

**NATIONAL INSTITUTE OF TECHNOLOGY KARNATAKA  
SURATHKAL**



**DEPARTMENT OF  
COMPUTER SCIENCE AND ENGINEERING**

**CO362  
INFORMATION SECURITY  
COURSE PROJECT REPORT**

*K Rahul Reddy*  
171CO119

*Sushruth V*  
171CO148

# TABLE OF CONTENTS

<b>INTRODUCTION</b>	<b>1</b>
<b>IMPLEMENTATION</b>	<b>2</b>
APPLICATION DETAILS	2
DAC	3
MAC	4
RBAC	6
<b>RESULTS</b>	<b>8</b>
DAC	8
MAC	9
RBAC	11
<b>CONCLUSION AND REFERENCES</b>	<b>13</b>

## INTRODUCTION

Access Control is the set of tools that are used to control access to resources in a computing environment. Resources that need access control mechanisms to protect them can include personal information of users, employee credentials, or business secrets. Access Control is a basic and mandatory technology that needs to be deployed in web apps, websites, mobile applications, institutes, and organizations.

There can be two types of access control systems: physical or logical. Physical access control systems are deployed to keep bad actors away from campuses, buildings, rooms, and physical IT assets which house sensitive resources. Logical access control systems are deployed to monitor access to the resources from the computer network, trojan files, applications, and harmful data.

Logical access control can be achieved by assigning credentials or access cards to every user of the system. In such cases access control can be achieved by performing identity authentication and authorization of users and entities by scrutinizing their required login credentials that can include passwords, personal identification numbers (PINs), biometric scans, security tokens or other authentication factors.

These security controls work by identifying an individual or entity, verifying that the person or application is who or what it claims to be, and authorizing the access level and set of actions associated with the username or IP address.

With all these details, the aim of access control systems is to minimize the risk of unauthorized access to physical and logical systems. Access control is a fundamental component of security compliance programs that ensures security technology and access control policies are in place to protect confidential information, such as customer data. Most organizations have infrastructure and procedures that limit access to networks, computer systems, applications, files, and sensitive data, such as personally identifiable information and intellectual property.

There are three main types of access control mechanisms:

1. Discretionary access control (DAC): Owners or administrators of the resource set the policies defining who or what is authorized to access the resource
2. Mandatory access control (MAC): Access rights are regulated by a central authority based on multiple levels of security.
3. Role-based access control (RBAC): Restricts access to resources based on individuals or groups with defined roles rather than the identities of individual users.

Please find the code base for the project here: [https://github.com/vsushruth/IS\\_AccessControl](https://github.com/vsushruth/IS_AccessControl)

## IMPLEMENTATION

As per the requirements set for this project, we will be implementing all three access control mechanisms; DAC, MAC, and RBAC on the same application resulting in three different versions of the same application, each with one of the above-mentioned access control mechanisms running on it.

### APPLICATION DETAILS

The application we will be using to implement the access control mechanisms is an inventory management system for a chain of godowns that store various items for a front end retail shop. The system is built to manage multiple godowns that can order items from multiple suppliers, manage the orders, and change the list of items that are handled by the chain.

Our system is built with a SQL database in the backend, with a PHP service layer and a front end built by HTML 5, CSS 3, and Bootstrap 4. The system provides various functionalities to its users such as viewing the list of godowns managed by the chain, the list of suppliers that can be contacted for the purpose of restocking, details about the items being handled by each godown, the purchase history of each godown. There are also features that allow users to add a newly established godown, add new suppliers with whom a contract has been set up, put in a new purchase order with a particular supplier for a specific godown.

The SQL database has 6 tables that hold information and details of the business details. The tables are:

1. Employee Table - Contains details about employees of the chain, which includes employee ids, names, and addresses.
2. Godowns Table - High level table details about all the godowns that are part of the chain.
3. Godown Details Table - Low level details about every godown, the stock present in each one.
4. Suppliers Table - Details about all the suppliers that can be transacted with.
5. Purchases Table - High level details about the purchases, connecting the supplier and the godown between whom the transaction takes place.
6. Purchase Details Table - Low level detail about each transaction, specifying the details of the transactions, the id of purchase and the items being transacted.
7. Items Table - Has the list of all the items that the godown chain deals and trades with.

