Enterprise Resource Planning for College Administration

[1] Parth Amalgaonkar, [2] Supriya Jadhav

[1] Computer Engineering, Savitribai Phule Pune University, G.H. Raisoni Institute of Engineering and Technology
Pune, Maharashtra, India

[2] Computer Engineering, Savitribai Phule Pune University, G.H. Raisoni Institute of Engineering and Technology
Pune, Maharashtra, India

Abstract - Management of a large institute is a critical task in today's rapidly changing environment. ERP system has become an essential tool for managing such large organisations. Paperwork of organizations is hugely decentralized and scattered. Our project aims to completely replace the traditional system which is heavily paper-based, by our automated software and reduce this inefficiency. It will integrate all the management systems required for college administration along with providing data integrity, security and multiple levels of abstraction. The system also proposes an intelligent querying approach by adding an abstraction data layer to reduce the extensive load on server in such systems. Once deployed, the system will effectively reduce the manual work by automating and integrating all the processes involved in college management, thus improving the overall management and thereby increasing the quality and efficiency of the organisation.

Keywords - *ERP, integration, information, intelligent-querying, management, security.*

1. Introduction

An ERP is a system which acts as a complete software solution for managing all kinds of organizations like industries, institutes, enterprises, et al. It is fast replacing all traditional business processes by providing an optimal, automated platform with various tools for managing specific work processes. It integrates all data and processes in a single software package, promoting centralization, integrity, efficiency, security and simplicity in the workflow, which is generally absent in traditional paper-based systems. The proposed system is an ERP designed for college administration, aimed at integrating all institute processes and sub-processes into a single unified system. It will automate all the processes in order to reduce paperwork and thus increase productivity of the organization by speeding up all the processes. The college will function better if the faculties, administrators and students are freed up of the manual paper work required for managing the institute, which gives them time to focus on their actual functions.

The system is designed to manage all the processes, right from admission to storing, modifying and analyzing student data. Extensive information is available at your fingertips through this system. Viewing student data, managing student information, for examinations, courses management, scheduling exam, result and related issues are made simple and easy. There are intelligent capabilities, by using the EIRQ algorithm, to aid in finding student information and working on student records. This can make the system easier to navigate and to use maximizing the effectiveness of time and other resources. ERP allows the keeping of personnel data in a form that can be easily accessed and analyzed in a consistent way.

Each of the modules in ERP covers many other aspects. The system records basic personal information, admission information, and course information regarding the student. The cutting edge systems provide the ability to read the database and enter relevant data to applicable database fields, notify student and provide results.

ERP functions involve:

- Manage new admissions
- Manage courses
- Student Basic Information
- Manage faculty
- Exam scheduling
- Result management
- Declaration global notes

- Manage attendance
- Manage batches
- Report generation
- Library management
- Event management
- TPO module

In this system, every user has a Login ID and Password. Also, all users have different permission rights to access the applications. These rights are Dynamic and can be changed. The view of the application changes according to the authority and permission rights granted to the user by the main administrator.

2. Literature Survey

2.1 Existing System

Management of educational institutions has always been a reason of deep concern for people associated with it. In existing system there are limited options for the management as most of the administrative tasks have to be done manually. In the current IT world, organizations are striving to make administration process digitized.

So far, not many institutes have succeeded in bringing together their entire administration procedure. Furthermore, the task of integrating so many different modules of a firm makes complete digitization extremely difficult to achieve. The present system has certain major disadvantages. A few to be listed can be excessive paperwork, time consuming process flow, laborious work environment for employees, difficulty to access historical data and all these problems lead to inefficient working of government sector causing dissatisfaction in the general public. Apart from the above stated problems there is lack of transparency in the existing system. This being one of the major drawbacks in the system needs special attention.

