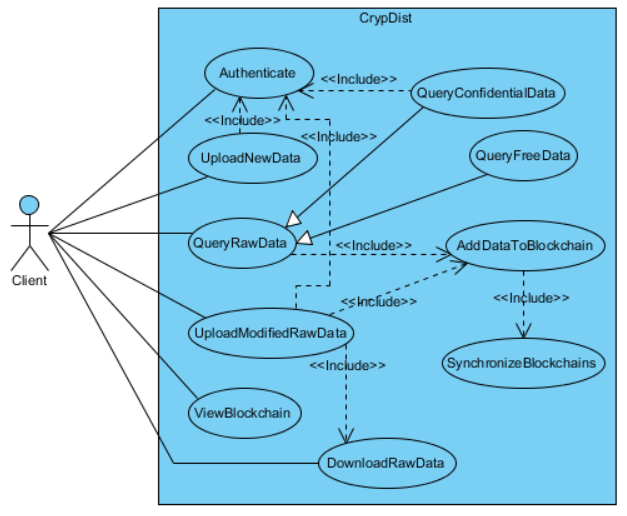
**USE CASE MODEL**

****

**Use case name:** Authenticate

**Participating actor:** Client

**Pre-condition:** Client must be in the main menu.

**Post-condition:** System authenticates the client.

**Entry condition:** System opens the authentication screen.

**Exit conditions:**

* Client receives the confirmation.
* Client receives the rejection.

**Main flow of events:**

* Client chooses the option to authenticate.
* System displays the authentication screen.
* Client enters his information.
* System checks the database and give the message **accordingly**.

**////////////////////////////////////////////////////////////////////////////**

**Use case name:** QueryConfidentialData

**Participating actor:** Client

**Pre-conditions:**

* Client must have been authenticated to the system.
* Client must be in the main menu.

**Post-condition:** System returns the queried data.

**Entry condition:** System opens the querying screen.

**Exit condition:**

* System displays the data.
* System gives an SQL syntax error.

**Main flow of events:**

* Client chooses the option to query data.
* System displays the querying screen.
* Client enters the SQL command.
* System executes the command and returns the data.

**Alternative flows:**

* System displays SQL syntax error if the command includes it.

**////////////////////////////////////////////////////////////////////////////**

**Use case name:** QueryFreeData

**Participating actor:** Client

**Pre-condition:** Client must be in the main menu.

**Post-condition:** System returns the queried data.

**Entry condition:** System opens the querying screen.

**Exit condition:**

* System displays the data.
* System gives an SQL syntax error.

**Main flow of events:**

* Client chooses the option to query data.
* System displays the querying screen.
* Client enters the SQL command.
* System executes the command and returns the data.

**Alternative flows:**

* System displays SQL syntax error if the command includes it.

**////////////////////////////////////////////////////////////////////////////**

**Use case name:** UploadNewData

**Participating actor:** Client

**Pre-condition:** Client must be in the main menu.

**Post-condition:** System uploads the data.

**Entry condition:** System opens the upload screen.

**Main flow of events:**

* Client chooses the option to upload data.
* Client browses for the data.
* System uploads the data.

**////////////////////////////////////////////////////////////////////////////**

**Use case name:** DownloadRawData

**Participating actor:** Client

**Pre-condition:** Client must be in the main menu.

**Post-condition:** System returns the requested data.

**Entry condition:** System opens the update screen.

**Main flow of events:**

* Client chooses the option to update data.
* Client requests the data by using the results of the previous queries.
* System returns the data.

**////////////////////////////////////////////////////////////////////////////**

**Use case name:** UploadModifiedRawData

**Participating actor:** Client

**Pre-condition:** Client must have been downloaded some part of the data.

**Post-condition:** System uploads the modified data.

**Entry condition:** System opens the upload screen.

**Exit condition:** Client receives the confirmation message.

**Main flow of events:**

* Client browses for the data.
* System uploads the data and gives the confirmation message.

