# Client-Server Banking System

ITIDA-FINAL PROJECT REPORT

Submitted by : Mohamed Adel Amin
Submitted to : Coretech/IMT Headquarters
Date : 1/7/2024

Coretech
Innovations

School
A coretech Company

# Table of Contents

# Introduction

In the ever-evolving landscape of the financial industry, technological advancements play a pivotal role in enhancing the efficiency, security, and accessibility of banking systems. This report delves into the design and implementation of a client-server banking system, leveraging the power of TCP sockets and the Qt framework. The goal of this project is to develop a robust and scalable solution that facilitates secure communication between clients and servers, fostering a seamless and responsive banking experience.

As financial institutions continue to embrace digital transformation, the importance of reliable and secure banking systems cannot be overstated. The utilization of Transmission Control Protocol (TCP) for communication ensures a dependable and error-free data exchange between clients and servers. This report explores how TCP sockets provide a foundation for establishing a stable connection infrastructure, allowing for the secure transmission of sensitive financial data.

Furthermore, the integration of the Qt framework adds a layer of versatility and user-friendly features to the client-server banking system. Qt's cross-platform capabilities and intuitive development tools empower developers to create dynamic and visually appealing user interfaces, enhancing the overall user experience for both clients and bank personnel.

## CLIENT APPLICATION

The Client Application is responsible for user or admin interface for sending a specific request , which means Sending a specific order to the server to do something . These orders are the User and Admin class functions Intended to perform a specific functionality then sending the request to the socket file and wait for a response from the Server to fulfill client needs . The Admin and User Class both inherits from the Client class Which includes shared methods and parameters between both user and admin .

## SERVER APPLICATION

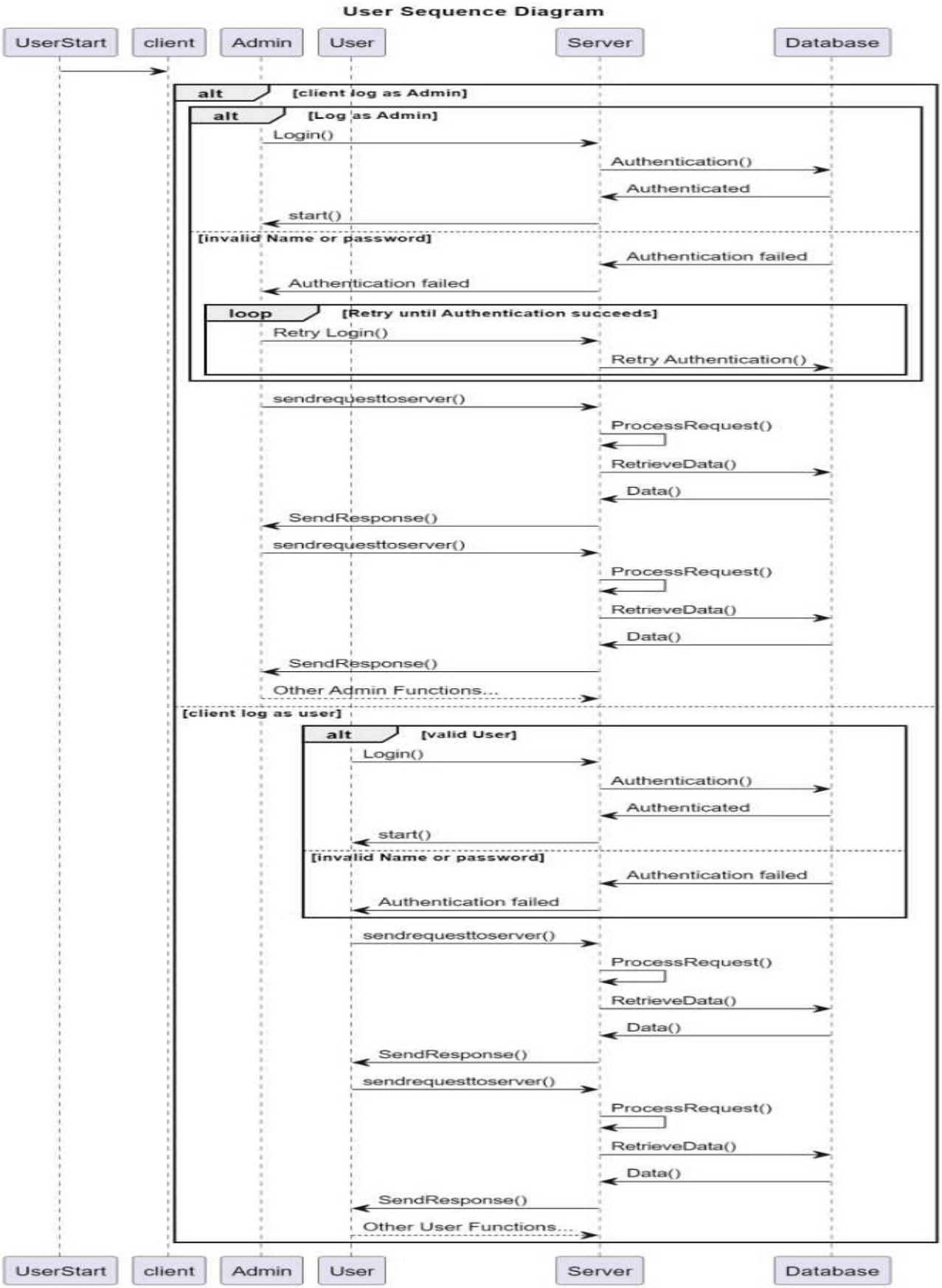
The Server Application is responsible for responding to client requests whether user or admin by sending the response to the socket file and receiving it in the client section with the returned data from each request given . The JsonHandler Class contains methods which will be called by the server Application for the right response for each request , it writes to and reads from the two data base files created to perform the functionality of each request . Two database files has been created , the BANK_DB file which contains all user information and the Login file which contains User and Admin login information . The server class receives the Client application requests and then check for login credentials by each client to be able to access the System database to send the right response . the JsonHandler class methods will be called specifically based on each request to make changes on the two database files . The server class inherits the JsonHandler class To easily use it's methods and parameters to be able to check first on the upcoming requests and then send the right response for each one calling JsonHandler class methods in the server source file.