2.2 Proposed System

The proposed system provides an integrated, user configurable and dynamic software solution, which will help institute to get the wide range, detailed and summarized information of administrative and academic nature, in different forms required at different level of the organizational hierarchy such as students, parents, departments, accounts and other interested linked organizations. Along with bringing together everything, it is necessary to ensure that the system's integrity is maintained. Thus the system will provide various levels of user authorization through data abstraction and keep the repository secure by encrypting essential data. We also propose a smart way of optimized querying to the database by adding an extra aggregation and distribution layer that will first scan and reduce different queries before firing them on database which will optimize the system's overall performance. Also various data mining techniques could be applied to the database which could generate additional information that may help influence different modules and help them in taking management decisions easily. Thus, ERP helps in improving the overall management and thereby increasing the quality and efficiency of the organization and hence leading to better outputs than earlier. It allows the management to make correct decisions more quickly and allows other users to focus on their own jobs.

3. Architecture

Our system works in scalable three tier architecture as shown in fig. The system is logically divided in three parts viz.

- Client tier
- Business tier.
- Database or EIS tier

Client tier will consist of all the logic for displaying the data on the client side irrespective of changes made in the logical or database layer. It will be made up of HTML, DHTML and JavaScript along with components of JSF embedded in JSP for more controlled access and better client side event management. Business or logical layer will contain all the logic for implementation of the whole system. The actual implementation of algorithm and control flow of the system is mapped in this layer. This layer will make use of advanced technologies in Java like Servlets, Hibernates and XML for synchronized task handling on event generation from clients by providing appropriate response by querying the database. As this layer is isolated from representation of data and its access, any changes made in implementation logic will not force changes to be made in the other two layers. This reduces the complexity and thus provides ease of maintainability.

The database tier will hold a centralized database in MySQL for all the information of the system. Data will be stored in normalized form so that manipulations on data can be efficient. This system will have the following advantages over the conventional system architecture:

1. Managing data is independent from the physical storage.

2. Migration to new graphical environments is easier and faster.

3. It is possible to make changes on the presentation level without affecting the other two layers i.e. business or data access layer.

4. As each tier is independent it is possible for simultaneous development tasks to be executed leading to faster development process.

5. Since the client doesn't have direct access to the database or business logic system is more secure and reliable.

6. Failure of a layer doesn't affect other layer thus making it easy for maintainability

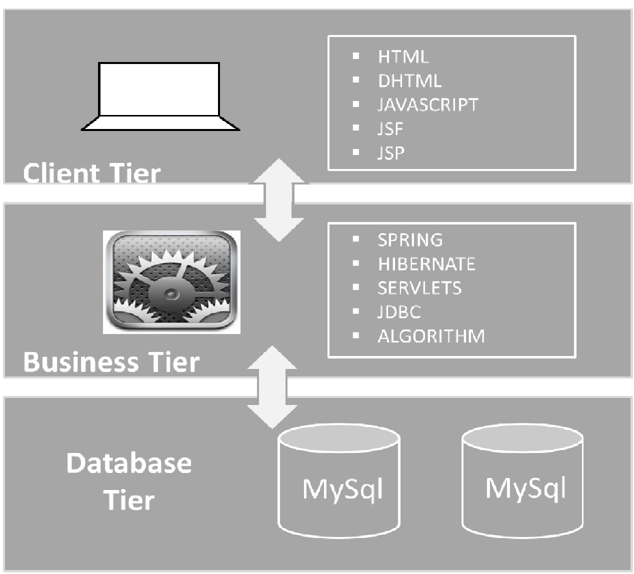

Fig. 1 System Architecture

4. Algorithm

4.1 EIRQ Algorithm

As shown in Fig. 2 there are 3 entities in the system model:

