“VPN Network”

UE20CS352 : OOADJ

MINI PROJECT

Submitted by :

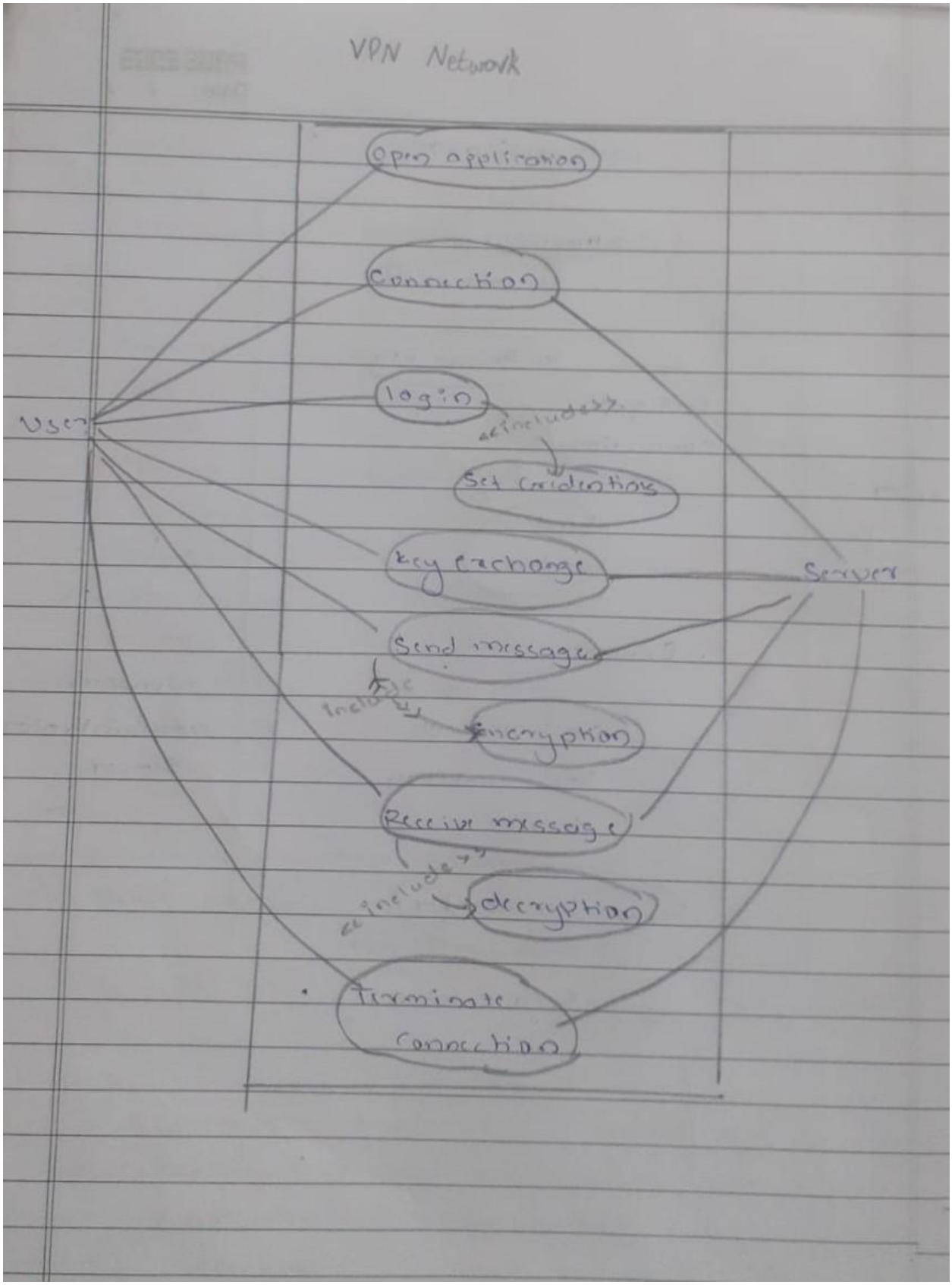
SANGAMESH S	PES1UG2CS374
SHASHANK K	PES1UG20CS392
SHREYANSH S	PES1UG20CS404
SIDDHARTH K	PES1UG20CS424

