JNCT
LNCT Group of Colleges

- Department of computer science engineering -

Object-Oriented Programming Case Study

CS: 305

: Basic Email System in C++ :

Subtitles:
Simulating an Email System with Inbox, Outbox, and Message Sending Functionality.

Submitted by: CYBERCRAFTER

1. Saloni Chouhan (Leader) 0131CS221169
2. Shivam Sahu 0131CS221187
3. Sahil Irpache 0131CS221166

Guided by: Mr. Ravindra Tanwar

CONTENT :-

- Introduction to the Case Study.
- Brief overview of the email system.
- Important of object-oriented programming in this design,
- Use class diagram.
- Flow diagram.
- Class and object diagram.
- Entity-relation diagram.
- Input-output analysis of case study.
- Conclusion.

- Brief overview of the email system -

An email system is a way for people to send and receive messages electronically. It uses the internet to transmit messages from one person to another. Each user has an email address, and they can compose messages, attach files, and send them to other users. The messages are stored on servers, and users can access their emails from any device with internet access. Email systems also include features like folders, spam filters, and the ability to organize and manage messages efficiently.

- Importance of object-oriented programming in the design -

- In the context of an email system, object-oriented programming (OOP) refers to organizing and designing the code based on the principles of OOP. Here's a simplified explanation:

1. **Objects:** In an email system, objects could include elements like emails, contacts, and attachments. Each of these could be represented as objects in the code.
2. **Classes:** Classes are like blueprints for objects. For instance, you might have a "Message" class that defines the properties and behaviors of an email. This class could have attributes like sender, recipient, subject, and methods like send() and receive().
3. **Encapsulation:** Encapsulation involves bundling data (attributes) and the methods that operate on the data into a single unit, a class in this case. For example, the "Email" class encapsulates the data and functions related to emails.
4. **Inheritance:** Inheritance allows a class to inherit properties and behaviors from another class. In an email system, you might have a "Draft" class that inherits from the "Message" class, inheriting its properties and methods.
5. **Polymorphism:** Polymorphism allows different classes to be treated as instances of the same class through a common interface. For instance, both an "Email" class and a "Draft" class could implement a "send()" method, but each would execute differently.

- Requirements and Features -

- Inbox, Outbox, and Message Sending functionality : -

Inbox:

- The Inbox is like a digital mailbox where received emails are stored.
- It's the central location where users can view and manage incoming messages.
- Users can organize emails, mark them as read or unread, and perform actions such as replying or forwarding.

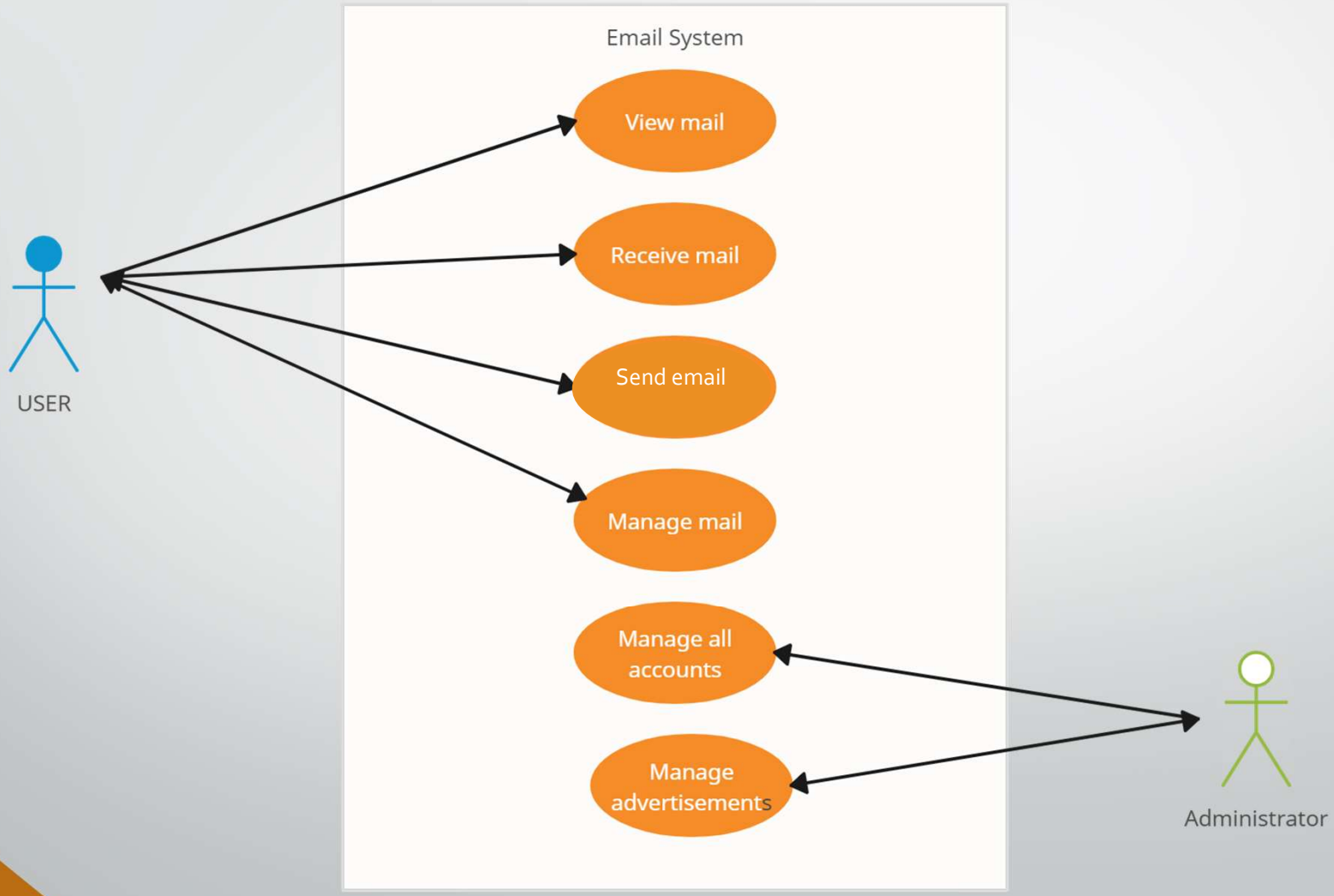
Outbox:

- The Outbox is where emails are temporarily stored before they are sent.
- When a user composes an email, it goes to the Outbox until it's ready to be sent.
- Once the email is sent, it moves from the Outbox to the recipient's Inbox.

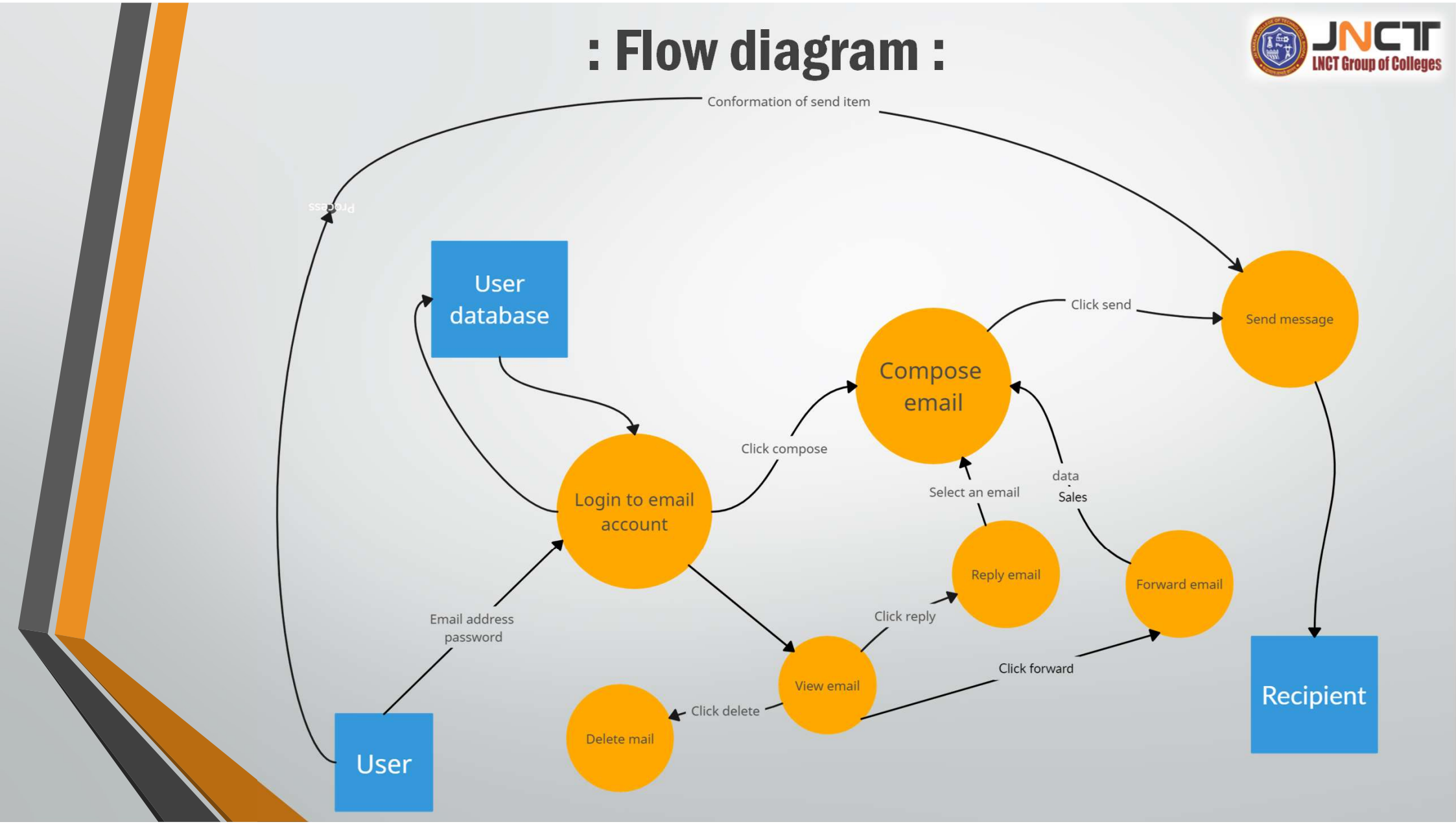
Message Sending functionality:

- Users compose new messages using the email system's interface or client.
- They enter the recipient's email address, write the message, and may attach files.
- Clicking "send" initiates the process of transferring the email from the Outbox to the recipient's Inbox.
- The email system handles the delivery process, ensuring the message reaches the intended recipient.

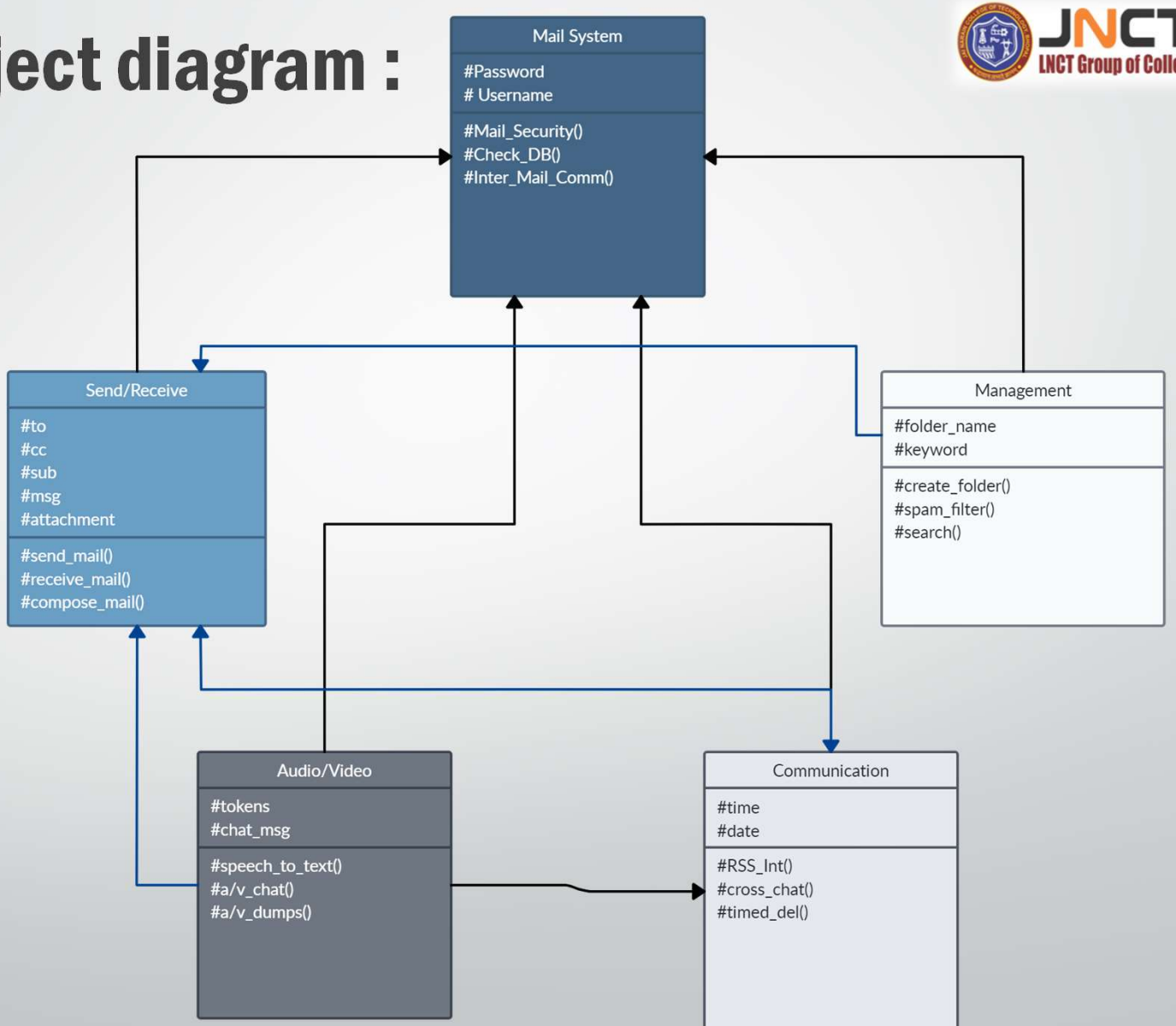
: Use case diagram :



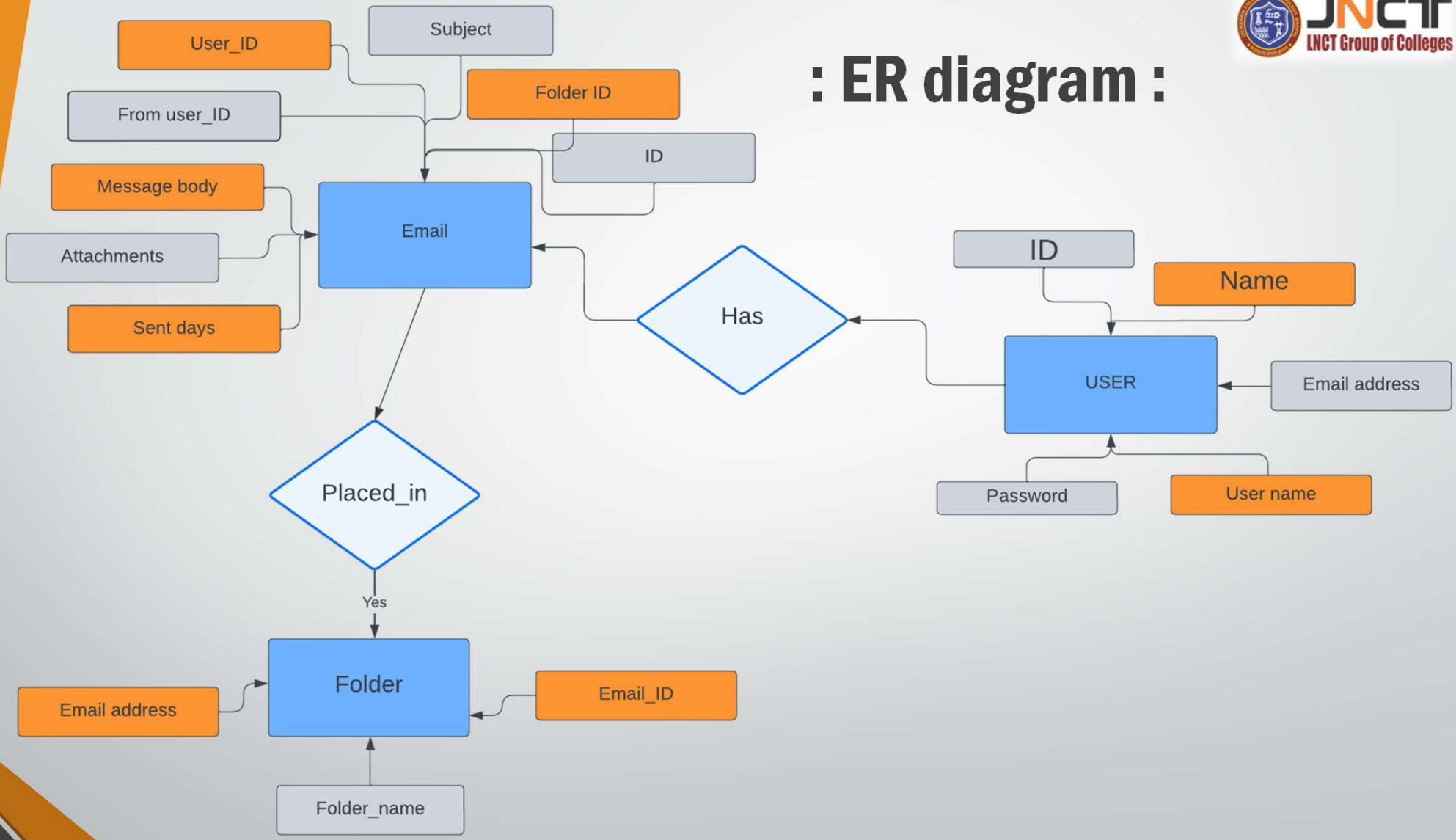
: Flow diagram :



: Class and object diagram :



: ER diagram :


User_ID
Subject
Folder ID
From user_ID
ID
Message body
Email
Attachments
Sent days
Has
ID
Name
USER
Email address
Password
User name
Placed_in
Yes
Folder
Email address
Email_ID
Folder_name

: Input-output analysis of case study :

JNCT
LNCT Group of Colleges

>Input

Contact Us

Name:

jessyyy

Email:

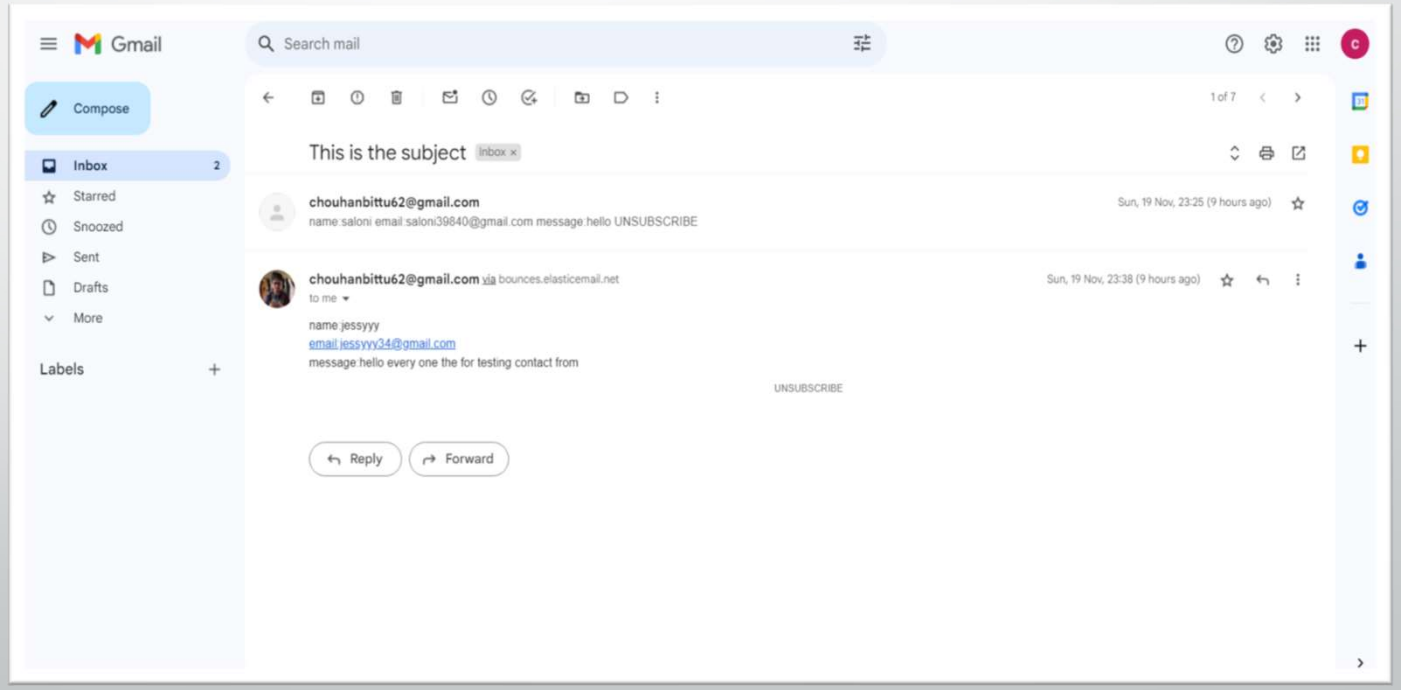
jessyyy123@gmail.com

Message:

this is for testing contact form.

submit

Output<



THANK YOU

By:-

Saloni Chouhan(Leader)

0131CS221169

Shivam Sahu

0131CS221187

Sahil Irpache

0131CS221166

In guidance of : Mr. Ravindra Tanwar

CYBERCRAFTER