## DRIVE LINK

Drive link contains video of the Project :

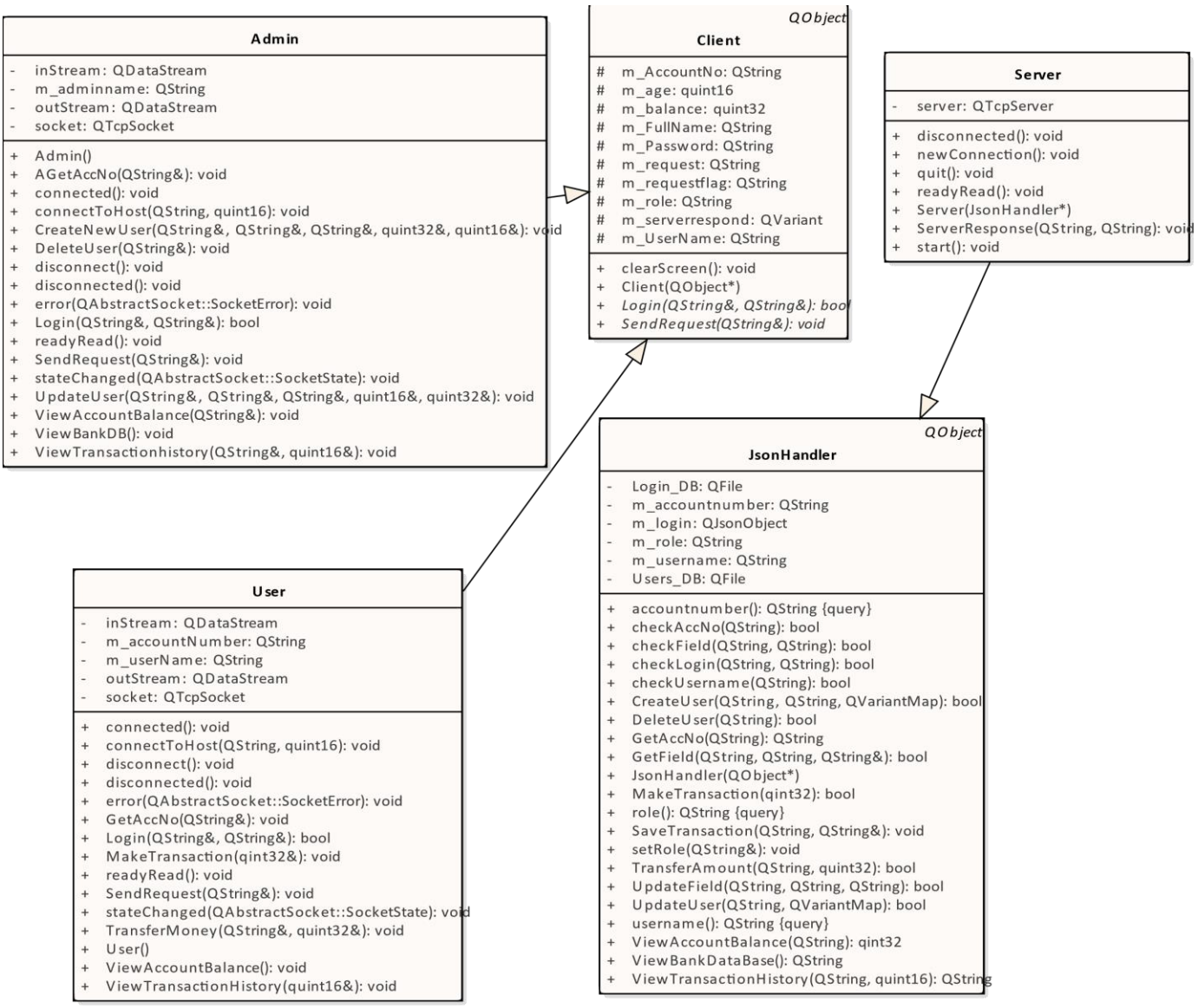
https://drive.google.com/drive/folders/1L-g0K4h2JP94V1GEQRSuUoqcdjc6ciag?usp=drive_link

## SEQUENCE DIAGRAM



## CLASS DIAGRAM

**A class diagram is a type of static structure diagram in the Unified Modeling Language (UML) that illustrates the structure of a system by showing the classes, their attributes, methods, and the relationships between them. Here is a breakdown of the components typically found in a class diagram:**



## CONCLUSION

In conclusion, the development and implementation of a client-server banking system utilizing TCP sockets and the Qt framework have yielded a robust and secure solution for modern financial institutions. Throughout the course of this project, several key observations and outcomes have emerged, highlighting the strengths and benefits of the chosen technologies.

**1. Reliability and Stability:** The integration of TCP sockets as the communication protocol has proven to be instrumental in ensuring a reliable and stable connection between the client and server components of the banking system. The inherent characteristics of TCP, such as error-checking and guaranteed delivery, contribute to the overall dependability of data transmission.

**2. Security Measures:** Security is paramount in the realm of banking systems, and the use of TCP sockets facilitates the implementation of encryption and other security measures to safeguard sensitive financial data. The secure communication channel established through TCP adds an essential layer of protection against potential threats, ensuring the confidentiality and integrity of the transmitted information.

**3. Scalability and Responsiveness:** The client-server architecture employed in this banking system allows for seamless scalability. As the number of clients grows, the server can efficiently handle multiple connections simultaneously, ensuring responsiveness and optimal performance. This scalability is crucial in accommodating the dynamic demands of a modern banking environment.

**4. User-Friendly Interface with Qt:** The integration of the Qt framework has enhanced the user experience by providing a visually appealing and intuitive interface. Qt's cross-platform capabilities ensure consistency across different operating systems, while its design tools enable developers to create dynamic and interactive interfaces that meet the expectations of both clients and banking personnel.

**5. Future Development and Enhancements:** The modular architecture of the client-server banking system allows for future development and enhancements. Whether it be the introduction of new banking features, improved security measures, or support for additional platforms, the flexibility of the implemented solution paves the way for continuous innovation and adaptation to evolving industry standards.

In essence, the client-server banking system developed with TCP sockets and Qt stands as a testament to the synergy between reliable communication protocols and versatile user interface frameworks. This project not only addresses the immediate needs of a secure and responsive banking system but also lays the foundation for ongoing advancements in the ever-evolving landscape of financial technology.