
+ getUserBase() : HashMap<String, User>
+ getInstance() : Model
Folder
- messages : LinkedList<Message>
- name : String
+ getMessages() : LinkedList<Message>
+ haveMessage(id : String) : boolean
+ removeMessage(id : String) : void
+ addMessage(m : Message) : void
+ Folder()
+ rename(name : String) : void
+ Folder(name : String)
+ setName(name : String) : void
+ getName() : String
+ setMessages(messages : LinkedList<Message>) : void
MessageDirector


~ contactName : String
+ fromJson(json_contact : String) : void
+ toJson() : String
+ deleteEmail(emailAddress : String) : void
+ addEmail(emailAddress : String) : void
+ setContactName(contactName : String) : void
+ getContactName() : String
+ setEmails(emails : List<String>) : void
+ Contact(contactName : String)
+ getEmails() : List<String>
+ Contact()
Contacts
+ contacts : HashMap<String, Contact>
+ addNewEmail(contactName : String, newEmail : String) : void
+ deleteContactInfo(contactName : String, editedEmail : String) : void
+ editContact(json_Contact : String) : void
+ editContactInfo(contactName : String, editedEmail : String, newEmail : String) : void
+ editContactName(oldName : String, newName : String) : void
+ deleteContact(contactName : String) : void
+ addContact(contactName : String, email : List<String>) : void
+ Contacts()
+ setContacts(contacts : HashMap<String, Contact>) : void
+ getContacts() : HashMap<String, Contact>
User
- collection : Collection
- contacts : Contacts
- password : String
- email : String
+ fromJson(json : String) : void
+ toJson() : String
+ User()
+ renameFolder(oldName : String, newName : String) : boolean
+ deleteFolder(folder : String) : void
+ addFolder(folder : String) : boolean
+ addDraft(message : Message) : void
+ moveMes(to : String, from : String, message : Message) : void
+ deleteMes(folder : String, message : Message) : void
+ addRes(message : Message) : void
+ addSen(message : Message) : void
+ addNewEmail(contactName : String, newEmail : String) : void
+ deleteContactInfo(contactName : String, editedEmail : String) : void
+ editContactInfo(json_contact : String) : void
+ editContactName(oldName : String, newName : String) : void
+ deleteContact(contactName : String) : void
+ addContact(contactName : String, email : List<String>) : boolean
+ setEmail(email : String) : void
+ getEmail() : String
+ setPassword(password : String) : void
+ setCollection(collection : Collection) : void
+ getCollection() : Collection
+ setContacts(contacts : Contacts) : void
+ getContacts() : Contacts
+ getPassword() : String
+ User(faris : String, s : String)
<<interface>> Filter
meetCriteria(list : LinkedList<Message>) : LinkedList<Message>
DateCriteria
~ value : Date
~ attributeName : String
+ meetCriteria(list : LinkedList<Message>) : LinkedList<Message>
+ DateCriteria(attributeName : String, value : long)
StringCriteria
~ value : String
~ attributeName : String
+ meetCriteria(list : LinkedList<Message>) : LinkedList<Message>
+ StringCriteria(attributeName : String, value : String)
NumCriteria

