International Journal on Advanced Science Engineering Information Technology

Vol.9 (2019) No. 1
ISSN: 2088-5334

A Software Development Methodology for Secure Web Application

Junho Lee[#], Jungwoong Woo[*], Cheongan Lee[+], Kyungsoo Joo[$]

[#] *Department of Computer Science, Soonchunhyang University, 22 Soonchunhyang-ro, Asan, 31538, Republic of Korea*
E-mail: wnsgh461@naver.com

[*] *RockPlace, 844 Eonju-ro, Gangnam-gu, Seoul, 06020, Republic of Korea*
E-mail: jyone0715@gmail.com

[+] *AIBrain, 1 Gwanak-ro, Gwanak-gu, Seoul, 08826, Republic of Korea*

[$] *Department of Computer Software Engineering, Soonchunhyang University, 22 Soonchunhyang-ro, Asan, 31538, Republic of Korea*
E-mail: gsoojoo@sch.ac.kr

***Abstract*—In recent years, there has been a demand for Web applications with complex functions. In addition, most web applications efficiently manage data based on databases. While the key and critical dimension of developing these Web applications is analysis and design, most object-oriented analysis and design methods do not have a consistent view of the database. In addition, Java Enterprise Edition (EE) -based technologies are used in Web application implementations, but they do not provide any correlation with the database. On the other hand, as users' demands for security increase, security becomes more important. To this end, Java EE and database systems provide security solutions. However, it does not provide any correlation with object-oriented analysis and design methodology. As a result, it is difficult to develop secure web applications in a consistent way from analysis to implementation. In this paper, we propose a consistent software development methodology from analysis to implementation of secure web applications. The proposed software development methodology for web application development uses UMLsec, a security-emphasized modeling language, and object-relational (O-R) mapping for relational database design. It also uses Java servlets and SQL to implement analysis and design results based on role-based access control (RBAC). The software development methodology for the secure web application proposed in this paper has been applied to the development of the online banking system, from the design stage of the user's requirements analysis to the implementation of the web application.**

***Keywords*—web application; development methodology; secure web; secure web application; software development.**

I. INTRODUCTION

As security-related requirements are growing, the importance of security also has been gradually increasing. Thus, it is essential to derive security issues from the beginning of the analysis and consistently consider those issues from design to implementation [1], [2]. Java EE and the database provide solutions to resolve those security issues such as role-based access control. However, if we do not consider security from the analysis and design, those solutions of Java EE and the database can be used inconsistent ways to produce a vulnerable web application from cyber-attacks [3]–[5].

Therefore, we propose a software development methodology for a secure web application. This methodology reflects security consistently from requirement analysis to implementation with CBD (Component Based Development), UMLsec, and O-R mapping [6]. Implementation for security is based on role-based access control using Java EE, and DDL (Data Definition Language) and DCL (Data Control Language) of SQL.

The rest of this article is organized as follows: Section 2 introduces related works. Section 3 explains proposed a software development methodology for secure web application and applies the proposed methodology on a banking system for the verification of the methodology. Finally, Section 4 concludes this paper.

II. MATERIAL AND METHOD

A. Object-oriented analysis and design methodologies

The goal of the CBD methodology is to develop software systems based on components that can respond appropriately to changes in requirements that the user wants [7]. On the other hand, it is possible to create an object-oriented

program as a conceptual model designed with the existing object-oriented analysis design methodology, but this does not provide a consistent design methodology for security [4].

B. The design methodology of the Relational Database

The information engineering methodology has been widely used to design a relational database [8]. Concerning security matters, the database system supports role-based access control technologies. However, the information engineering methodology does not consider the security matters at the initial stage of design, and as a consequence, it cannot provide consistency in security throughout the entire development process.

III. Results And Discussion

In this paper, we propose a software development methodology for secure web applications. The proposed methodology adds the definition of security requirements in the requirements analysis as shown in Fig. 1, and the definition of the added requirements uses UMLsec. Also, UMLsec is used to emphasize security in the analysis and design phase. Also, O-R mapping is applied to relational database design. Finally, in the implementation phase, we implemented the role-based access control using Java EE, DDL and DCL syntax based on the result of the previous step. The requirements other than security were designed by applying the CBD methodology.

