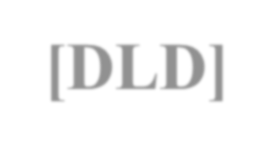
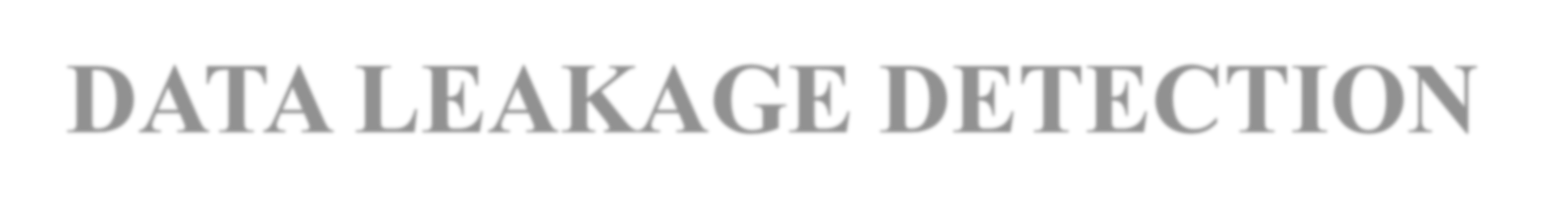
**DATA LEAKAGE DETECTION**



**[DLD]**

**BY:-**

**Subhashini Vanama**

**Solasa Manikanta Kumar**

**Pranita Salla**

**Under the guidance of**

**Ms. Roshani Ade**

# AGENDA

* PROBLEM DEFINITION
* PROBLEM SETUP AND MATHEMATICAL NOTATION
* SYSTEM ARCHITECTURE DESIGN
* SOFTWARE AND HARDWARE REQUIREMENT
* SCREEN SHOTS
* UML DIAGRAMS  ADVANTAGES
* FUTURE SCOPES
* CONCLUSION
* REFERENCES

# PROBLEM DEFINITION

* In the course of doing business, sometimes sensitive data must be handed over to supposedly trusted third parties.
* **Our goal** is to detect when the distributor's sensitive data has been leaked by agents, through probability calculation using number of download for a particular agent.

# PROBLEM SETUP AND NOTATION

**Mathematical model**

**Title:-**

DATA LEAKAGE DETECTION.

**Problem statement**: -

To build a application that helps in **Detecting the data** which has been leaked. Also it helps in finding **Guilty Agent** from the given set of agents which has leaked the data using

**Probability Distribution through number of Downloads.**

**Problem description:**

Let,

DLD is the system such that DLD={A,D,T,U,R,S,U\*,C,M,F}.

1.{**A**} is the Administrator who controls entire operation’s performed in the Software

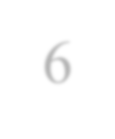
2.{**D**} is the Distributor who will send data T to different agents U.

1. **T** is the set of data object that are supplied to agents.

**T** can be of any type and size**, e.g**., they could be tuples in a relation, or relations in a database. **T**= {t1,t2,t3,...tn}

1. **U** is the set of Agents who will receive the data from the distributor A

**U**={u1,u2,u3,...un}

1. **R** is the record set of Data objects which is sent to agents **R**={t1,t3,t5..tm} **R is a Subset of T**
2. **S** is the record set of data objects which are leaked. **S**={t1,t3,t5..tm} **S is a Subset of T**
3. **U\*** is the set of all agents which may have leaked the data **U\***={u1,u3,...um} **U\* is a subset of U**
4. **C** is the set of conditions which will be given by the agents to the distributor.

**C**={cond1,cond2,cond3,...,condn}

1. **M** is set of data objects to be send in Sample Data Request algorithm **M**={m1,m2,m3,...,mn}

**ACTIVITY:**

**SAMPLE** is a function for a data allocation for any mi subset of records from T. The transition can be shown as:

**Ri =SAMPLE(T, mi)**

**EXPLICIT** is a function for a data allocation for which satisfies the condition.

## Ri =EXPLICIT(T,condi)

**SELECTAGENT** is the function used in EXPLICIT algorithm for finding the agent .

## SELECTAGENT(R1,R2….Rn)

**SELECTOBJECT** is the function used in SAMPLE algorithm for selecting the data Objects

## SELECTOBJECT(i,Ri)

**SIMPLE ENCRYPTO** is the function used to ENCRYPT the file to be sent to the Agent

**DATA STRUCTURES USED:**

**Array**: To store the no of data objects T ,No of agents U , record set R and to display the particular output.

**Execution of functions :**

The functions will be executed on a daily basis for number of times whenever distributor wants to send the data to the agent and vice versa using C and M.

**FUNCTIONAL DEPENDENCY DIAGRAM:**

The functional dependency of the system depends upon the conditions which are given by the agent and no of records which distributor decides to send to the agents.

Conditions

[

C

]

No. Of

Records

[

M

]

Record Set R

which is required

by the Distributor

Agents

[

U

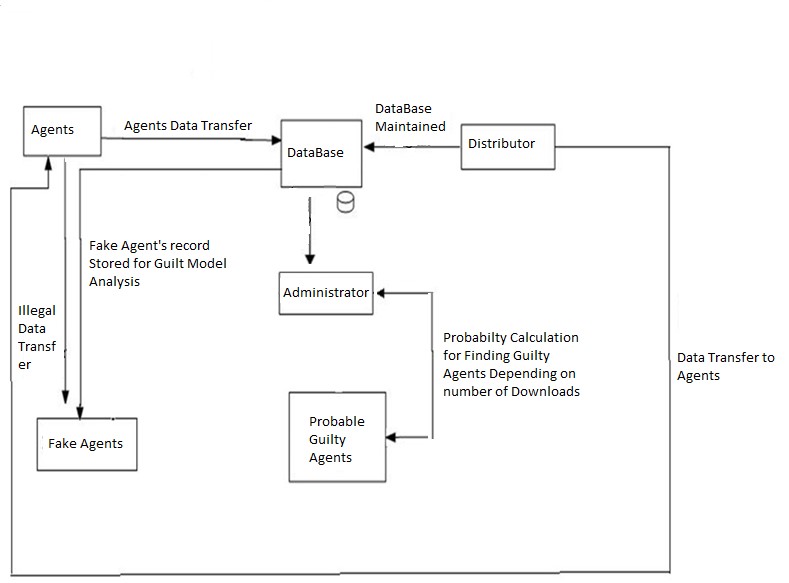
]