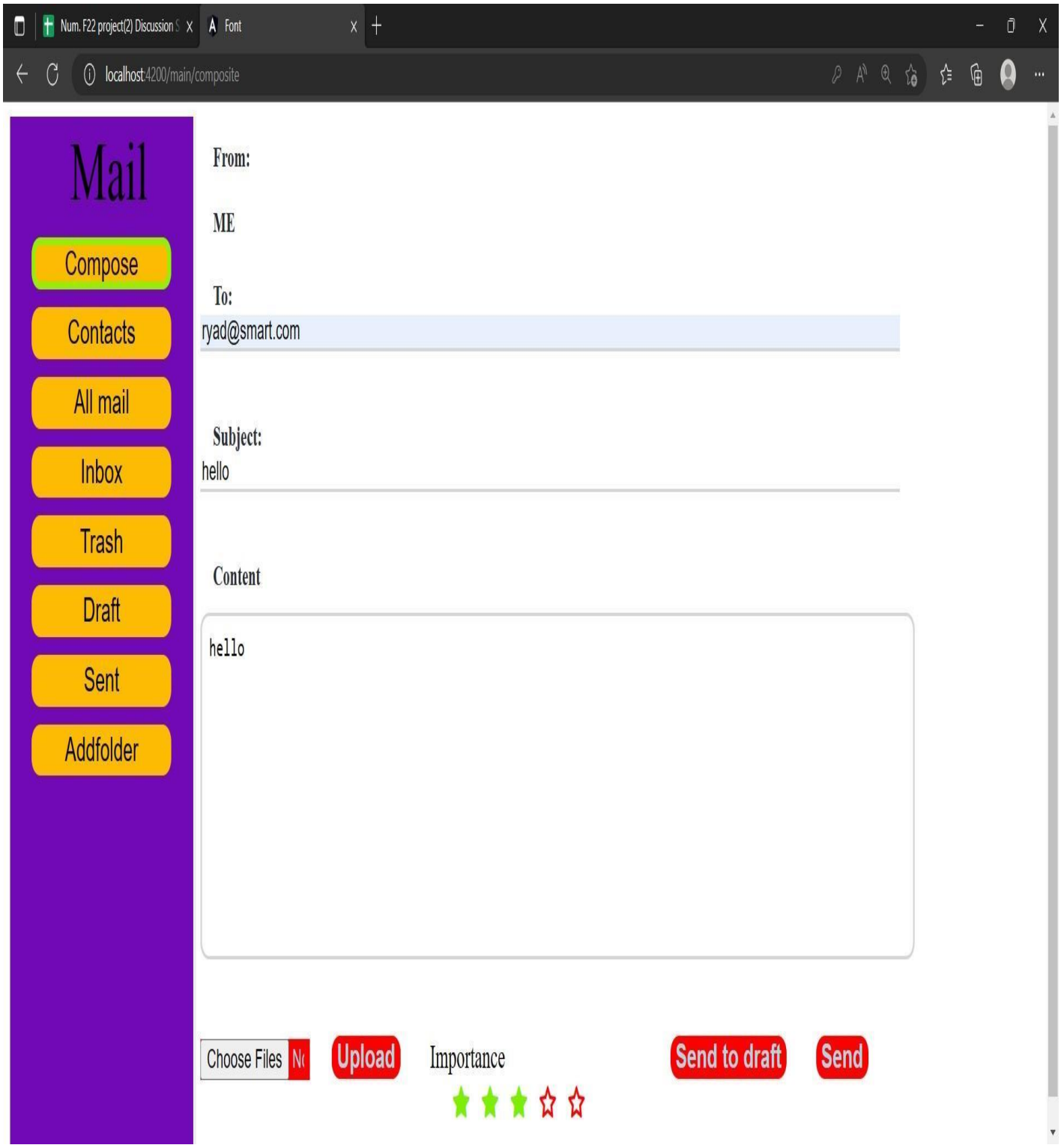


Front:

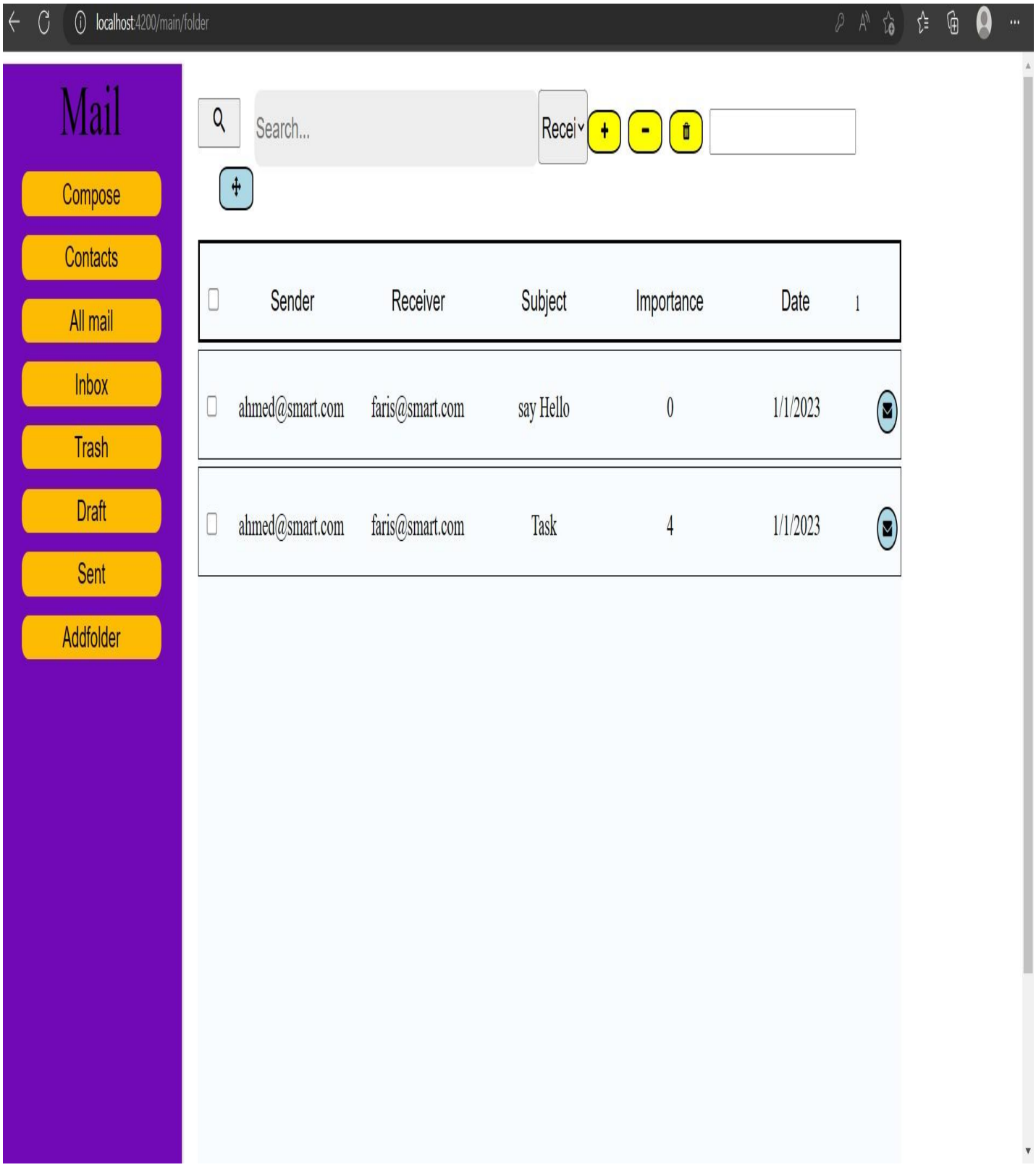
Login:



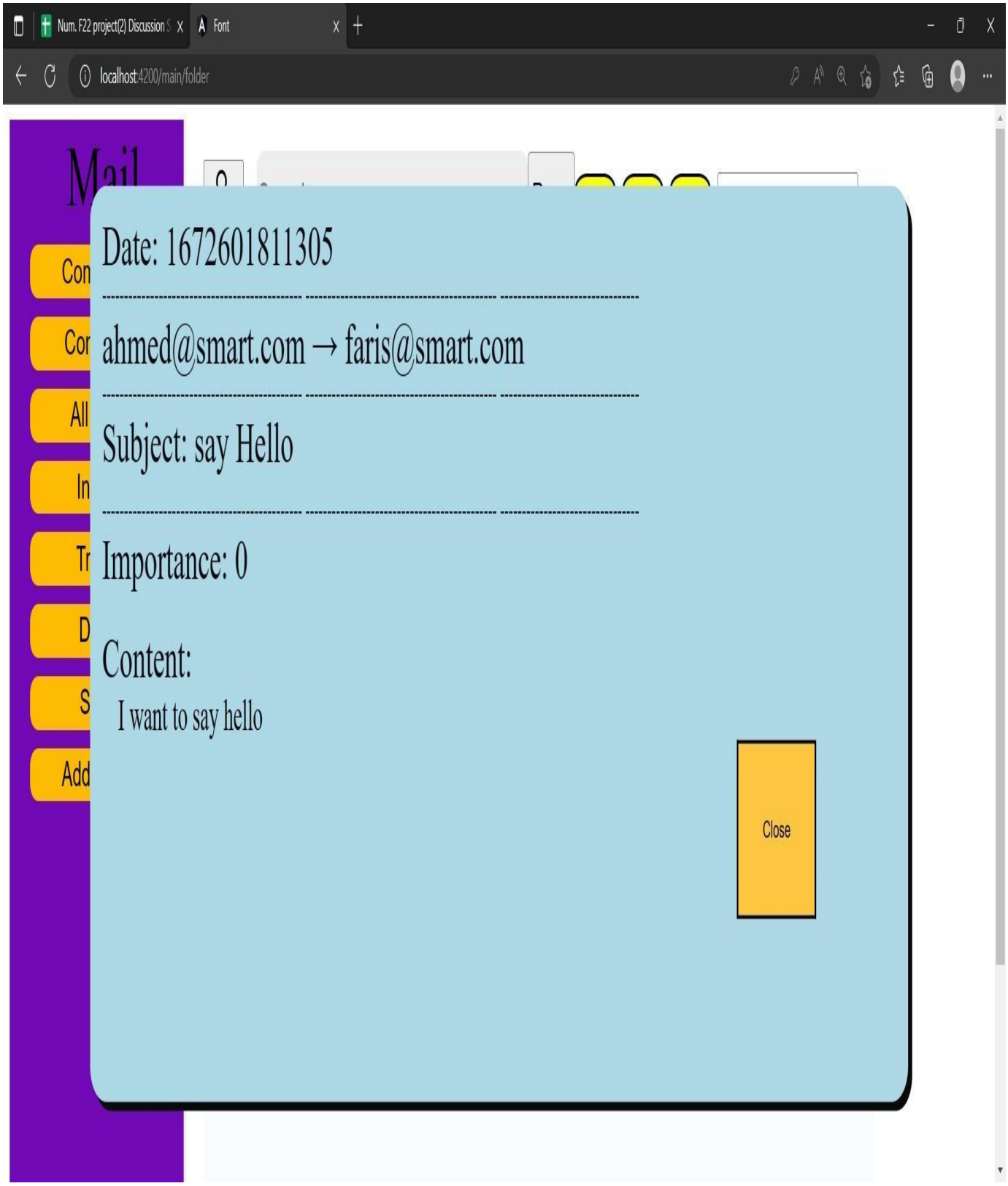
Compose message:


Num. F22 project(2) Discussion
Font
localhost:4200/main/composite
Mail
Compose
Contacts
All mail
Inbox
Trash
Draft
Sent
Addfolder
From:
ME
To:
ryad@smart.com
Subject:
hello
Content
hello
Choose Files
Upload
Importance
Send to draft
Send

Allmails:


localhost:4200/main/folder
Mail
Compose
Contacts
All mail
Inbox
Trash
Draft
Sent
Addfolder
Search...
Recei
Sender
Receiver
Subject
Importance
Date
ahmed@smart.com
faris@smart.com
say Hello
0
1/1/2023
ahmed@smart.com
faris@smart.com
Task
4
1/1/2023

Showing message:

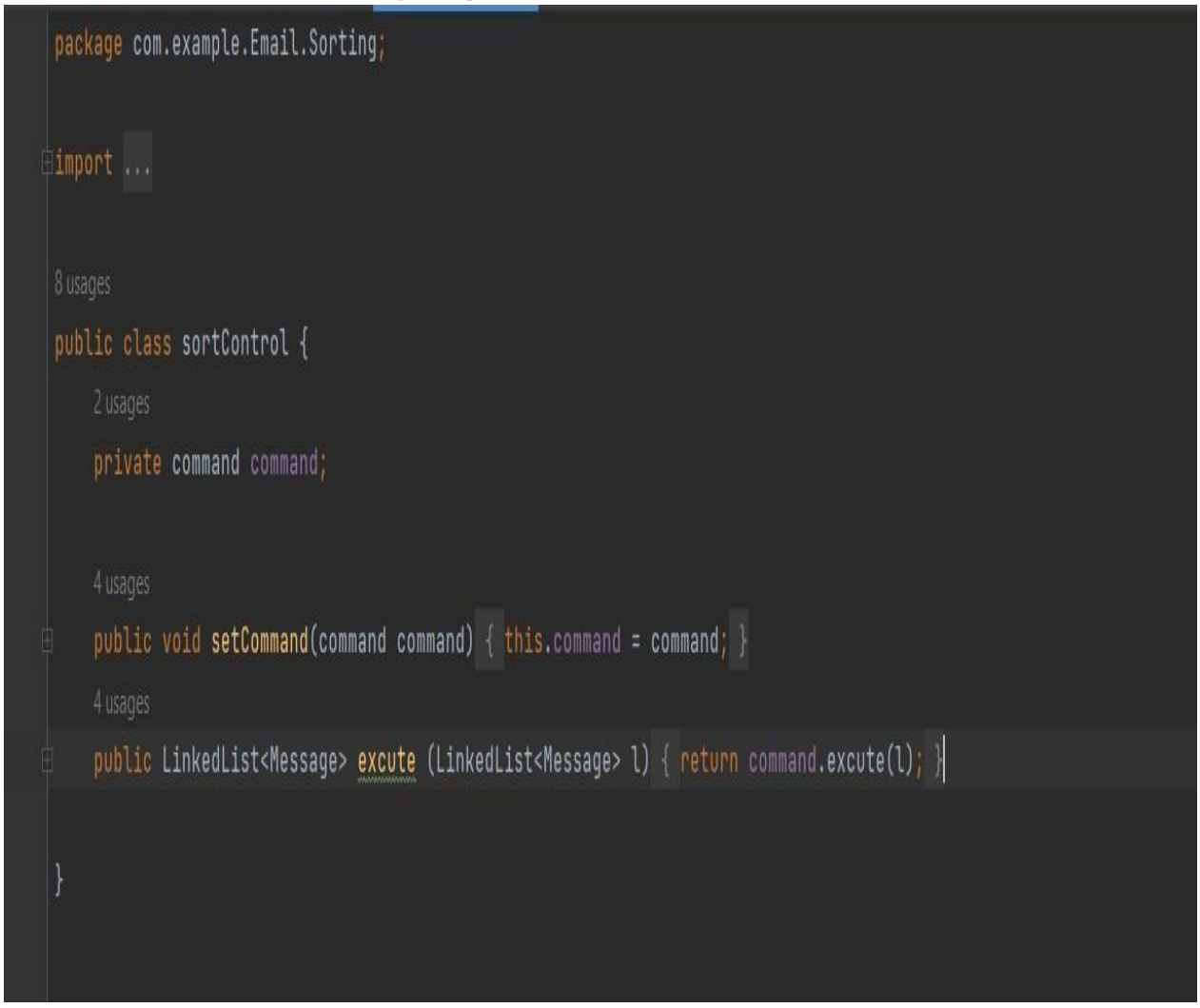


Design patterns:

1.filter design pattern:

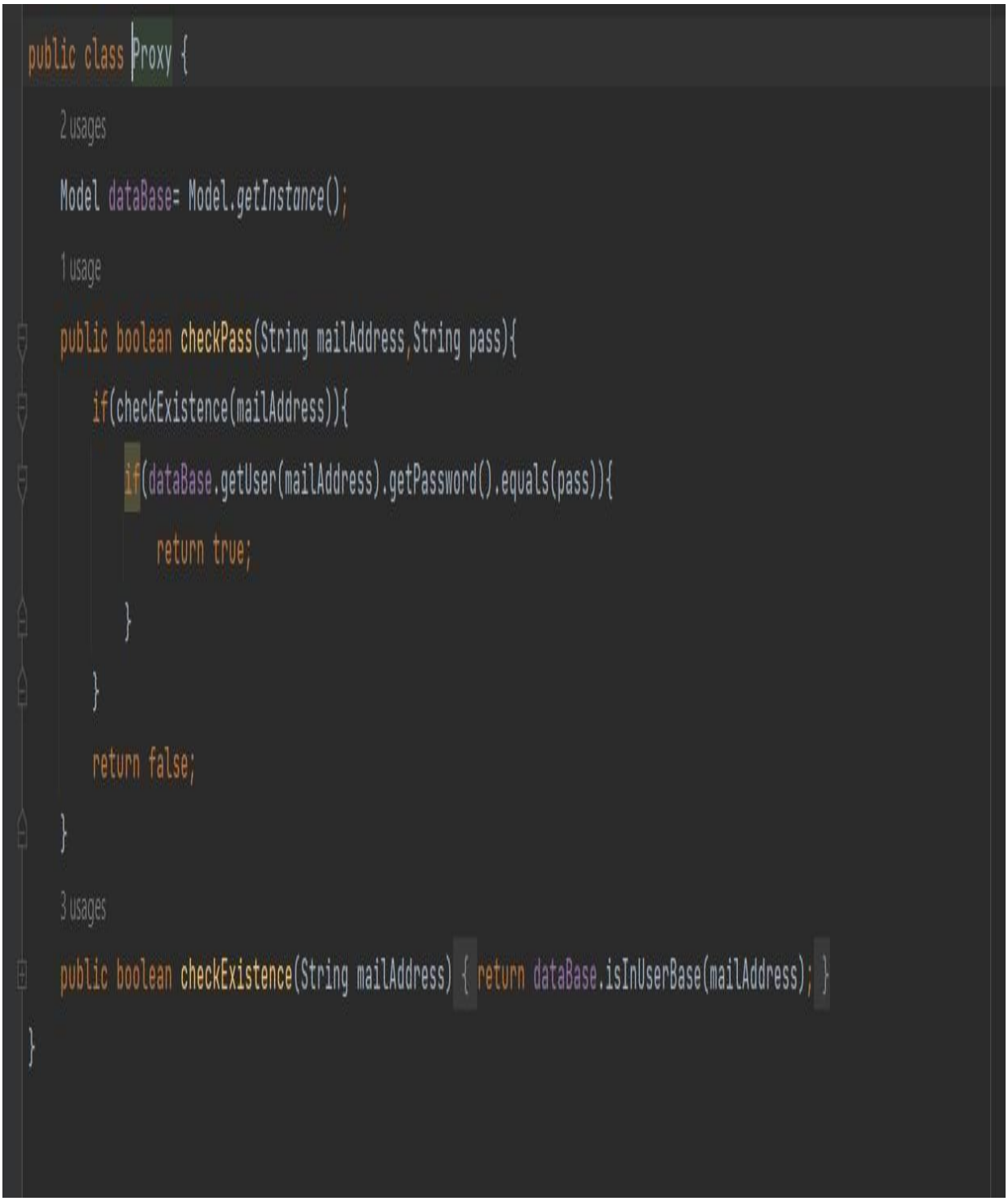
```
2 usages
public class FilterOperations {
    public LinkedList<Message> searchOnString(LinkedList<Message> messages, String attributeName, String value){
        Filter filter=new StringCriteria(attributeName,value);
        return filter.meetCriteria(messages);
    }
    public LinkedList<Message> searchOnNum(LinkedList<Message> messages,String attributeName,long value){
        Filter filter=new NumCriteria(attributeName,value);
        return filter.meetCriteria(messages);
    }
    public LinkedList<Message> searchOnDate(LinkedList<Message> messages, String attributeName, long value){
        Filter filter=new DateCriteria(attributeName,value);
        return filter.meetCriteria(messages);
    }
    1 usage
    public Contact searchOnContacts(Contacts contacts, String name) { return contacts.getContacts().get(name); }
}
```