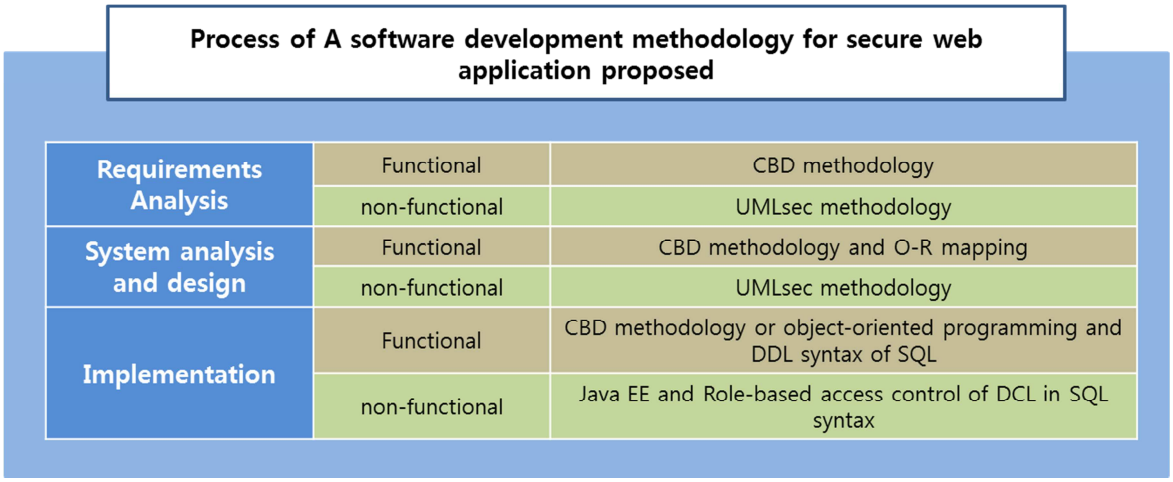

Fig. 1 The process of a software development methodology for secure web application

A. Requirement Analysis

1) Writing Requirement List: Through the requirements definition phase, users are identified and verified for the requirements they have with the software. At this stage, not only basic requirements but also security requirements are derived [6], [9]. Table 1 lists some of the requirements of the online banking system, including the requirements definition of security requirements.

TABLE I
Online Banking System Requirement List.

- The user can use the lookup service.
- The lookup service can check balance, check transaction list, check history, and download.
- The user can pay a variety of taxes using the payment service.
- Users may use transaction services.
- Transaction service is a function that includes functions such as money transfer.
- Administrators have overall access to the system through the management function, and can also create and delete new accounts, adjust balances, cancel transactions, and set user ratings.
- Can set system permissions for specific users.
- This system is only used with login.
- This system must have to data management and protection.

Table 2 can make a detail definition only the requirement of security from the requirement list in Table 1. Table 2 has 4 security types. Number 1 is for the authority of an administrator. Number 2 and 3 is for certification, authorization. Number 4 is necessary for data integrity and confidentiality.

TABLE II
Type of Security Requirements.

Type	Description
Security	• Administrators have overall access to the system through management function and can also create and delete new accounts, adjust balances, revoke Deal, and set user ratings. • This system is only used with login. • Administrators can set a system rating for specific users. • This system must have to data management and protection.

The use case is a list of steps to perform a task and a way to capture the requirements of new software or a software revision [6], [9]. The use cases are created according to the list of some user requirements of the online banking system defined in Table 1, and the security requirements are expanded by applying the UMLsec methodology [2]. After writing a use case, we should stipulate a summary, actors, priority, pre/post conditions, a scenario, and non-functional requirements for each item of the use case [9]. Furthermore, for the security use case, we should write security issues concisely and clearly in non-functional requirements by referring to Table 1. When creating a use case model, each function that the system will provide is expressed by a use case, and the actor represents an entity outside the system that interacts with the use case. To visualize the model, we use the use case diagram of UML [9], [10].

2) Writing Use Case: The use case is a list of steps to perform a task and a way to capture the requirements of new software or a software revision [6], [9]. The use cases are created according to the list of some user requirements of the online banking system defined in Table 1, and the security requirements are expanded by applying the UMLsec methodology [2]. Table 3 is part of the use case of the online banking system. Table 4 shows the extended use case for security.

TABLE III
LIST OF USE CASE.

Use case	Description
Register	Users who will use the system must register.
Login	Users who will be using the system must log in.
Rating setting	The administrator sets the user's rating.

TABLE IV
USE CASE WITH SECURITY REQUIREMENT.

Use Case: Rating Setting
Risks related to actors
- Users can verify their information. Administrators can view and edit information for all users.

I/O data that requires security & I/O data that does not require security
The behavior of modify system
- The user must have a register.
- The certification process is a step that the user must take. Otherwise, the user will not use the system.
- The system should output an error message when the information entered during the authentication step is incorrect.
- Administrator set the User access rating.
- The user can see the result of the system.