Distributor

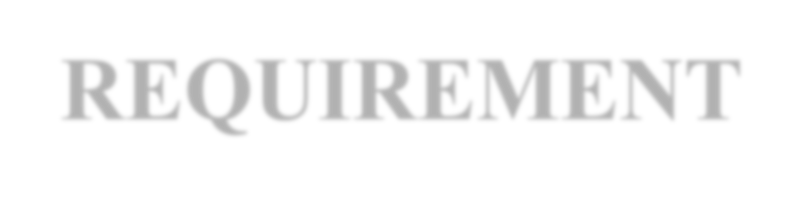
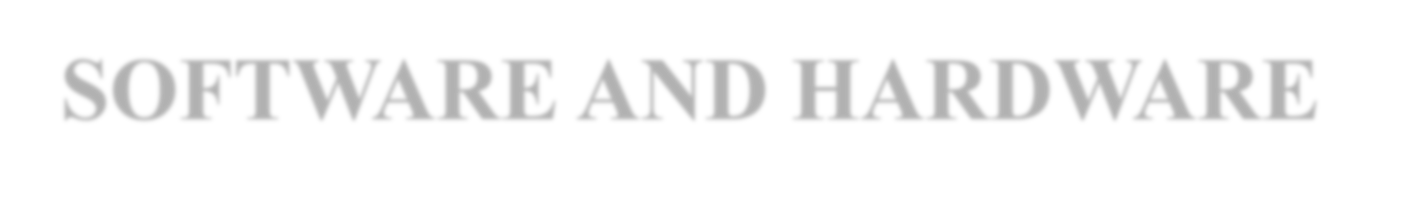
[

A

]

**SYSTEM ARCHITECTURE DIAGRAM**

# SOFTWARE AND HARDWARE



## REQUIREMENT

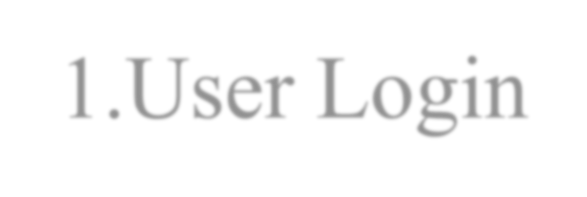
### Hardware Interfaces

* 2.4 GHZ, 80 GB HDD for installation.
* 512 MB memory.
* Users can use any PC based browser clients with IE 5.5 upwards.

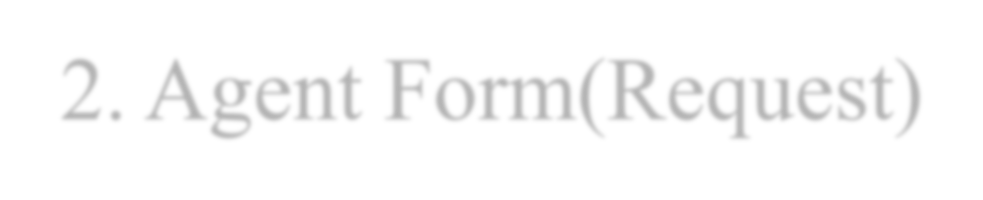
### Software Interfaces

* JDK 1.6
* Java Swing
* Net beans 6.5
* Socket programming
* Triple AES algorithm

# SCREEN SHOTS

1.User Login

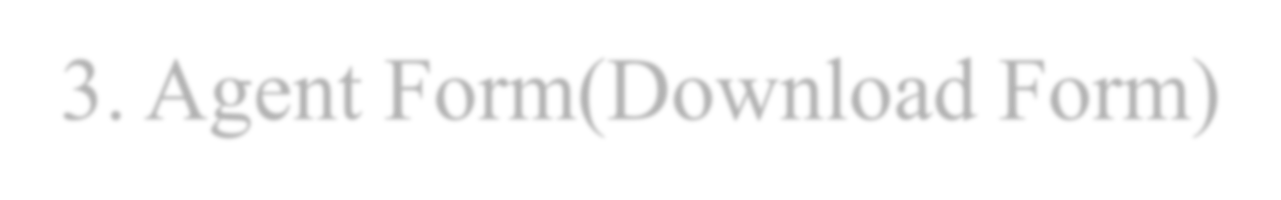
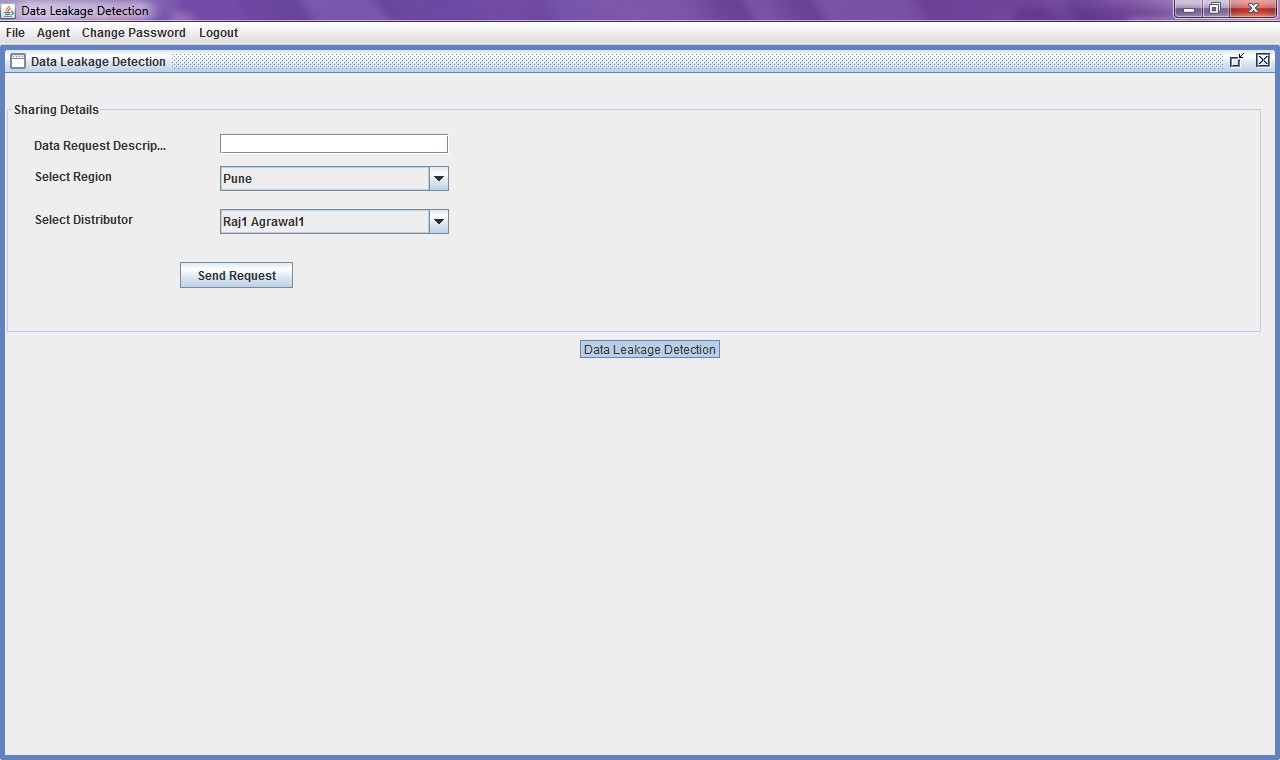




2

. Agent Form(Request

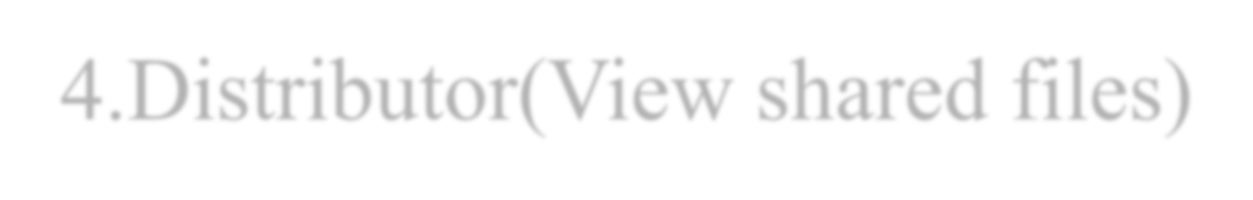
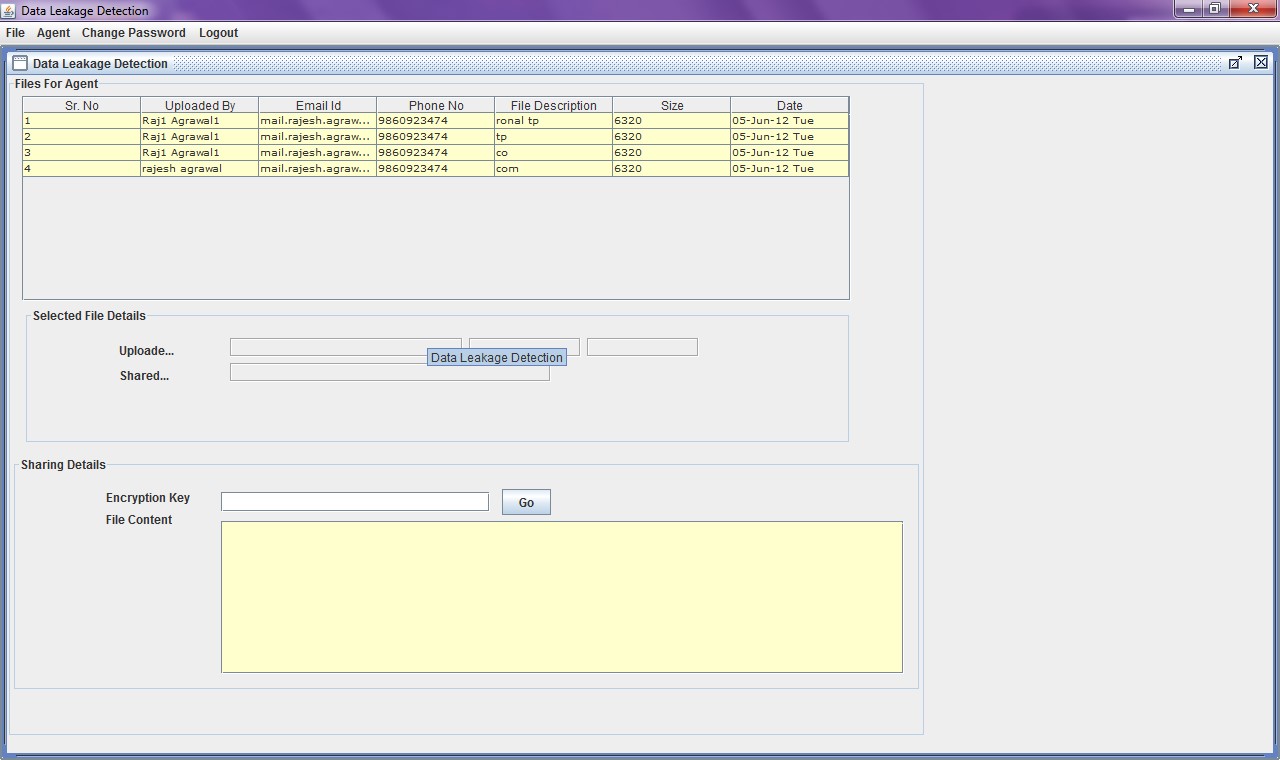
)



3

. Agent Form(Download Form

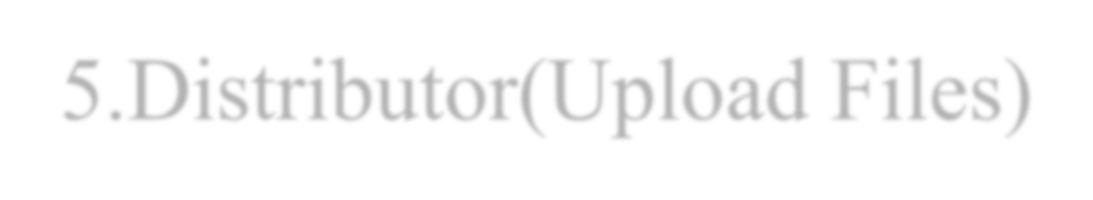
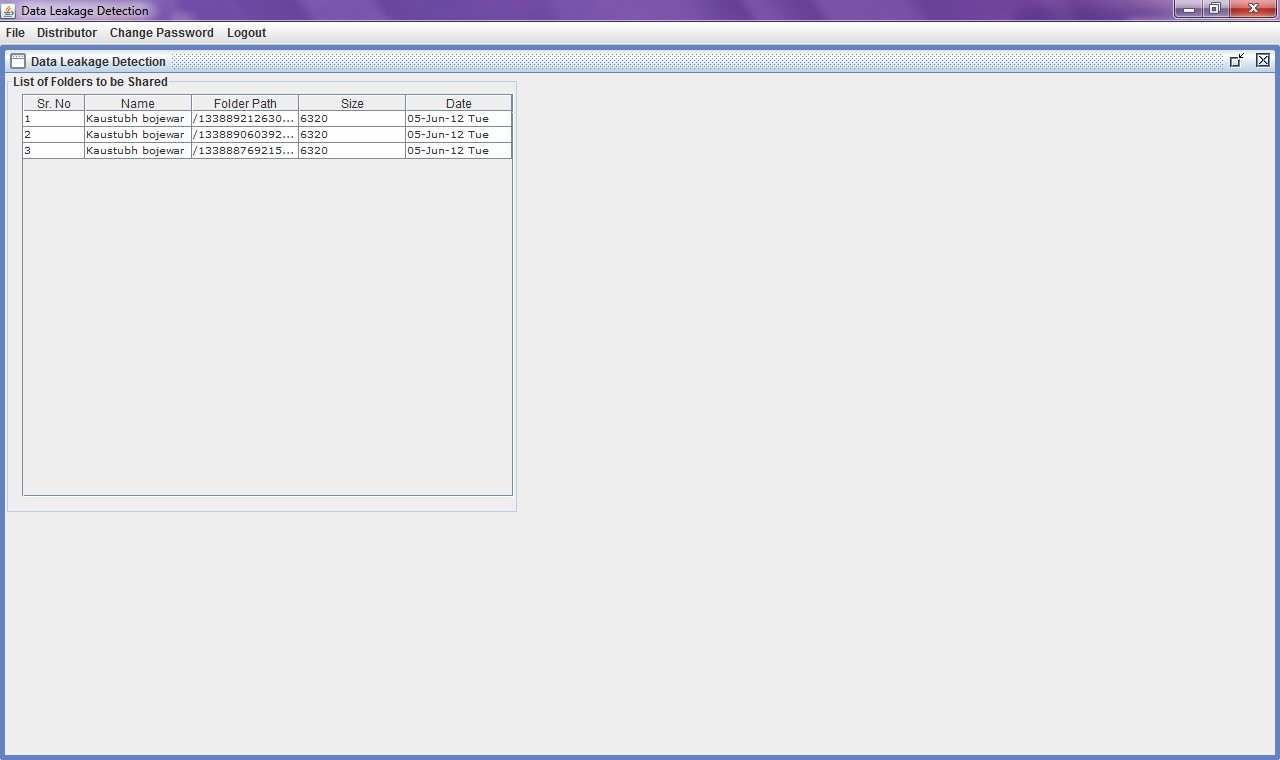
)



4

.Distributor(View shared files

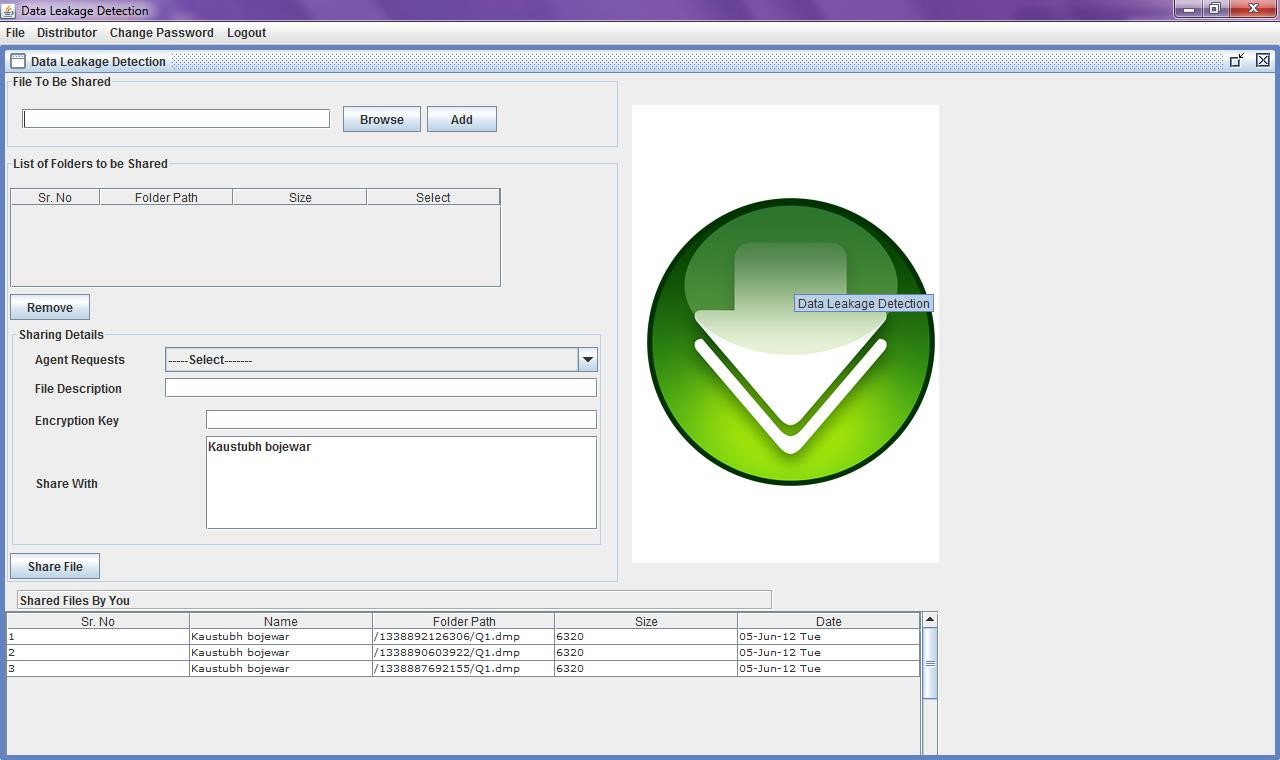
)



5

.Distributor(Upload Files

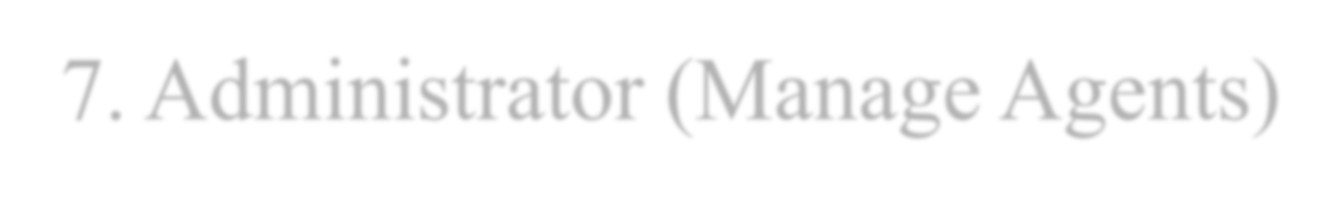
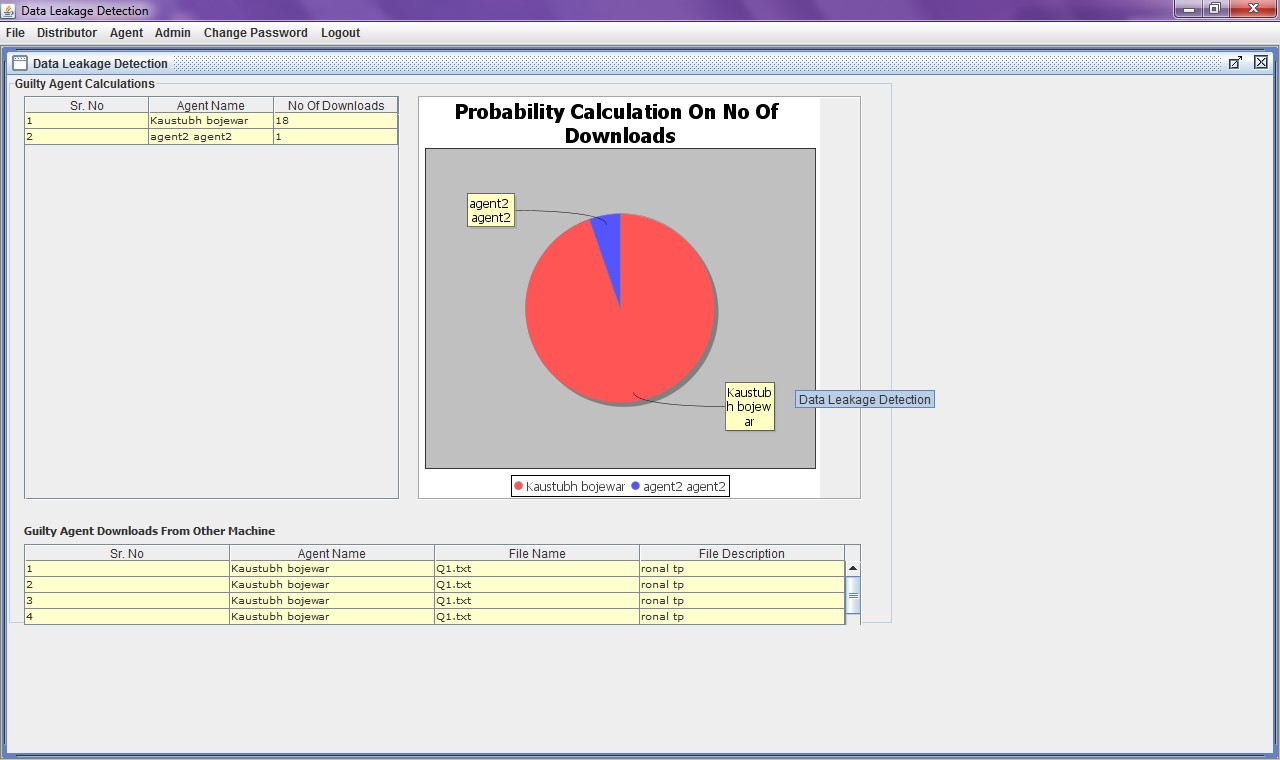
)



6

. Administrator ( Probability Calc

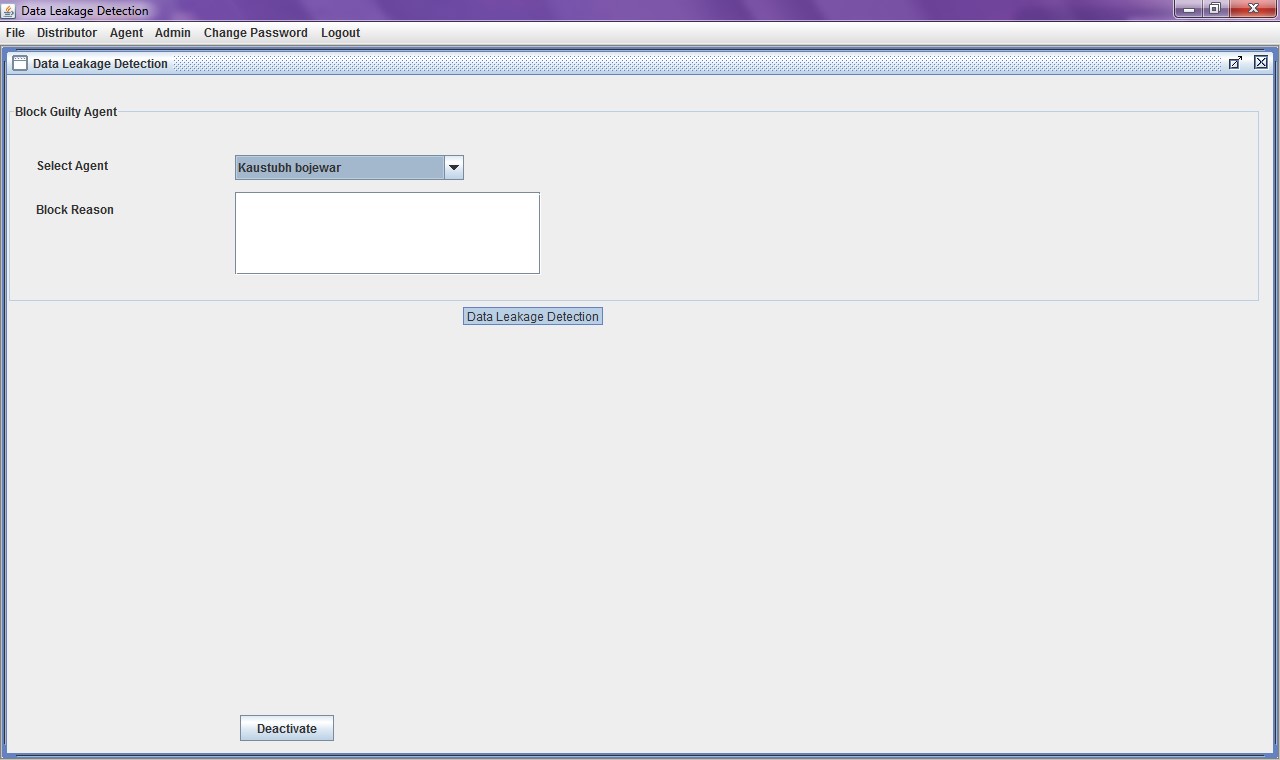
)



7

. Administrator (Manage Agents

)