2.command design pattern



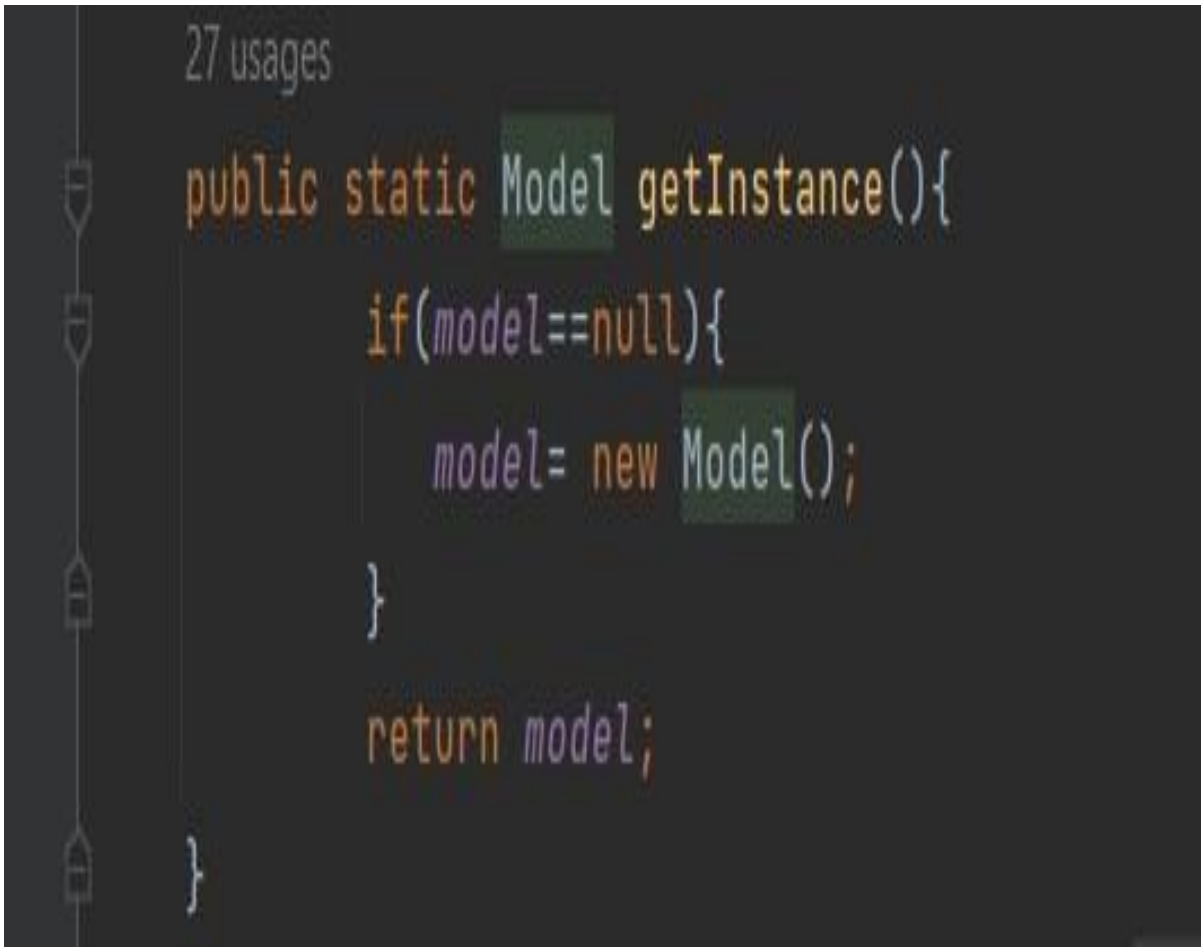
```
package com.example.Email.Sorting;

import ...

8 usages
public class sortControl {
    2 usages
    private command command;

    4 usages
    public void setCommand(command command) { this.command = command; }
    4 usages
    public LinkedList<Message> excute (LinkedList<Message> l) { return command.excute(l); }

}
```

3. proxy design pattern

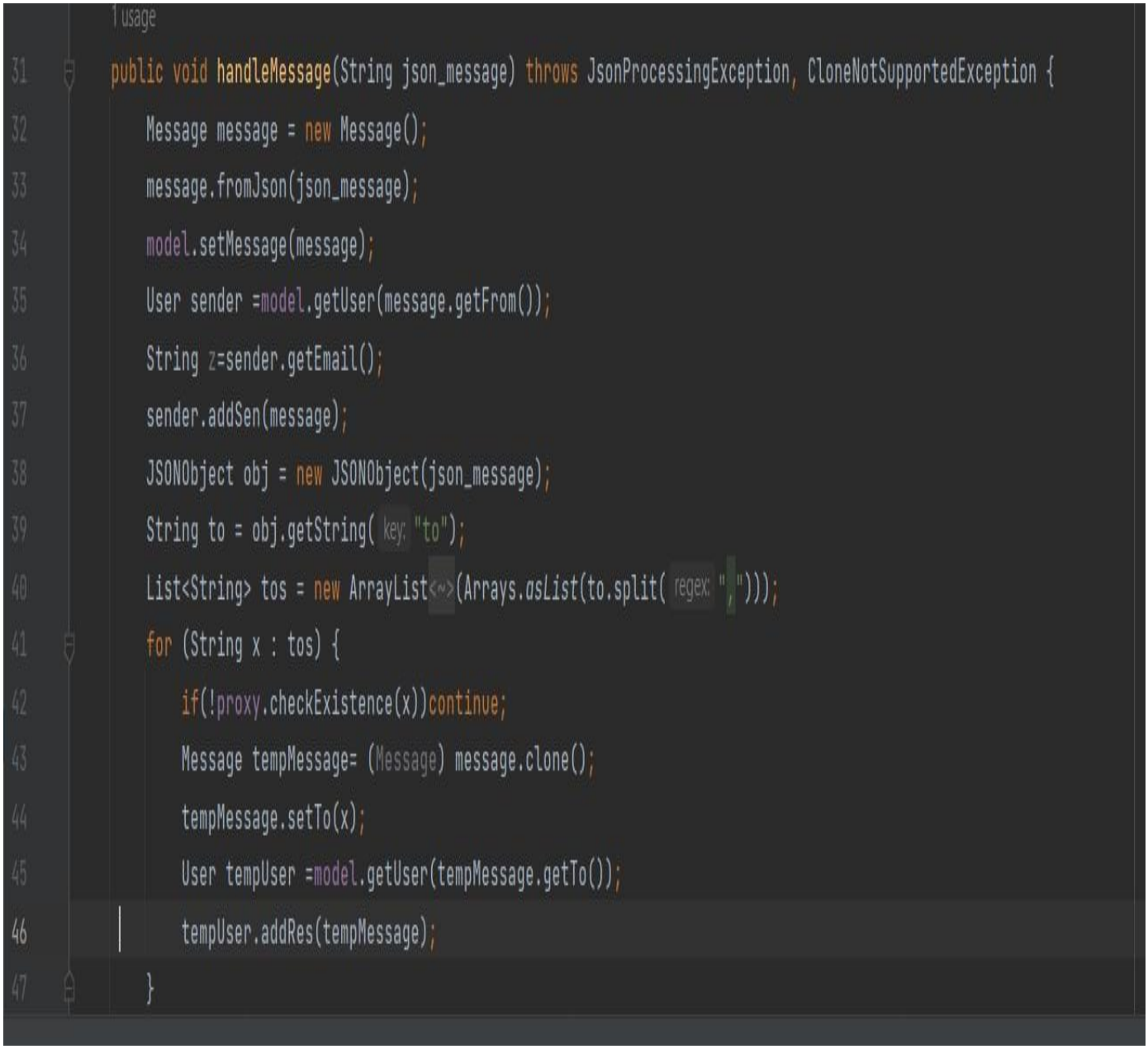


```
public class Proxy {
    2 usages
    Model dataBase= Model.getInstance();
    1 usage
    public boolean checkPass(String mailAddress,String pass){
        if(checkExistence(mailAddress)){
            if(dataBase.getUser(mailAddress).getPassword().equals(pass)){
                return true;
            }
        }
        return false;
    }
    3 usages
    public boolean checkExistence(String mailAddress) { return dataBase.isInUserBase(mailAddress); }
}
```