3) Elaboration of Use Case Model

After writing a use case, we should stipulate a summary, actors, priority, pre/post conditions, a scenario, and non-functional requirements for each item of the use case [9]. Furthermore, for the security use case, we should write security issues concisely and clearly in non-functional requirements by referring to Table 2. The use cases of security requiring grading described by Table 5. The variety of situations, i.e., scenarios was written by use case description [9]. Table 6 is a general scenario for setting the use case user's ratings.

TABLE V
USE CASE DESCRIPTION FOR RATING SETTING

Item	Description		
Name	Rating Setting		
Summary	Administrators can set access for each user.		
Relevant Actors	Main Actor	Administrator	
Priority	1	Importance	1(High)
		Difficulty	1(High)
Preconditions	• The user must log in with the administrator account. • The user you want to set up must be registered.		
Postconditions	• The user's login status should not be released. • Through the system, the administrator can check the changed user information. • The rating of the changed user must be recorded in the system.		
Scenario	General scenario	General scenario betwixt the system and actor	
Non-functional Requirements	Security requirement • All systems are accessible by the administrator. • Administrators can set system access for specific users.		

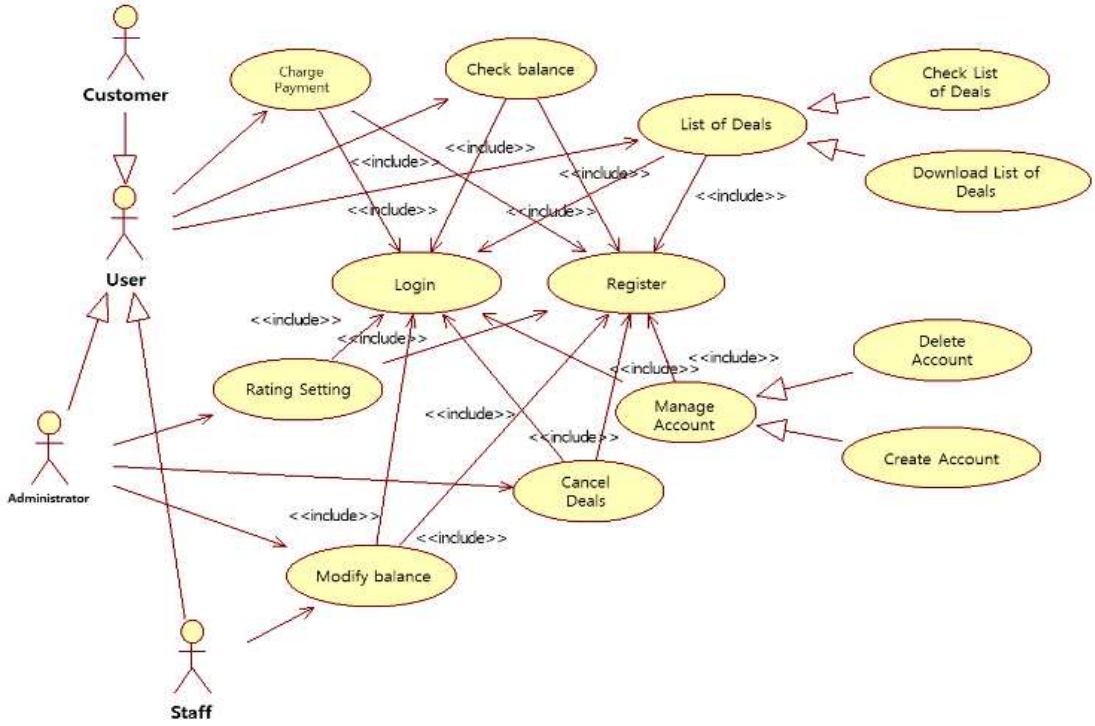

Fig. 2 Use case model for an online banking system

TABLE VI
GENERAL SCENARIO OF A USER'S RATING SETTING USE CASE

- The user must be Sign up.
- The user input the ID and password for the administrator on the login page then presses the button.
- The system displays the administrator page. The administrator chooses the rating setting from the administrator page.
- On the rating setting page, you can see the rating of the user. To change the rating, press the rating setting button.
- The system displays the detailed rating information page.
- Display information page: rating, name, ID
- When the rating is modified, the administrator presses the OK button. The system has two functions such as save the changed data and update the detailed rating data page.
- To go to the previous page click the revoke.

4) Writing Use Case Model

When creating a use case model, each function that the system provides are represented by use cases, and actors represent entities outside the system that interact with use cases. To visualize the model, we use the use case diagram of UML, which shows the relationship between actors and use cases [9], [10]. Fig. 2 was written the use case model of the online banking system.

B. System Analysis and Design

Identification of each element of the system to satisfy the user's requirements can be performed at the system analysis, and the design stage can be based on the requirements model of the previous step [9]. Fig. 3 is the system analysis and design process of a software development methodology for the proposed secure web application.

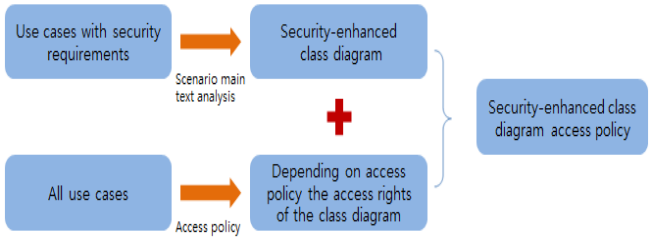

Fig. 3 The process of creating class diagrams that emphasize security by the access policy

The textual analysis of the use case is a method to extract required classes for the system from the general scenario of a use case which is written based on requirements from users [6], [9]. Boundary, control, and entity classes can be extracted by analysis of the use case body. Boundaries classes are extracted from phrases as if ~ screen and nouns are having permanency are extracted as entity classes.

Boundary classes cannot access entity classes directly, so control classes are used for business and control logic [9]. Next, individual actors must have access by use case as described in [2]. In this phase, one can describe clearly about access rights for use cases based on the security use case description and the general use case description [11]. The created access policy represents the access right to the class diagram to be derived later. Table 7 defines access policies for some use cases of online banking systems.

TABLE VII
ACTOR-BASED USE CASE ACCESS POLICY

	Customer	Staff	Administrator
Register	X	X	X
Login	X	X	X
Manage Account	P	X	X
Check Balance	P	X	X
Check List of Deals	P	X	X
Download List of Deals	P	X	X
Charge Payment	X	-	X
Create Account	-	-	X
Delete Account	-	-	X
Modify Balance	-	X	X
Cancel Deals	-	-	X
Rating setting	-	-	X
All permission(X), Some permission(P), No permission(-)			

After writing the access policy, we should describe an analysis class diagram by textual analysis of a use case scenario: extracting classes and defining relationships of the classes. Classes derived from use cases with security requirements are classes that emphasize security. Each class creates an access policy as shown in Table 7 according to the UMLsec methodology in the << secrecy >> stereotype.

In the refinement of the analysis class diagram, the use case scenario is further analyzed in text based on the security-enhanced class diagram derived from the previous activity, and the attributes and operations of each analysis class are defined [8], [10].

1) Applying to MVC Pattern based on Java EE

Apply the MVC pattern to the detail analysis class as ① - ④[6].

① The class using the << entity >> stereotype is mapped to Model.

② Class that uses << boundary >> stereotype is implemented as JSP etc. as View.

③ << control >> A class that uses stereotypes is implemented as a servlet in the role of the controller.

④ << secrecy >> Classes that use stereotypes are classes that should be emphasized security. If used with << control >> and << boundary >> stereotypes, implement them by using role-based access control of Java EE. If it is used with << entity >> stereotype, it is implemented using role-based access control of the database.

2) Design Method of Relational Database

The design of a relational database is performed through O-R mapping. O-R mapping allows you to convert analytic class diagrams derived from OOAD (object-oriented analysis and design) methodologies into relational data models. By using Table 8, the class diagram can be converted to a relational database schema [12]–[14].

TABLE VIII
O-R MAPPING FOR CONVERSION TO RELATIONAL DATABASE

- The class is mapped to the table.
- The class attribute is mapped to a column of the table.
- The class attribute type is mapped to column type of table
- For the classes having no generalization, the primary integer key is created. For {oid},{oid} tag column is added to primary key constraint.
- Subclasses add each parent key to the constraints of the primary key and foreign key.
- If an attribute contains {nullable} tag, NULL or NOT NULL is added in table attribute.
- If an attribute has an initial value, DEFAULT is added in the column.
- Association classes add the primary key for each role perform table to the constraint of a primary key and foreign key.
- If {alternate oid=<n>} tag is found, column for UNIQUE constraint is added.
- CHECK is added on each specified constraint.
- The foreign key is created in the table wherein a referring is made in association with 0.1 and 1.1 rule.
- Primary key, for the complex set having a foreign key of settable, is created. For the primary key, an additional column is added.
- For optimization, binary association classes are moved to an adequate "N" side table.
- Triple association, which is not an association class is, created by N: N table.
- In N: N and triple associations, the constraint for primary key and foreign key are created from the key for role perform table.
- 16. For many-to-many association having no associate-ion class, primary key and foreign key are created.