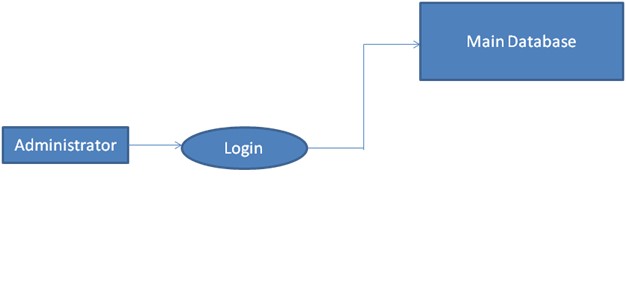
## UML DIAGRAMS

* Data Flow Diagram
* Use Case Diagram
* Class Diagram
* Sequence Diagram
* Activity Diagram

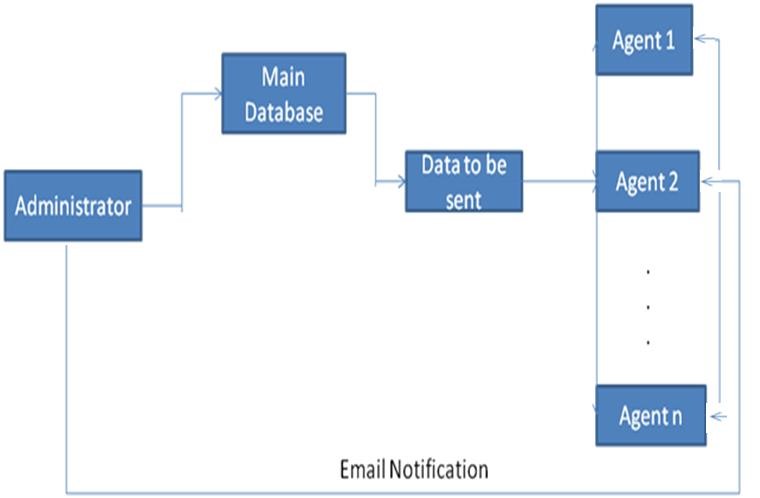
DATA LEAKAGE DETECTION

## 1. Data Flow Diagram

**Level 0**



**Level 1**



**Level 2**

Agent 2

Login

View Data

Send Data To

Anonymous

User

Agent 1

Login

View Data

.

Administrator

Email

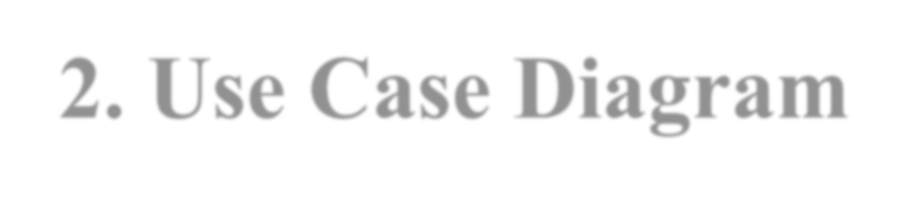
Notification

. .

Agent n

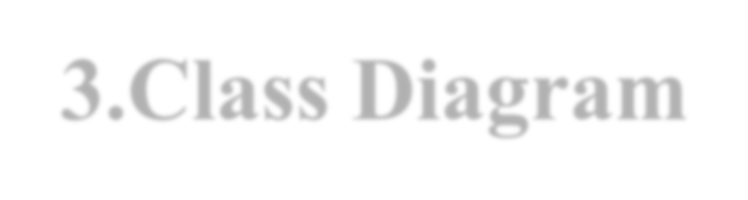
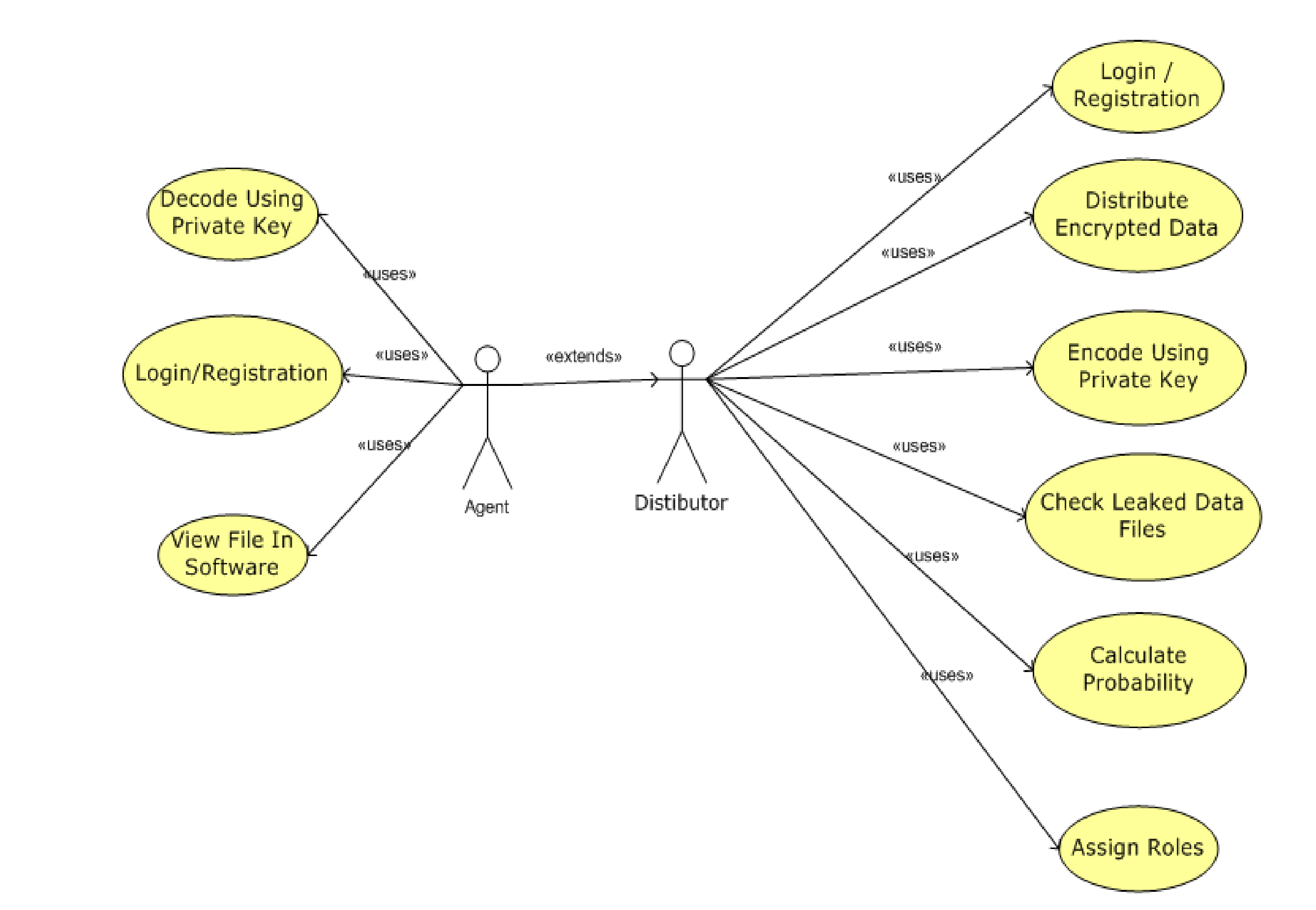
Login