**////////////////////////////////////////////////////////////////////////////**

**Use case name:** AddDataToBlockchain

**Participating actor:** Client

**Pre-condition:** Client must have been uploaded or queried data.

**Post-condition:** System adds the data to the chain.

**Main flow of events:**

* System adds the data to the chain.

**////////////////////////////////////////////////////////////////////////////**

**Use case name:** ViewBlockchain

**Participating actor:** Client

**Pre-condition:** Client must be in the main menu.

**Post-condition:** System displays the block information.

**Main flow of events:**

* Client chooses the option to view the block information in the main screen.
* System displays the information.

**////////////////////////////////////////////////////////////////////////////**

**Use case name:** SynchronizeBlockchains

**Participating actor:** Client

**Pre-condition:** System must have added data to the blockchain.

**Post-condition:** System synchronizes the blockchains.

**Main flow of events:**

* System synchronizes the blockchains.

**SCENARIOS**

**Query Confidential Data**

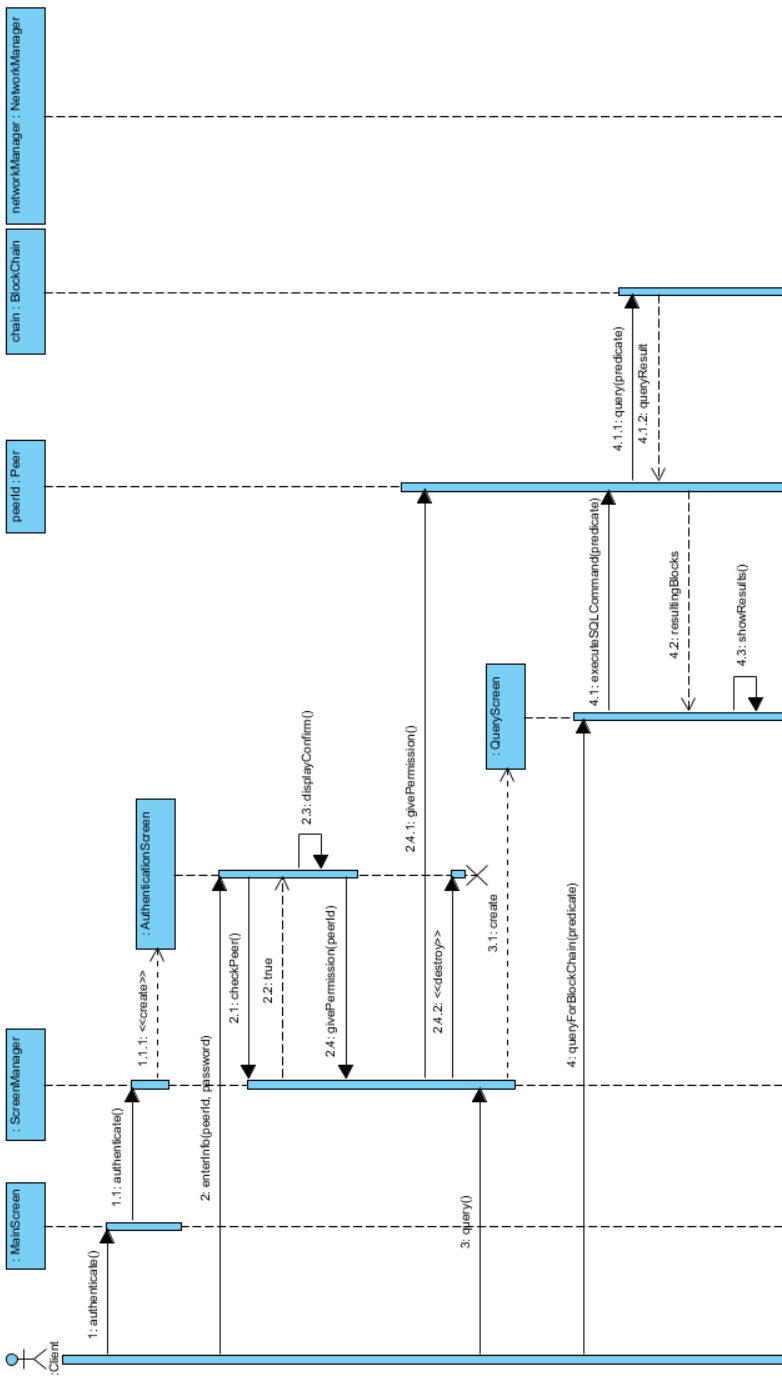
Client selects the authentication option in the main menu. Then System opens the authentication screen. Client enters his information and System checks the information. After confirmation, System returns to main menu and client chooses the option to query data. Then System opens the data query screen. Client enters an SQL command to the text area in the screen. System parses the command and executes it in the server. After that, System returns the data in a binary file.

