

# DS Assignment 2

Battu Varshit  
201402029

1. Distributed Banking System using RMI
  - a. I created a class named "Bank" which holds the name of the person, account type, contact info, remaining balance.
  - b. I created another class named "Transaction" which holds the transaction\_id, account number, date and time of the transaction and the type of transaction(Credit/Debit).
  - c. I defined a HashMap for the accounts where key is the account number and values are the other details of the person.
    - i. Accounts = {'ACNo':[name, ac\_type, contact\_info, balance]}
  - d. I defined another HashMap for all the transactions where transaction id is the key and account number, date, time, type of transaction are the values.
    - i. Transactions = {'t\_id':[date, time, ac\_no, type]}
  - e. Now in a new terminal tab enter "rmiregistry".
  - f. In a new tab start the server by entering "javac Server.java". Then do "java Server".
  - g. As soon as the server starts the Accounts Dictionary gets populated.
    - i. The two default account numbers are 111111 are 222222.
  - h. Now start the client in another new tab by entering "javac Client.java". Then do "java Client".
    - i. To deposit to a particular account enter "deposit ac\_no amount".
    - j. To withdraw enter "withdraw ac\_no amount".
    - k. To check the balance enter "balance ac\_no".
    - l. To get all the transactions between two dates enter "transactionD ac\_no yyyy-mm-dd yyyy-mm-dd". The first date is the start date and the second date is the end date.
    - m. To get all the transactions of a particular user enter "transactionA ac\_no".
    - n. All the transactions are will be printed in a neat tabular format.

```

balance 111111
Balance is 10000.0
deposit 111111 19191
Balance Incremented 2 29191.0
withdraw 111111 1991
New balance is 3 27200.0
deposit 111111 110
Balance Incremented 4 27310.0
deposit 222222 1101
Balance Incremented 5 43966.43
withdraw 222222 10001
New balance is 6 33965.43
transactionA 222222
AC No      T_Date
222222     2017-09-10
222222     2017-09-10
transactionA 111111
AC No      T_Date
111111     2017-09-10
111111     2017-09-10
111111     2017-09-10
deposit 222222 10001
Balance Incremented 7 43966.43
transactionA 222222
AC No      T_Date
222222     2017-09-10
222222     2017-09-10
222222     2017-09-10
withdraw 222222 440000
No sufficient funds

```

i. Accounts = {ACNo:[name, ac\_type, contact\_info, balance]}  
 d. I defined another HashMap for all the transactions where transaction number, date, time, type of transaction are the values  
 i. Transactions = {t\_id:[date, time, ac\_no, type]}  
 e. Now in a new terminal tab enter "rmiregistry".  
 f. In a new tab start the server by entering "javac Server.java". Then "Server".  
 g. As soon as the server starts the Accounts Dictionary gets populated  
 i. The two default account numbers are 111111 and 222222.  
 h. Now start the client in another new tab by entering "javac Client.java" "java Client".  
 i. To deposit to a particular account enter "deposit ac\_no amount".  
 j. To withdraw enter "withdraw ac\_no amount".  
 k. To check the balance enter "balance ac\_no".  
 l. To get all the transactions between two dates enter "transactionD start\_date end\_date".  
 m. To get all the transactions of a particular user enter "transactionA ac\_no".  
 n. All the transactions are printed in a neat tabular format.  
 o. Server model using RMI  
 a. Enter "rmiregistry" in a new terminal tab.  
 b. In a new tab enter "javac server2.java". Then enter "java Server2".  
 c. In another new tab enter "javac client2.java". Then enter "java client2".  
 d. Used the Miller Rabin test which is inbuilt in Java BigInteger module.  
 i. Enter "miller number" to do a primality test.  
 e. Implemented the Palindrome test.  
 i. Enter "palindrome number" to check if it's a palindrome.  
 f. Implemented an iterative fibonacci number generator.  
 i. Enter "fibonacci n" to get the nth fibonacci number.  
 g. Implemented a function to change upper to lowercase and lower to upper case.  
 i. Enter "utol UPPERCASE" to convert to lowercase.  
 ii. Enter "ltou lowercase" to convert to uppercase.  
 h. Implemented the Diffie Hellman Key exchange protocol for the above functions.  
 i. A set of private and public keys is generated at both the client and server side.  
 j. The public key is shared between the server and client.  
 k. A secret key is generated at the client and server which is used to encrypt and decrypt the messages.

## 2. Theory

### 3. Client Server model using RMI

- Enter "rmiregistry" in a new terminal tab.
- In a new tab enter "javac server2.java". Then enter "java Server2".
- In another new tab enter "javac client2.java". Then enter "java client2".
- Used the Miller Rabin test which is inbuilt in Java BigInteger module.
  - Enter "miller number" to do a primality test.
- Implemented the Palindrome test.
  - Enter "palindrome number" to check if it's a palindrome.
- Implemented an iterative fibonacci number generator.
  - Enter "fibonacci n" to get the nth fibonacci number.
- Implemented a function to change upper to lowercase and lower to upper case.
  - Enter "utol UPPERCASE" to convert to lowercase.
  - Enter "ltou lowercase" to convert to uppercase.
- Implemented the Diffie Hellman Key exchange protocol for the above functions.
  - A set of private and public keys is generated at both the client and server side.
  - The public key is shared between the server and client.
  - A secret key is generated at the client and server which is used to encrypt and decrypt the messages.

- I. The server receives encrypted data from the client, decrypts it, gets the answer, encrypts it and sends it back to the client. The client decrypts the message and prints it to stdout.

```
Generating Keys....  
Sent Client Public Key to Server  
Received Public Key from Server  
Generated Secret Key  
miller 21  
composite  
miller 101  
prime  
palindrome 110011  
Palindrome  
palindrome 12345  
Not a Palindrome  
fibonacci 3  
2  
fibonacci 10  
55  
utol DISTRIBUTED  
distributed  
ltou distributed  
DISTRIBUTED
```