View Data



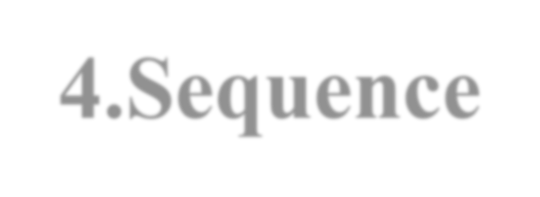
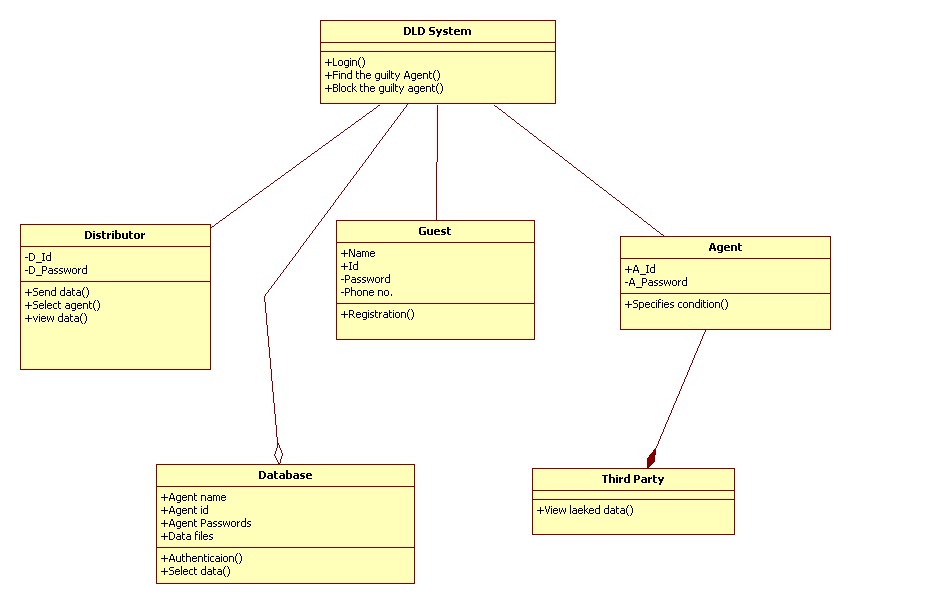
**2**

**. Use Case Diagram**



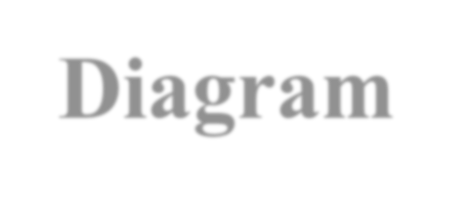
**3**

**.Class Diagram**

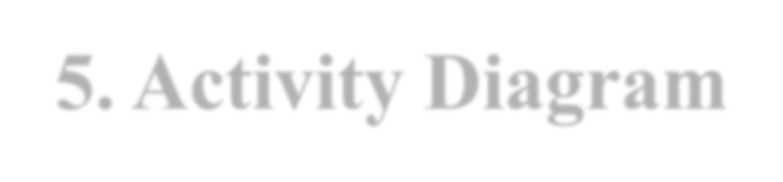
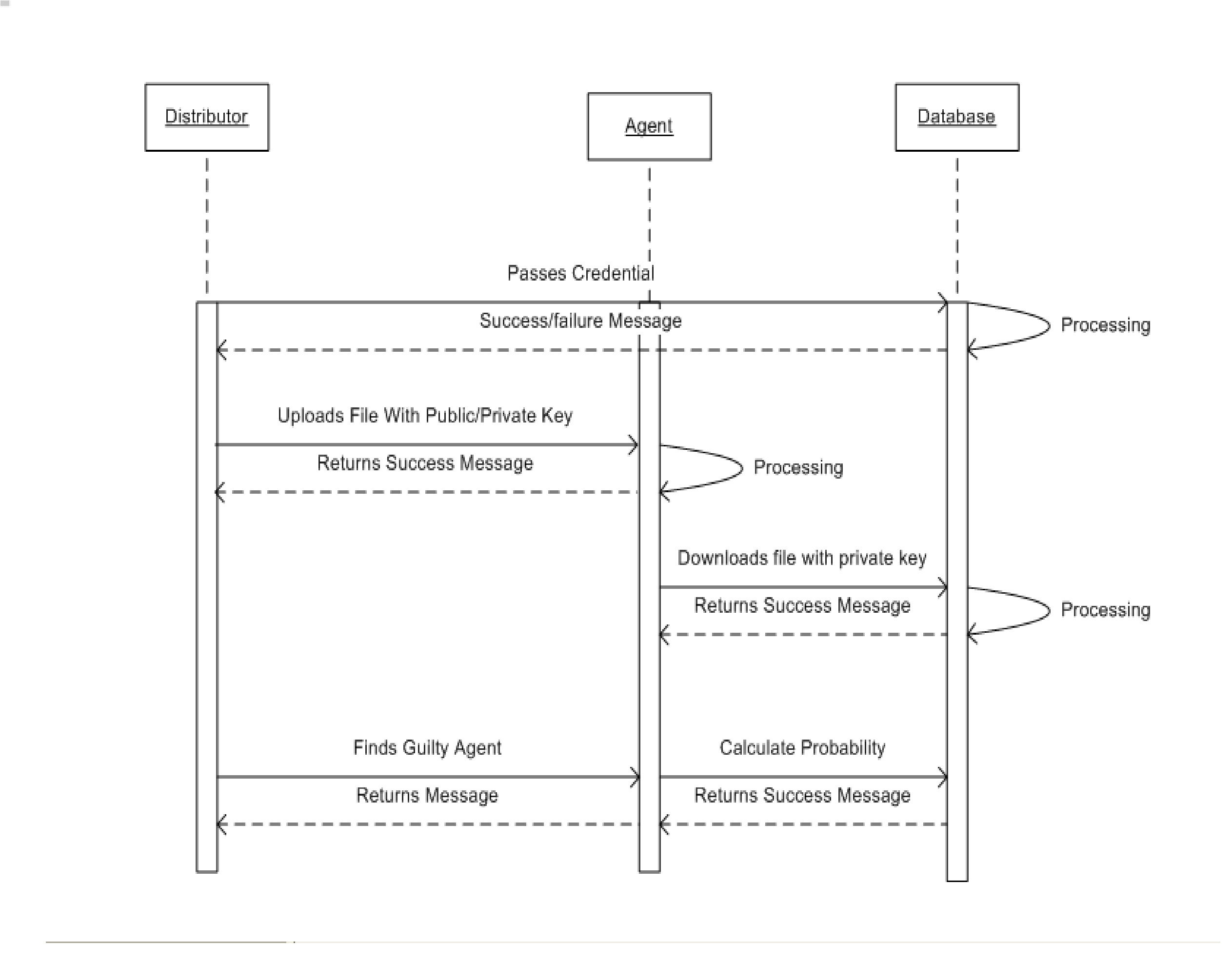


**4**

**.Sequence**

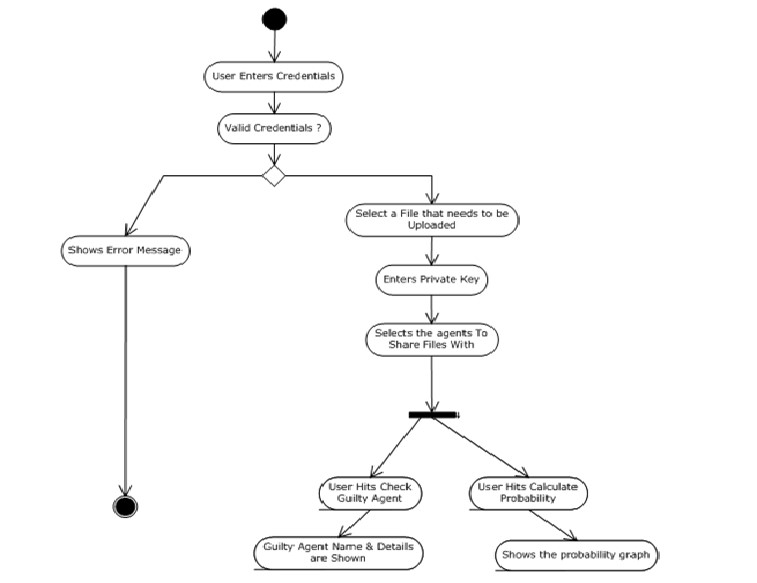


**Diagram**



**5**

**. Activity Diagram**

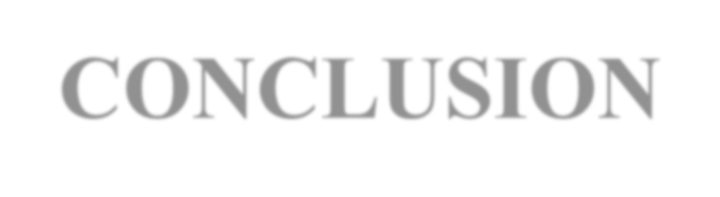


### ADVANTAGES

* This system includes the *data hiding* along with the provisional software with which only the data can be accessed.
* This system gives *privileged access to the administrator (data distributor) as well as the agents* registered by the distributors. Only registered agents can access the system. The user accounts can be activated as well as cancelled.
* The exported file will be accessed only by the system. The agent has given only the permission to access the software and view the data. If the data is leaked by the agent’ system the path and agent information will be sent to the distributor thereby the identity of the leaked user can be traced.

## FUTURE SCOPE

* Currently, we are dealing with only text files in this project but in future we will try to deal with all types of files.
* Recent research papers say that it is not possible to find the exact guilty agent who has leaked the data. Instead, we are finding out the probability of the agent being guilty or who has leaked the data through calculation of number of downloads.
* For more security, we will also provide a verification code on the agent’s mobile in future.



**CONCLUSION**

DATA LEAKAGE DETECTION

Login

-

Registration

Module

Upload file using

secret key

File sharing with

agents

Start HTTP

server

Check guilty

agent

Receives request

for download

Decrypts using

secret key

Java security

framework

Socket

programming

HTTP server

Java swing API

Technology

Agent’s module

Distributor’s module

Data Leakage

Detection System

### REFERENCES

* “Data Leakage Detection”Panagiotis Papadimitriou, Student Member, IEEE, and Hector Garcia-Molina, Member, IEEE  R. Agrawal and J. Kiernan, “Watermarking Relational Databases,” Proc. 28th Int’l Conf. Very Large Data Bases (VLDB ’02), VLDB Endowment, pp. 155-166, 2002.
* P. Bonatti, S.D.C. di Vimercati, and P. Samarati, “An Algebra for Composing Access Control Policies,” ACM Trans. Information and System Security, vol. 5, no. 1, pp. 1-35, 2002.
* P. Buneman, S. Khanna, and W.C. Tan, “Why and Where: A Characterization of Data Provenance,” Proc. Eighth Int’l

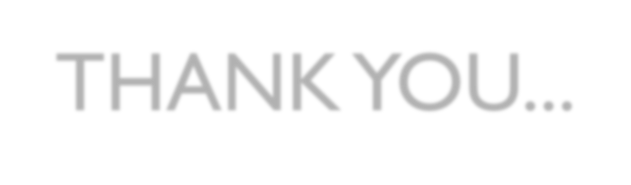
Conf. Database Theory (ICDT ’01), J.V. den Bussche and V. Vianu, eds., pp. 316-330, Jan. 2001

* P. Buneman and W.-C. Tan, “Provenance in Databases,” Proc. ACM SIGMOD, pp. 1171-1173, 2007.
* Y. Cui and J. Widom, “Lineage Tracing for General Data Warehouse Transformations,” The VLDB J., vol. 12, pp. 4158, 2003.
* F. Hartung and B. Girod, “Watermarking of Uncompressed and Compressed Video,” Signal Processing, vol. 66, no. 3, pp. 283-301,
* 1998.
* S. Jajodia, P. Samarati, M.L. Sapino, and V.S.

Subrahmanian,“Flexible Support for Multiple Access Control Policies,” ACM

* Trans. Database Systems, vol. 26, no. 2, pp. 214-260, 2001.  Y. Li, V. Swarup, and S. Jajodia, “Fingerprinting RelationalDatabases: Schemes and Specialties,” IEEE Trans. Dependable and
* Secure Computing, vol. 2, no. 1, pp. 34-45, Jan.-Mar. 2005.  B. Mungamuru and H. Garcia-Molina, “Privacy, Preservation and Performance: The 3 P’s of Distributed Data

Management,” technical report, Stanford Univ., 2008.

THANK YOU...