When we look at the functionalities provided by the system, we can come to the logical conclusion that not every employee of the organization should be able to access all the functionalities of the system. Only people of certain credentials, clearance levels, and roles should be able to handle more business sensitive functionalities and information. This calls for

the implementation of access control systems that restrict the functionality available and the data visible to various users based on their place in the system.

---

## DAC

Discretionary Access Control is one of the access control mechanisms that we have implemented. DAC can be defined as a mechanism restricting access to objects based on the identity of subjects. Here the discretionary part suggests that any user with the permission that allows them to share their permissions with other users is able to do so on their own discretion. The special permission, if present in any system, has to be set appropriately by the policies that govern the system.

In our application, we have implemented DAC using an *Access Matrix*. An access matrix is a control model in which resources and functionalities that need access control are classified as *Objects* and the set of users known as *Subjects*. There exists a set of *Rights* that can be assigned in relation to each Object and Subject pair.

The access matrix that has been implemented in our application has the following parts:

- Subjects: Employees of the godown chain
- Objects: The information and functionalities provided by the application
- Rights:
  - Read Access: Any subject with this right is able to read the data in the object
  - Write Access: Any subject with this right is able to add new data into the object or modify what is already present in it.
  - Owner: This is the special right that allows subjects who possess this right to be able to grant read and write access to other subjects

The access matrix is stored in the SQL database as a standalone table. The columns in the table correspond to the parts described above. The details of the subjects are stored as the Employee ID that is used to login to the application. The objects are of five kinds, the first three are the objects, 'item', 'godowns' and 'suppliers', which allow users to access the high-level details of the items being traded, the details of all godowns in the chain and the details of all the suppliers in connection with the godown chain. The fourth type of object is the set of single godowns, through which the users can access the stock held in the godown. The final type is the godown-purchase kind, that specifies the rights a user has to see the purchase history of a certain godown and be able to purchase more stock for a certain godown.

+ Options				
Employee_ID	Objects_ID	Read_Access	Write_Access	Owner
1	items	1	1	1
1	2p	1	1	1
1	2	1	1	1
1	godowns	1	0	0
5	suppliers	1	0	0
5	items	1	0	0

Access Matrix for DAC Mechanism

Any user with the Read Access right to a certain object is able to see the data held in that object in terms of a table in the front-end of the application. If the read access is not enabled then trying to access an object will be met by an error message in the front-end. The same situation is designed for Write Access rights; any user with the right is able to fill forms to modify entries in the object or add new tuples into the object. If a user does not have the right to add or modify will see an error message in the place of the form

Coming to the Owner right, any user with this right is able to see the details of all the users who have a relationship with the object to which they hold the Owner right. This right over an object also provides a user with the extra functionality of being able to add or modify the rights of other users with respect to the object.

## MAC

Mandatory access control is an access control mechanism in which the creator or owner of a resource is not allowed to control the access rights of other users to the resource. The access to all the resources in the system is assigned by the system administrator and is enforced by the system based on certain criteria assigned to every user in terms of their credentials.

Our implementation of the MAC mechanism is via the concept of security levels, ranging from 1 to 4, that is assigned to all the subjects and objects that are part of the system. Here, the definition and meaning of the terms 'subjects' and 'objects' are consistent with that in the case of the DAC mechanism. The security level 1 is one with the least clearance while security level 4 has the most clearance.

The implementation of these security levels is implemented via a modified version of the access matrix defined in the case of DAC. The access matrix in this context only contains the various objects and the associated security level of the objects. An object can only hold one

security at a time, multiple values are not allowed. On the other hand, the security levels of the subjects, who in this situation are the employees of the chain, will be included in the Employees Details table along with their ID.

+ Options	
Objects_ID	Clearance
godowns	1
1p	3
purchases	3
items	3
1	4

Access Matrix for MAC Mechanism

+ Options									
		Employee_ID	Employee_Name	Supervisor_ID	Home_Number	Street	Pincode	Password	Clearance
<input type="checkbox"/>	Edit	1	Rahul	NULL	asd	asd	131	7c4a8d09ca3762af61e59520943dc26494f8941b	4
<input type="checkbox"/>	Edit	2	K Rahul Reddy	1	123	Srinivas Nagar, Nitk	575025	7b52009b64fd0a2a49e6d8a939753077792b0554	0
<input type="checkbox"/>	Edit	3	Sush	2	12	abc	564789	7b52009b64fd0a2a49e6d8a939753077792b0554	0
<input type="checkbox"/>	Edit	4	abc	1	123	asd	123456	7ab515d12bd2c1431745511ac4ee13fed15ab578	0
<input type="checkbox"/>	Edit	5	rahul	1	1	1	1	7c4a8d09ca3762af61e59520943dc26494f8941b	2
<input type="checkbox"/>	Edit	6	K Rahul Reddy	1	12	Srinivas Nagar	575025	7c4a8d09ca3762af61e59520943dc26494f8941b	0

Employees Details Table with their Credentials and Security Levels

The concept of read and write access rights here is not as straight forward as the DAC system. These rights are derived via the comparison between the security levels of the user and the objects they are trying to access. The rules for the assignment is defined as below:

- Read Access - A user with a security level greater than or equal to that of an object has read access to that object
- Write Access - A user with a security level strictly greater than the security level of the object has write access to it.

The criteria set for read and write accesses are not defined by the concept of mandatory access control or that of security level. These criteria are set by the policy chosen by the application or the organization. Thus, in this particular application of ours, we have chosen to enact the above-mentioned methods to grant read and write access to the users

## RBAC

Role-based access control is a policy-neutral access-control mechanism that works with roles and privileges. The concept of roles is that every user who has an account in the system has been assigned a role. The role assigned to a user determines the privileges that the person has with respect to the objects present in the system. RBAC is the most commonly used access control mechanism in the industry.

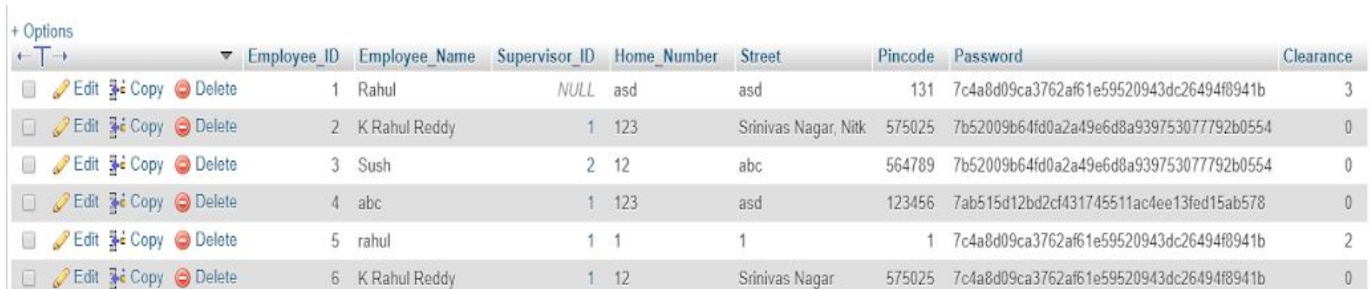
Our implementation of the RBAC system is done, again, via a variation of the access control matrix. The matrix is stored as a table in the database similar to the DAC mechanisms. But unlike DAC, the relation between an object and a subject is not unique to each employee but is unique to a group of employees who take on the same role in the godown chain.

The specific roles assigned to each of the employees is stored once again in the Employee Details. The roles that an employee can take are described in terms of associate levels from 1 to 3. A concept that differentiates MAC with this mechanism is that a user can take on multiple roles in the organization, unlike the strict singularity of MAC. But when a user in the RBAC system logs into the application, he can log in as a user with only one single role at a time, and when the person needs to access functionality assigned to another role of his, he needs to switch his credentials to the other role and login to the application once more.



Roles	Objects_ID	Read_Access	Write_Access
3	items	1	0
3	godowns	1	1
3	1	1	1
3	1p	1	1

Access Matrix for RBAC Mechanism



Employee_ID	Employee_Name	Supervisor_ID	Home_Number	Street	Pincode	Password	Clearance
1	Rahul	NULL	asd	asd	131	7c4a8d09ca3762af61e59520943dc26494f8941b	3
2	K Rahul Reddy	1	123	Srinivas Nagar, Nitk	575025	7b52009b64fd0a2a49e6d8a939753077792b0554	0
3	Sush	2	12	abc	564789	7b52009b64fd0a2a49e6d8a939753077792b0554	0
4	abc	1	123	asd	123456	7ab515d12bd2cf431745511ac4ee13fed15ab578	0
5	rahul	1	1	1	1	7c4a8d09ca3762af61e59520943dc26494f8941b	2
6	K Rahul Reddy	1	12	Srinivas Nagar	575025	7c4a8d09ca3762af61e59520943dc26494f8941b	0

Employees Details Table with their Credentials and Employment Role Levels



The access matrix contains the following column:

- Role
- Object ID
- Read Access
- Write Access

Another difference between this mechanism and that of MAC is that there is no rigid policy that needs to be set up by the administrator about how read and write access are calculated.

## RESULTS

### DAC

Supermarket

Suppliers Purchase Godowns Items LOGOUT

WelcomeRahul

Your Access Rights

Object ID	Read Access	Write Access	Owner
Items	1	1	1
Sp	1	1	1
2	1	1	1
godowns	1	0	0

Objects you Own

Object ID	Read Access	Write Access	Owner
Items	1	1	1
Sp	1	1	1
2	1	1	1

Modify Access Rights :

Object ID  
Items

Employee ID  
1

Read Access  
1

Write Access  
0

Add

In the images above and below, we see that User 1 is provided with read and write access to object Items. This means that he has access to view the list of items that the godown is trading in and add more items to that list. This user also is the owner of three objects and thus is able to provide access to those three objects to other users.



All Items			
Item ID	Item Name	Item Units	Item Unit Price
1	Good dry	piece	6
2	Good dry cashew	piece	10
3	Good dry ginkan	piece	12
4	Tomato	kg	20
5	Potato	kg	15
6	Bread	Loaf	30
7	Mango	kg	50
8	Banana	dozen	35
9	Cashew	kg	1000
10	Oranges	kg	50
11	Apples	kg	100
12	oil	piece	15
13	MGM	piece	5
14	Corral	kg	10
15	Capsicum	kg	50
16	Birinjil	kg	20
17	Bitter guard	kg	25

You don't have the clearance to Add Items

Add Items :

[Back](#)

Since the current user does not have write access for the 'items' object, the user is not able see the form which allows users to add new items into the list

## MAC

### User 1:

Welcome Rahul	
Your Clearance Level : 4	
Objects you are cleared to access	
Object ID	Clearance Level
godowns	Level 1
1	Level 4
lp	Level 3
purchases	Level 3
items	Level 3

When a user logs in he is able to see their security level and the objects that they are able to access.

Item ID	Item Name	Item Units	Item Unit Price
1	Good dry	piece	5
2	Good dry cashew	piece	10
3	Good dry griben	piece	12
4	Tomato	kg	20
5	Potato	kg	15
6	Bread	loaf	30
7	Mango	kg	50
8	Banana	dozen	35
9	Cashew	kg	1000
10	Oranges	kg	50
11	Apples	kg	100
12	oil	piece	15
13	MMM	piece	5
14	Corrot	kg	10
15	Cogolcum	kg	50
16	Brigip	kg	20
17	Biter guard	kg	25

Add Items :

Name

Units of Measurement

Price per unit

Since this user has a security level of 4 and the items object has a security level of 3, he is able to both read from the object as well as write to it.

## User 5:

# Welcome rahul

## Your Clearance Level : 2

### Objects you are cleared to access

Object ID	Clearance Level
godowns	Level 1

Here we see that User 5 has a security level of 2 and thus is only able to access the 'godowns' object.

## All Items

You don't have access to this information

Contact the administrator for further information

You don't have the clearance to Add Items

Add Items :

[Back](#)

Logically, since this User only has a security level of 2, he is not able to access the 'items' object which has a security level of 3.

## RBAC

User 1:

## Welcome Rahul

### Your Role is: Level 3 Employee

#### Role-Based Access Matrix

Role	Object ID	Read Access	Write Access
Level 3 Employee	items	1	0
Level 3 Employee	godowns	1	1
Level 3 Employee	1	1	1
Level 3 Employee	1p	1	1

Similar to the MAC system, users can see their roles on the home page as they login. They can also see the access matrix for their role.