4.singleton design pattern



5.prototype design pattern



```
1 usage
public void handleMessage(String json_message) throws JsonProcessingException, CloneNotSupportedException {
    Message message = new Message();
    message.fromJson(json_message);
    model.setMessage(message);
    User sender =model.getUser(message.getFrom());
    String z=sender.getEmail();
    sender.addSen(message);
    JSONObject obj = new JSONObject(json_message);
    String to = obj.getString( key: "to");
    List<String> tos = new ArrayList<>(Arrays.asList(to.split( regex: ",")));
    for (String x : tos) {
        if(!proxy.checkExistence(x))continue;
        Message tempMessage= (Message) message.clone();
        tempMessage.setTo(x);
        User tempUser =model.getUser(tempMessage.getTo());
        tempUser.addRes(tempMessage);
    }
```

We used json files to save data base and load data base and we comnvert objects to json objects and sent it to front and convert json objects to objects In the front.

Model:

```java
package com.example.Email.Model;

import ...

60 usages
@Component
public class Model {
    9 usages
    HashMap<String , User> userBase=new HashMap<~>();
    8 usages
    HashMap<String , Message> mesBase=new HashMap<~>();
    3 usages
    private static Model model=null;
    2 usages
    protected static ObjectMapper mapper =new ObjectMapper();

    27 usages
    public static Model getInstance(){
```

We make id for every obj in front using func to random string from 10 char

```
static makeid() {
  var result = '';
  var characters = 'ABCDEFGHIJKLMNOPQRSTUVWXYZabcdefghijklmnopqrstuvwxyz0123456789';
  var charactersLength = characters.length;
  for (var i = 0; i < 10; i++) {
    result += characters.charAt(Math.floor( x: Math.random() * charactersLength));
  }
  return result;
}
```

User Guide:

- Should Run The code of Back end using InJ • Should

  Run angular (HTML , CSS,TYPESCRIPT) • Open local Host

  4200

To create new message click compose and to add new folder click add folder and to show all mails click inbox and to show mails in any folder click on it and to show contacts click on contacts.

You can send the mail or move it to drafts.

You can rename folder name or delete it.

To login click on login and to sign up click on sign up.

Vidoe Will BE with Pdf to show all Cons: