**Project Title: SECURE FILE SHARING SYSTEM**

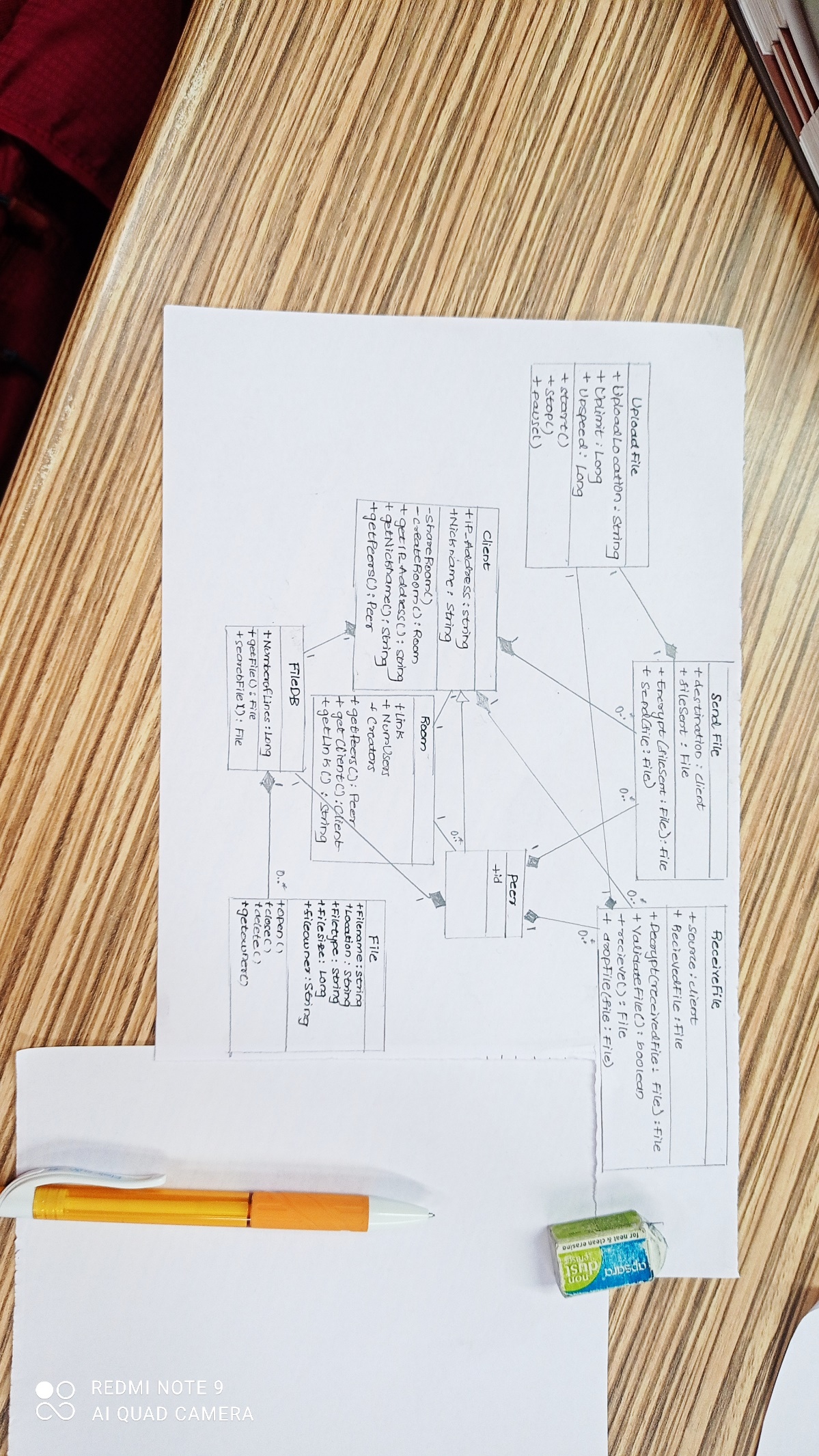
Team Details:

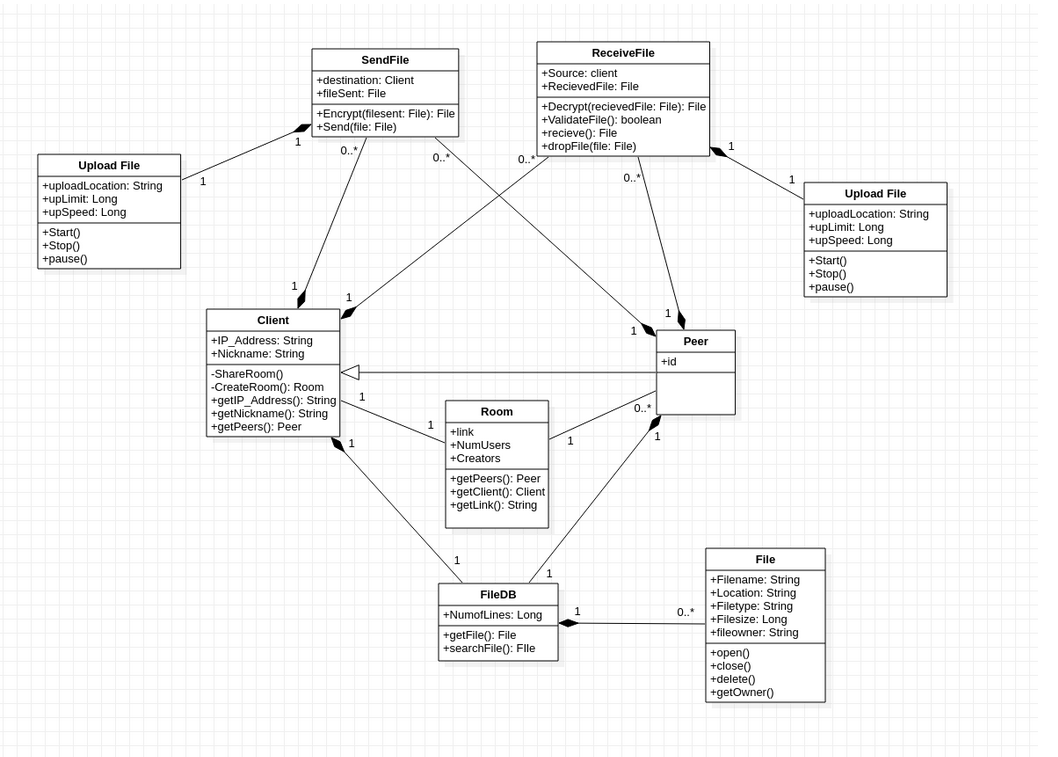
1. S. NIHARIKA - PES1UG20CS583

2. SAHANA RAO - PES1UG20CS588

3. SHREYA - PES1UG20CS591

**Class Diagram**

****



**Design Principles :**

**SOLID**

**● S - Single Responsibility Principle (SRP)**

All files in the model have single logical responsibility.

Eg: Classes like sendFile, receiveFile, Upload and Download file have only one responsibility of sending, receiving, uploading and downloading.

**● O - Open-Closed Principle (OCP)**

Class is open for extensions but closed for modifications.

Eg: Class peer inherits the properties of client class by

extending its properties.

**● L - The Liskov Substitution Principle (LSP)**

An object of a superclass should be replaceable by objects of its subclasses without causing issues in the application.

Eg: A client object can be replaced with a peer object without

any flaws.

NOTE: Client is the base class and peer is the derived class

**● I - Interface Segregation Principle (ISP)**

Make fine grained interfaces that are client-specific. Clients should not be forced to implement interfaces they do not use

There are no interfaces or classes with functionality

which overburden them or ones which are not used by

the client.

**● D - Dependency Inversion Principle (DIP)**

High level modules should not depend on low level modules, both should depend upon abstractions.

There is no high level class that directly depends on

classes at lower levels.

**GRASP PRINCIPLES:**

**Creator:** In the context of a secure file sharing system, this could mean that the system should have a dedicated module responsible for creating and managing new user accounts and permissions.

**Information Expert:** This principle suggests that responsibility for a particular piece of information should be assigned to the object that knows the most about that information. In the context of a secure file sharing system, this could mean that the system should have a dedicated module or class responsible for managing encryption and decryption of files, as well as other security-related tasks.

**Controller:** System should have a dedicated controller responsible for managing user authentication, file access permissions, and other security-related tasks.

**Low Coupling:** In the context of a secure file sharing system, this could mean designing the system so that changes to the encryption module do not affect the user authentication module.

**High Cohesion:** In the context of a secure file sharing system, this could mean designing the encryption module so that it is focused solely on encryption and decryption tasks, without trying to handle other security-related tasks like user authentication or file access permissions.