- Users
- ADL
- Database Server

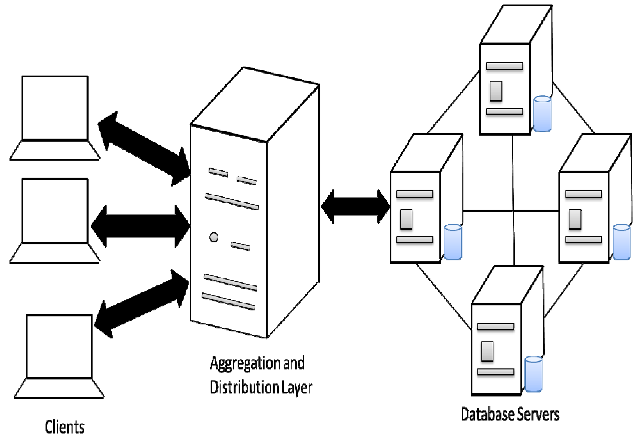

Fig. 2 System Model

In an organization, ADL will be deployed to authorize its staff to share files with the cloud. User's send their requests to the middleware server i.e. ADL, that will combine the requests from multiple user's and sends a single request to the cloud. The cloud executes the single query on collection of files and returns all the matched files to the aggregation and distribution layer. ADL will send the results to all the requested users. To aggregate the user requests, the ADL has to wait for some time before running EIRQ scheme, which will incur some querying delay. Differential query service is introduced, again to reduce the communication cost. To get the required percentage of matched files, user can input a rank to his request. This is beneficial, if there are large numbers of files matching a user's request but the user is in need of only a few of them. User privacy is divided into search and access privacy. In this work, queries are divided into different ranks and thus privacy of rank selected by the users also needs to be ensured. Privacy of the rank means, the cloud has to provide differential query services regardless of the rank selected by the users.

4.1.1 Design Goals

Cost Efficiency: The users can get the matched files on demand to reduce the communication cost.

User Privacy: The cloud knows nothing regarding what the user searching for, which file has been returned and rank chosen by the user.

Efficient Information retrieval for Ranked Query (EIRQ), in which each user can choose the rank of his query to determine the percentage of matched files to be returned. The basic idea of EIRQ is to construct a privacy preserving mask matrix that allows the cloud to filter out a certain percentage of matched files before returning to the ADL.

Two fundamental problems should be resolved: Firstly, we should determine the relationship between query rank and the percentage of matched files to be returned. Suppose that queries are classified into 0 ~ r ranks. Rank-0 queries have the highest rank and Rank-r queries have the lowest rank. In this paper, we simply determine this relationship by allowing Rank-i queries to retrieve (1 − i/r) percent of matched files. Therefore, Rank-0 queries can retrieve 100% of matched files, and Rank-r queries cannot retrieve any files. Secondly, we should determine which matched files will be returned and which will not. In this paper, we simply determine the probability of a file being returned by the highest rank of queries matching this file. Specifically, we first rank each keyword by the highest rank of queries choosing it, and then rank each file by the highest rank of its keywords. If the file rank is i, then the probability of being filtered out is i/r. Therefore, Rank-0 files will be mapped into a buffer with probability 1, and Rank-r files will not be mapped at all. Since unneeded files have been filtered out before mapping, the mapped files should survive in the buffer with probability 1.

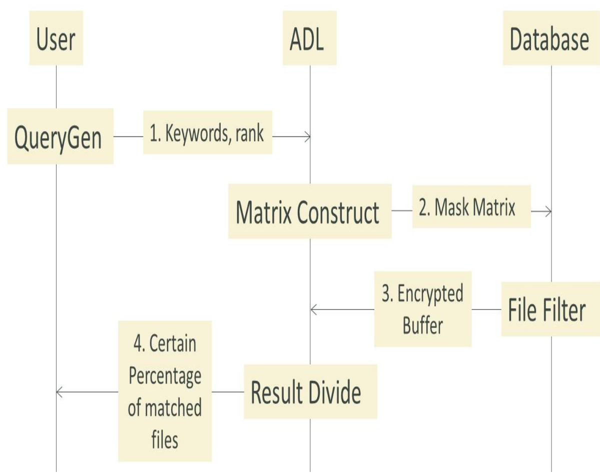

Fig. 3 EIRQ Scheme

[1] The user runs the QueryGen algorithm to send keywords and the rank of the query to the ADL. Since the ADL is assumed to be a trusted third party, this query will be sent without encryption.

[2] After aggregating enough user queries, the ADL runs the MatrixConstruct algorithm to send a mask matrix to the cloud. The mask matrix M is a d-row and r-column matrix, where d is the number of keywords in the dictionary, and r is the lowest query rank. Let M[i, j] denote the element in the i-th row and the jth column, and let l be the highest rank of queries that choose the i-th keyword Dic[i] in the dictionary. M is constructed as follows: for the i-th row of M that corresponds to Dic[i], M[i, 1], . . . , M[i, r − l] are set to 1, and M[i, r − l + 1], . . . , M[i, r] are set to 0, then each element is encrypted under the ADL's public key pk. For the rows that correspond to Rank-l keywords, the ADL sets the first r − l elements, rather than random r − l elements, to 1. The reason is to ensure that, given any Rank-l file Fj , when we choose a random number k, the probability of all of the k-th elements of the rows that correspond Fj 's keywords being 0 is l/r, which is determined by the highest rank of Fj's keywords.

[3] The cloud runs the FileFilter algorithm to return a buffer that contains a certain percentage of matched files to the ADL. Specifically, the cloud multiplies the kth elements of the rows that correspond to Fj's keywords together to form cj, where k = j mod r. Then, it powers |Fj | to cj to obtain ej, and maps the c-e pair into multiple entries of a buffer, as in the Ostrovsky scheme. Note that, with Step 2, we can make sure that, for a Rankl file Fj, the probability of cj being 0 is l/r, and thus the probability of Fj being filtered out is l/r.

[4] The ADL runs the ResultDivide algorithm to distribute search results to each user. File contents are recovered as the FileRecover algorithm in the Ostrovsky scheme. To allow the ADL to distribute files correctly, we require the cloud to attach keywords to the file content. Thus, the ADL can find out all of the files that match user's queries by executing keyword searches.

4.1.2 Algorithm

MatrixConstruct (run by the ADL with public key pk)
for i = 1 to d do
set l to be the highest rank of queries choosing Dic[i]
for j = 1 to r do
if $j \le r - l$ then
M[i, j] = Epk(1)
else
M[i, j] = Epk(0)
adjust γ and β so that file survival rate is 1

FileFilter (run by the database)
for each file Fj stored in the cloud do
for i = 1 to d do
k = j mod r; cj = _Dic[i] ∈Fj
M[i, k]; ej = c
|Fj |
j

map (cj, ej) γ times to a buffer of size β

4.2 ABE Algorithm

An ERP system that manages an organization contains files that need to be stored securely with specific access rights to its wide range of users. Prevailing encrypting techniques prove to be very inefficient in case of such large software system with tons of files and sharing at different levels. There always remains a challenge of blocking the attackers trying to gain access to sensitive information. Techniques such as symmetric key encryption, public-private key encryption or identity based encryption have had its own set of problems as attackers have been able to crack them due to weak access policies based on only unified program logic. Thus manipulating those encrypting techniques have become rather easy and have risen as a new threat to such systems. Also such techniques use single authority keys which put a large set of files at risk once at a time. Owing to these problems we put forth in this system a novel approach to tackle the challenge with multi authority Attribute based encryption.

The main advantage of such MA-ABE system would be its loose coupling with access policy as the client himself will be able to define a set of attributes for which his access policy will work to produce different keys for different levels of access by other clients. Thus we will be able to generate different keys for everyone in the system thereby providing multiple authorities at different levels. Secret keys for every user and cipher text would then be dependent on the attributes defined while encrypting rather than single key generation for the whole file. This would also help in implementing log encryption instead of encrypting each part of a log with the keys of all recipients; it is possible to encrypt the log only with attributes which match recipient's attributes.

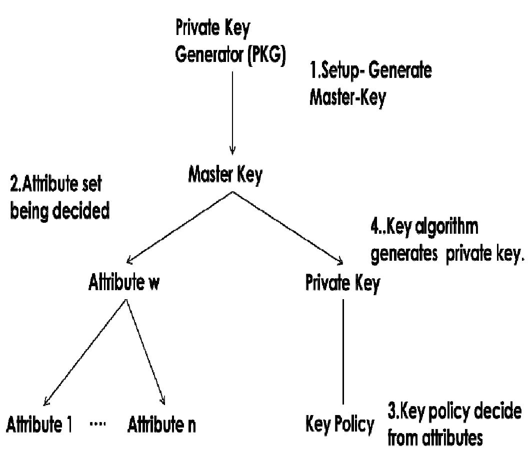

Fig. 4 Flowchart of ABE

Implementing ABE involves following steps as shown in diagram:

1. Master key record will first be generated for the whole system in which care would be taken to include each and every attribute of the system that would define a user.
2. After generating a master key, a set of attributes would be defined for every user in the system
3. Depending on the set of attributes an access policy would be adopted to generate a secret key for every user.
4. Once the secret key is obtained the user further uses this key to gain access to different files as per his authority level.

Let us understand this by considering a simple example as shown in diagram:

Assume three users of system with their respective set of attributes assigned as follows,

Ramesh: w= {Student, Computer Dept.}
Suresh: w'= {Teacher, Computer Dept.}
Paresh w''= {Principal, College}

Now each of them tries to access a file of 'Result' which is encrypted using a set of attributes as follows:
Result r= {(Principal) v ((Teacher) ^ (Computer Dept.))}

In such a case only Ramesh and Paresh will be able to access the file as their attribute set matches with the encrypted files attribute set. Thus their secret key only and not Suresh's will be able to decrypt the file.

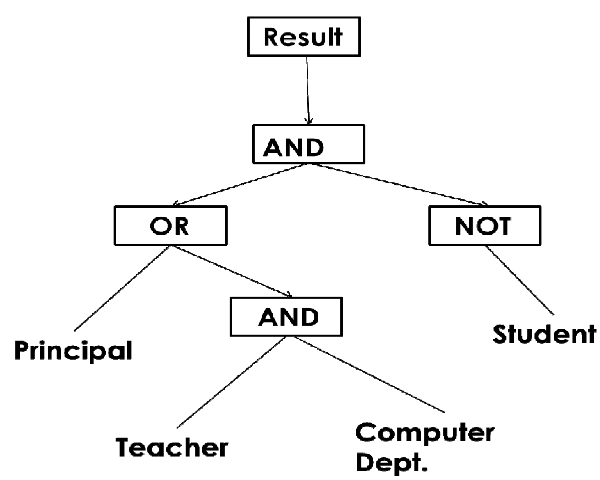

Fig. 5 Example of ABE

5. Mathematical Model

5.1. Problem Statement

- Let ‘S’ be an ERP system such that
  - S= {Server, Starting state, Ending state, A, Success state, Failure state, M, Actors, Database}
- Starting state-> Login page common for all, so many to one relationship at starting page.
- Admin-> One admin per Module…and main one admin, so one to one for every module admin and one to many for main admin.
- M= {Exam, TPO, accounts, Admission, Departments, Events, Library, Encryption}
- Actors= {Students, Teachers, Non-Teaching Staff}
- Database={ Admission, Accounts, Library, Department, Documents}
- End state: changes as per clients request

5.2. Activities

Activity 1
- Admin defines masters by creating their respective databases {admins, courses, departments, subjects, fees}
- F(A)->G(Database) where Database={Departments, Accounts, Library}

Activity 2
- Actor requests for the registration with the system {Students, teachers, non-teaching staff}
- F(Actors)->G(M[Admission]) where M={Admission}

Activity 3
- Accounts confirms minimal payment towards registration
- F(M[Accounts])->G(M[Admission]) where M={ Accounts, Admission}

Activity 4
- Admin validates the requests
- F(A)->G(M[Admission, Accounts])->H(Database)

Activity 5
- Actors fill the database
- F(Actors)->G(database)

Activity 6
- Department creates the academic calendar for actors
- F(M[Department, Events])->G(Actors)

Activity 7
- Department issues notifications for different actors {Timetables, question banks, circulars}
- F(M[Department, Events])->H(Actors)

Activity 8
- Department encrypts certain files and gives discrete authority to different actors
- F(M[Departments])->G(Encryption)->H(Actors)

Activity 9
- Admin creates college events in the database
- F(A)->G(M[Events])->H(Database)

Activity 10
- Library creates database of books
- F(M[Library])->G(Database(Books))

Activity 11
- Actors enquire about certain books in the library
- F(Actors)->G(Library)->H(Database[Books])

Activity 12
- Library issues books to the actors
- F(Library)->G(Database[Books])->H(Actors)

Activity 13
- Actors {Students} submit examination form to examination section
- F(Actors)->G(Examination)

Activity 14
- Examination section verifies the student examination request
- G(Examination) -> F(D[Student/Teacher, Accounts])

Activity 15
- Examination section issues encrypted results to different departments
- F(M[Examination])->G(Encryption)->H(M[Departments])
-

Activity 16
- Admin updates fee structure in accounts as and when needed
- F(A)->G(Database[accounts])

Activity 17

- Actors submit applications and documents to the system
- F(Actors)->G(Databases)

Activity 18

- TPO scans the database for certain queries
- F(M[TPO])->G(Database[Student])

6. Conclusion and Future Scope

The scope of this project is one with perpetual improvement. Owing to the loosely coupled three-tier architecture, updates and upgrades to the system are easy to add. As the database is ever-expanding, in the future it can be easily deployed on the cloud for big data storage. This is because of the portability of the system. The ABE used in our system goes a long way in making storage of data on third-party cloud networks possible by ensuring database security and data integrity. In the future, the system can be easily enhanced by adding more attributes to the database, by adding more sub-modules to increase the depth and scalability of the system. More features can be added to include features like notifications to parents, which let the parents to be up to date with their ward's progress.

To conclude, this project is nearly exhaustive of all the features necessary for managing an institute. The three-tier architecture provides depth, security and portability to the system. The GUI is easy to use, elegant and full of features. The EIRQ algorithm makes searching quick and easy. The ABE makes the data secure and reliable. The system is designed to be easily upgradable to enhance the system's capabilities. Hence, this system can be easily deployed to replace the current paper-based administration system with a better, more advanced and efficient software system.

Acknowledgments

We are thankful to our guide, Mr. Mahesh Bhandari for his constant support and guidance. We are also grateful to Persistent Systems Ltd. for sponsoring us and providing us with industrial exposure. We are indebted towards our H.O.D Mrs. Deeksha Bhardwaj for providing us with her needed support throughout the phase of research. Also we would like to thank our principal Dr. R.D. Kharadkar for providing us the necessary facilities in college. Lastly we would also like to thank staff members for their help when needed.

References

[1] Qin Liu, C. Tan, Member, IEEE, Jie Wu, Fellow, IEEE and Guojun Wang, Member IEEE, 2013.

[2] 'Scalable and Secure Sharing of Personal Health Records in Cloud Computing using Attribute based Encryption' – Ming Li, Member, IEEE, Shucheng Yu, Member, IEEE, Yao Zheng, Student Member IEEE, Kui Ren, Senior member, IEEE and Wenjing Lou, Senior Member, IEEE, 2013.

[3] 'Efficient Information retrieval for ranked queries for cost-effective Cloud Environment' – Q. Liu, C. Tan, J. Wu, G. Wang, 2012.

[4] S. Yu, C. Wang, K. Ren, and W. Lou, "Attribute Based Data Sharing with Attribute Revocation," Proc. Fifth ACM Symp. Information, Computer and Comm. Security (ASIACCS '10), 2010.

[5] R. Curtmola, J. Garay, S. Kamara, and R. Ostrovsky, "Searchable symmetric encryption: improved definitions and efficient con- structions," in Proc. of ACM CCS, 2006.

[6] Q. Liu, C. C. Tan, J. Wu, and G. Wang, "Efficient information retrieval for ranked queries in cost-effective cloud environments," in Proc. of IEEE INFOCOM, 2012.

Parth Amalgaonkar
He is a student pursuing B.E. under Pune University from G.H. Raisoni College of Engineering and Technology. He is working on a project "Enterprise Resource Planning for College Administration" in B.E. which is sponsored by Persistent Systems Ltd.

Supriya Jadhav
She is a student pursuing B.E. under Pune University from G.H. Raisoni College of Engineering and Technology. She is working on a project "Enterprise Resource Planning for College Administration" in B.E. which is sponsored by Persistent Systems Ltd.