All Items

Item ID	Item Name	Item Units	Item Unit Price
1	Good day	piece	8
2	Good day cashew	piece	10
3	Good day golden	piece	12
4	Tomato	kg	20
5	Potato	kg	15
6	Bread	Loaf	30
7	Mango	kg	50
8	Banana	dozen	35
9	Cashew	kg	1000
10	Oranges	kg	50
11	Apples	kg	100
12	asd	piece	15
13	MSM	piece	5
14	Carrot	kg	10
15	Capsicum	kg	50
16	Brinjal	kg	20
17	Bitter guard	kg	25

You don't have the clearance to Add Items

Add Items :

Since the current user is a level 3 employee, he has the access to read the 'items' object but does not have the access to write to it. This is reflected in the front end of the application.

User 5:

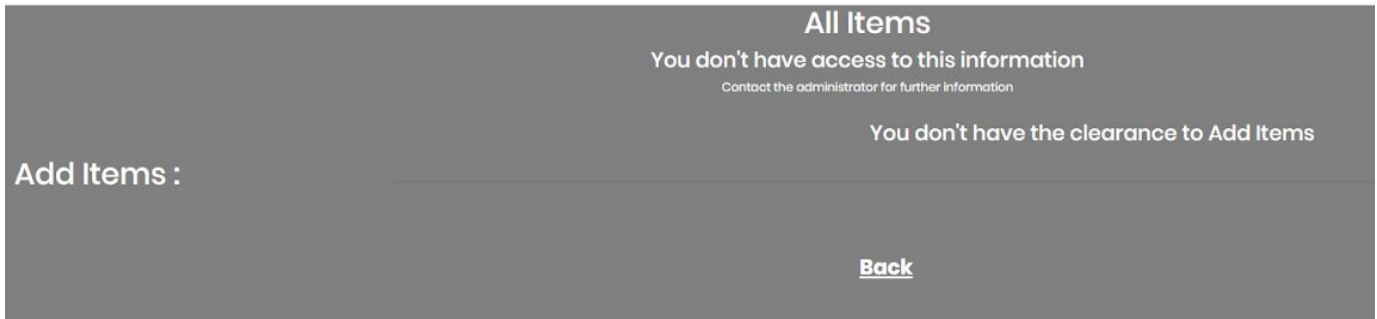
Welcome rahul

Your Role is: Level 2 Employee

Role-Based Access Matrix

Role	Object ID	Read Access	Write Access
Level 3 Employee	items	1	0
Level 3 Employee	godowns	1	1
Level 3 Employee	1	1	1
Level 3 Employee	lp	1	1

This user has the role of a Level 2 employee



Since a Level 2 employee does not have access to read or write into the 'items' object, they are not able to see any data

---

## CONCLUSION AND REFERENCES

Please find the code base for the project here: [https://github.com/vsushruth/IS\\_AccessControl](https://github.com/vsushruth/IS_AccessControl)

Anywhere and anything that has resources that need to be monitored and access to which needs critical control needs to implement access control systems. The specific details of the access control system that needs to be deployed for a particular organization or an application can only be articulated by the policies and conditions set up by the same organization or application. In today's corporations that treat data as the new oil, secure and foolproof access control mechanisms need to be the backbone of their security systems. Granting access to sensitive information or resources to a bad actor can lead to consequences that may turn out catastrophic for the organization. Thus, the case for designing new and improved access control mechanisms and implementing them in clever ways is the need of the hour

### References

1. <https://searchsecurity.techtarget.com/definition/access-control>
2. [https://en.wikipedia.org/wiki/Discretionary\\_access\\_control](https://en.wikipedia.org/wiki/Discretionary_access_control)
3. [https://en.wikipedia.org/wiki/Access\\_Control\\_Matrix](https://en.wikipedia.org/wiki/Access_Control_Matrix)
4. <https://searchsecurity.techtarget.com/definition/mandatory-access-control-MAC>
5. [https://en.wikipedia.org/wiki/Role-based\\_access\\_control](https://en.wikipedia.org/wiki/Role-based_access_control)
6. [https://en.wikipedia.org/wiki/Role-based\\_access\\_control](https://en.wikipedia.org/wiki/Role-based_access_control)