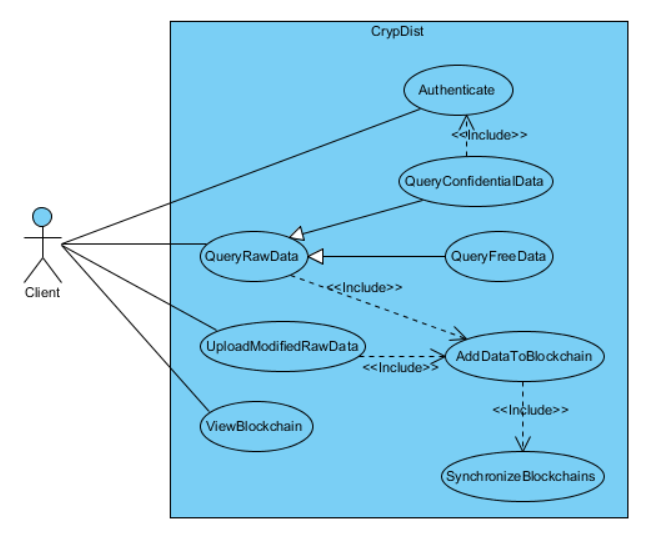
****

**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 displays 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 displays the data.

**Alternative flows:**

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

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

**Use case name:** UploadModifiedRawData

**Participating actor:** Client

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

**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 chooses the option to upload data.
* System displays the upload screen.
* 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 blockchain.

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

**Exit condition:** System returns to the main menu.

**Main flow of events:**

* Client chooses the option to view the blockchain.
* System opens the blockchain screen.
* System displays the blockchain.
* Client chooses the option to return to the main menu.
* System returns to the main menu.

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

**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.