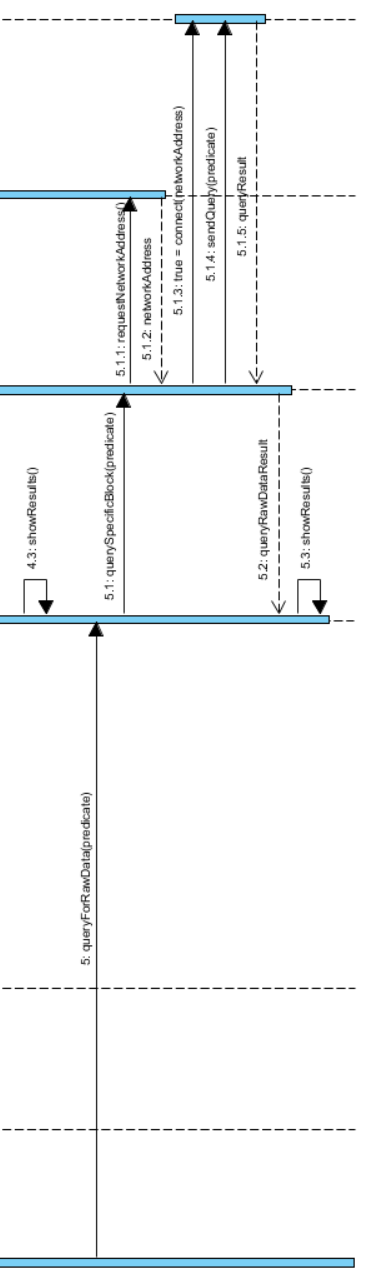
**Upgrade Data**

After authentication, Client selects the option to update data. System opens the data update screen. Then, Client selects the option to download the data by using the results of query. System returns the data. After that, Client selects the option to upload the modified data.

**SEQUENCE DIAGRAMS**

The following diagram belongs to the query confidential data scenario.