Problem statement and synopsis

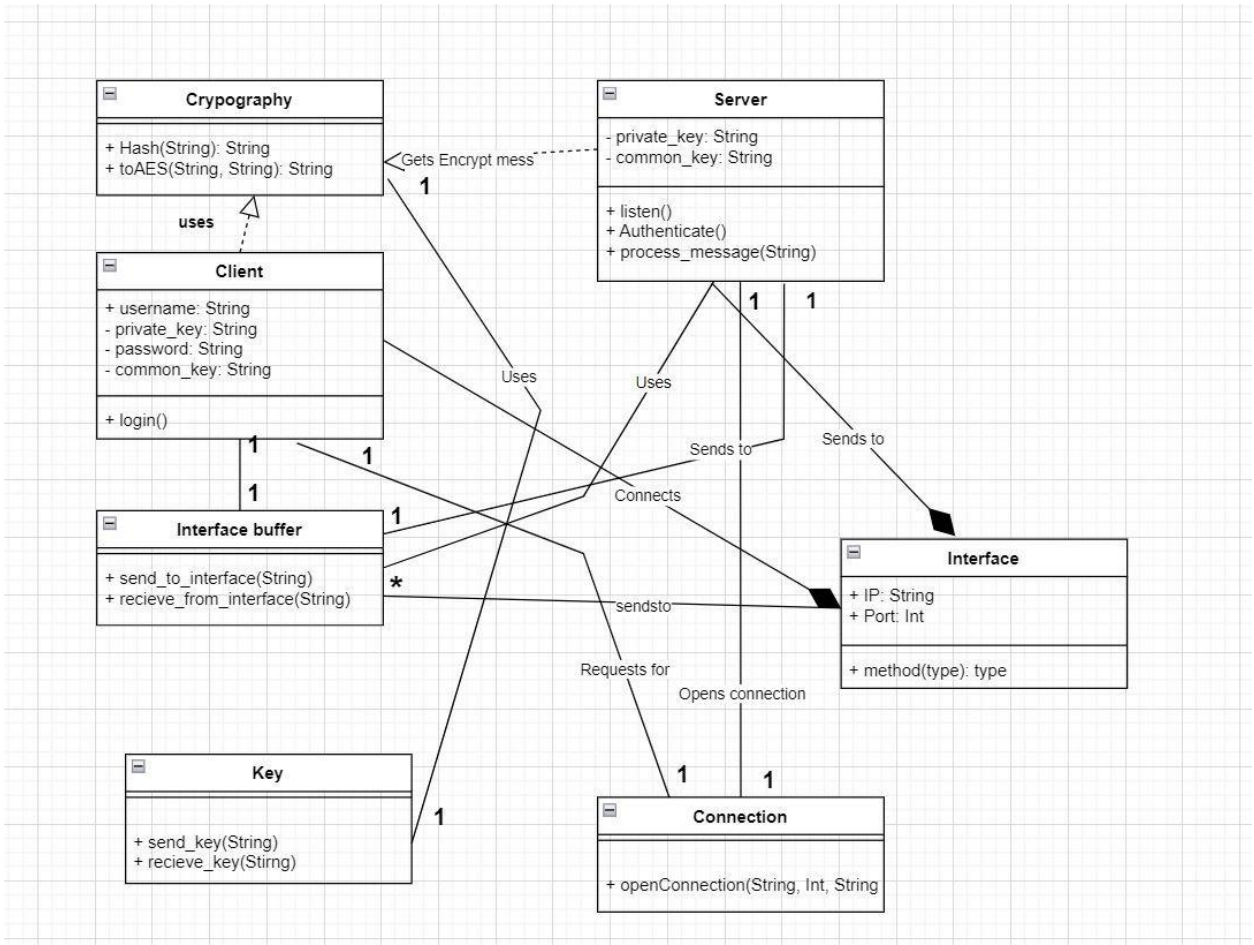
The aim of this project is to create a cost-effective and easy-to-use virtual private network (VPN) solution that allows remote users to securely access their private networks and sensitive information. Traditional methods of accessing private networks can be expensive and difficult to manage, and so there is a need for a simple and reliable VPN solution.

The proposed VPN solution will provide authentication and authorization mechanisms, encryption of data traffic, and support for multiple operating systems. It will be scalable, allowing for easy expansion as the number of users and network resources grow. The VPN solution will also provide reliable and fast performance while ensuring the confidentiality, integrity, and availability of transmitted data.

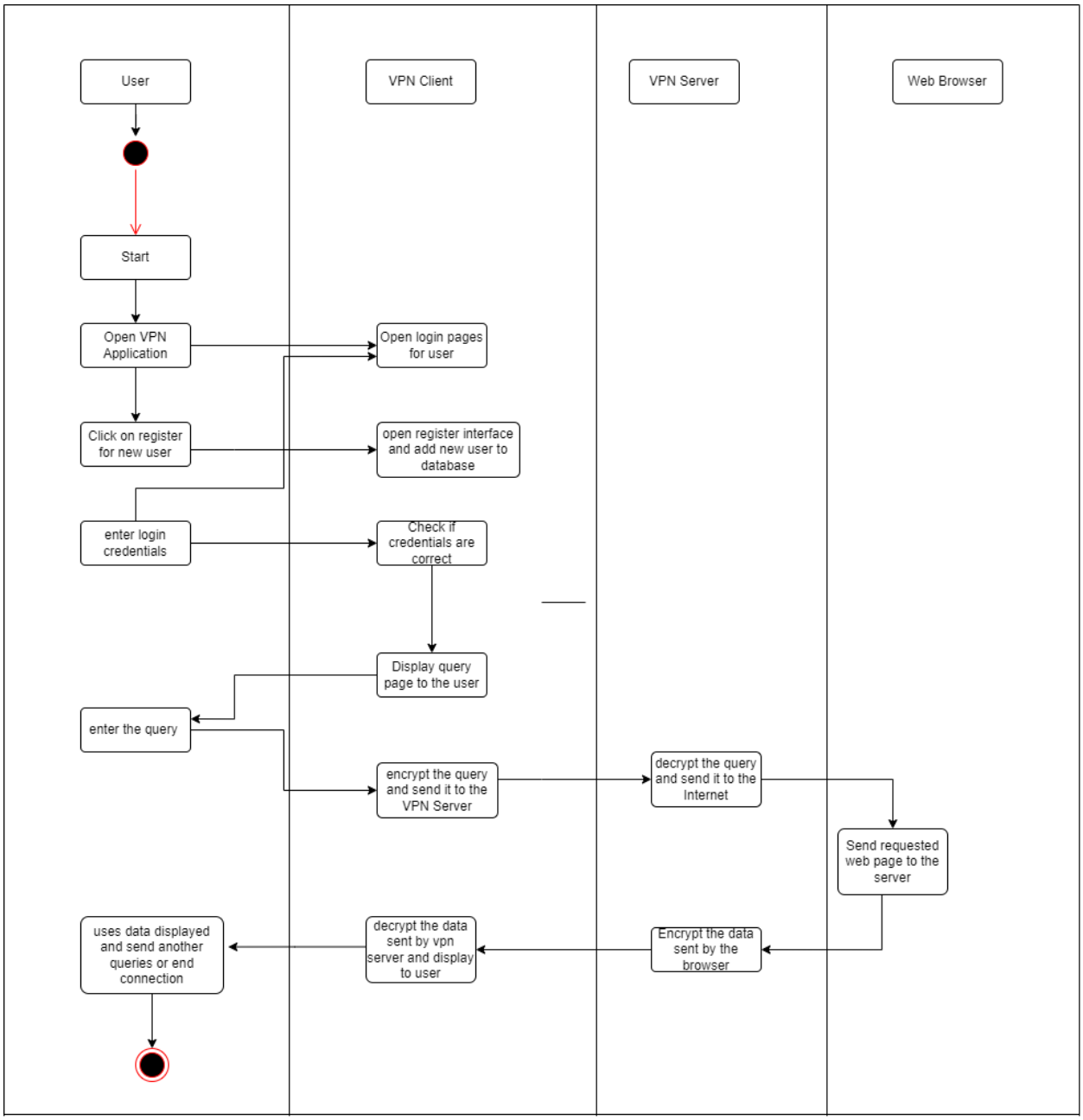
Use case diagram


VPN Network
Open application
Connection
login
User
<<includes>>
Set validations
key exchange
Server
Send message
<<include>>
encryption
Receive message
<<include>>
decryption
Terminate
connection

Class diagram



Activity Diagram



Architecture pattern , Design principles and design patterns used in the project

SRP(single responsibility principle) : most of the classes in the project have responsibility over a single part

of the functionality. For eg. Client and Server have their separate module.It ensures low coupling and testing becomes easy.

High Cohesion: In AES encryption algorithm, both encryption and decryption is done in the same class and also the similar responsibilities are taken by a single class.

Low Coupling: Every class acts as an independent module and there is not much interaction between classes. input to the class is either through interface or it is defined in the class like in deffie hellman key exchange algorithm shared key is given by user and all other parameter are defined in the class and similarly in aes algorithm only the input from other class is security key

ocp principle(open for extension and closed for modification) : the above principle in designing combined working of aes and deffie helman key exchange protocol the key exchange algorithm can be changed or extended without affecting aes algorithm or how it is connected to aes algorithm and also in similar situations

Github link to the code

https://github.com/NobleTempest/vpn

Individual contributions to the project

Sanagmesh(PES1UG20CS374)

AES algorithm implementation in java

User authentication

Shashank S(PES1UG20CS392)

Deffie helman key exchange protocol

SHREYANSH Srivastava(PES1UG20CS404)

Report

User Interface

SIDDHARTH K(PES1UG20CS424)

User interface

Server interface