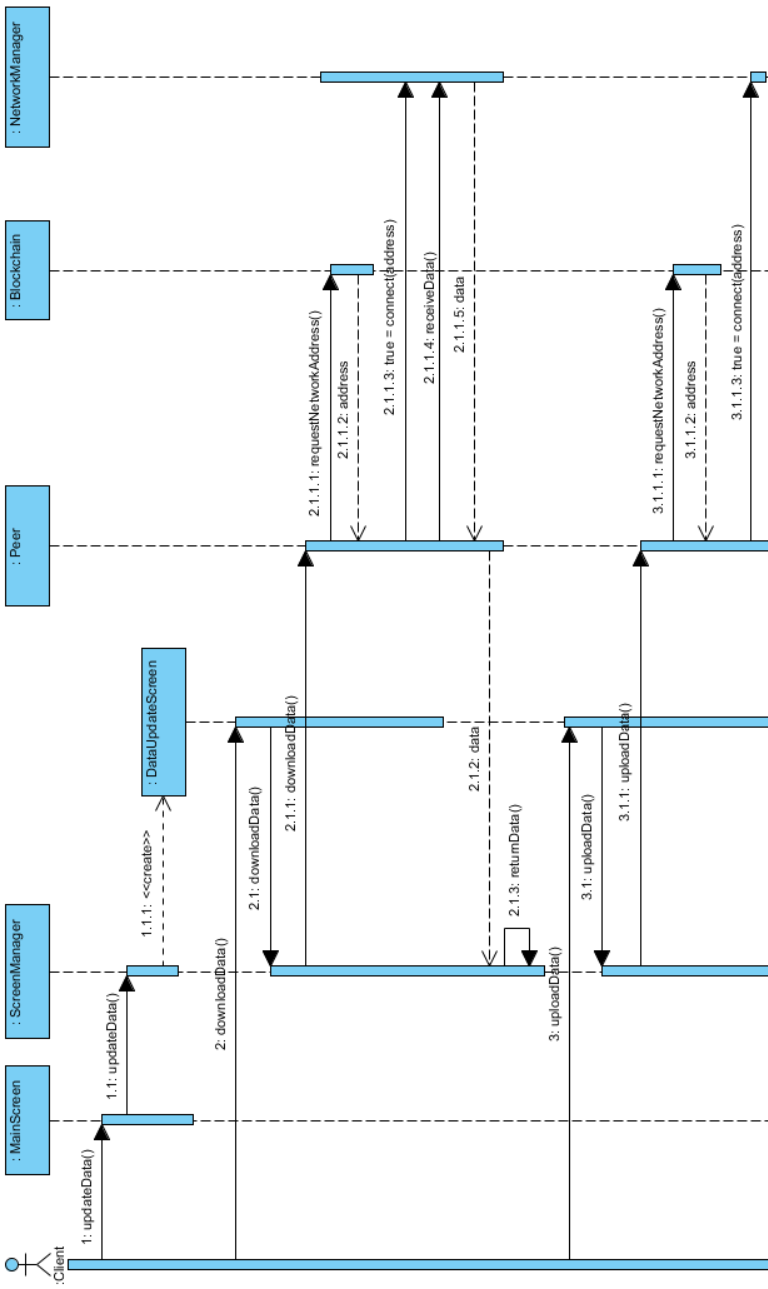
****

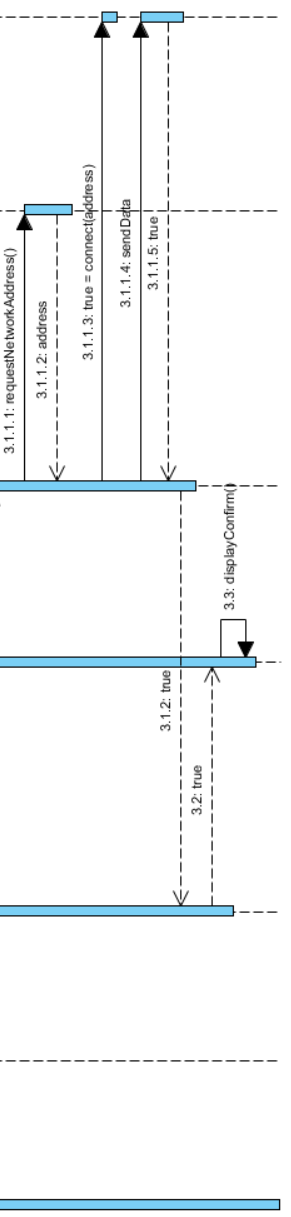
****

In the diagram, the actor is the client of the system. MainScreen is the boundary class which client interacts with. It sends the request of the client to the ScreenManager class which is control class responsible from the screen transitions. It creates new boundary classes which client can interact with. Peer object is the record of the client in the system and the SQL query is processed via its blockchain and network manager.

////////////////////////////////////////////////

The following diagram belongs to the upgrade data scenario.





Authentication is also required in this scenario, but it is not shown again since it is the same as the previous one. In the diagram, the actor is the client, main boundary class is MainScreen, and main control class is ScreenManager which also creates other boundary objects. The logic of the connections is the same as the query diagram except in this diagram, the system sends and receives data via network instead of querying.