C. Implementation

Because << control >> and << secrecy >> are used in the 'User Management' class associated with the 'Set Rating' use case, define roles to enforce the security mechanisms of Java EE. Table 9 defines roles for authentication and authorization. Table 10 shows the contents of the authentication. As mentioned above, there are four ways of CLIENT-CERT, DIGEST, FORM, and BASIC. In this paper, authentication is implemented as a FORM in '<form-login-page>' and '<form-error-page>,' and if the authentication is FORM, then developers can define to open the arbitrarily created pages. Fig. 5 shows the authentication for the FORM implementation. It shows the random login error page when an unregistered user login.

TABLE IX
ROLE DEFINING

```
- Tomcat-user.xml
<? xml version='1.0' encoding='utf-8'?>
…
    password="admin1234" roles="admin"/>
…
</tomcat-users>
```

TABLE X
IMPLEMENTATION OF AUTHENTICATION

```
- web.xml
<login-config>
<auth-method>FORM</auth-method>
<form-login-config>
<form-login-page>/login.jsp</form-login-page>
<form-error-page>/loginerror.html</form-error-page>
</form-login-config>
</login-config>
```

Table 11 and Table 12 lists the authorization steps. For requests to servlets, you must map the appropriate role to the deployment descriptor. Accessible resources and available HTTP methods must be specified. If a user who is not an administrator tried to access a management page, an error page is opened like in Fig. 5. Fig. 6 shows a customer management page, which is opened when an administrator accesses the page properly. The page is accessed through HTTPS for confidentiality and data integrity.

Fig. 4 Login error page

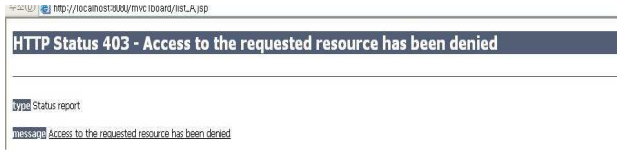

Fig. 5 Error page due to access permissions

TABLE XI
REGISTRATION OF ROLE

```
- Web.xml
<security-role>
<role-name>admin</role-name>
<role-name>customer</role-name>
<security-role>
```

TABLE XII
DEFINING OF RESTRICTION

```
- Web.xml
…
<url-pattern>/admin/Member.jsp</url-pattern>
      <http-method>GET</http-method>
      <http-method>POST</http-method>
…
</security-constraint>
```

Table 13 shows the user Information' table, one of the table schema converted through O-R mapping, created by 'CREATE TABLE' command of DDL syntax of SQL. As <<secrecy>> stereotype was applied during the design process, all the classes of created tables are security emphasized ones. So, for each table, the role-based access control can be configured by the 'GRANT' command of DCL syntax of SQL. Table 14 shows the access control schema for 'User' and 'Administrator' roles.

TABLE XIII
'INFORMATION_USER' TABLE

```
CREATE TABLE information_user (
id_num INTEGER PRIMARY KEY,
id_u VARCHAR(10) NOT NULL,
password_u INTEGER NOT NULL,
name_u VARCHAR(12) NOT NULL,
cellphone_u VARCHAR(15) NOT NULL,
address_u VARCHAR(30) NOT NULL,
grade_u VARCHAR(6) NOT NULL,
acc_num_u INTEGER REFERENCE account,
tran_num_u INTEGER REFERENCE transaction,
CONSTRAINT info_PK PRIMARY KEY (id_num, acc_num_u,
tran_num_u) );
```

TABLE XIV
ADMINISTRATOR AND USER ROLES: ACCESS SCHEMA

```
CREATE ROLE user_entry;
GRANT user_entry TO user_view;
…
GRANT ALL ON
transaction TO admin_view;
```

The software development methodology for the secure web application proposed in this paper provides a consistent analysis and design method for security, which is different from the existing OOAD methodology and provides correlation with Java EE and database, which UMLsec cannot provide, do. Accordingly, we proposed 'A software development methodology for a secure web application,' which is applied to the existing OOAD methodology, security, Java EE, and database.

We verified the efficacy of the proposed methodology by examples of the online banking system. Using authentication, we can prevent attacks that try to disguise an unauthorized user as an authorized user, like in Fig. 4. Using authorization, we can prevent one who tries to conceal one's grade, like in Fig. 5. Using confidentiality and data integrity, we can prevent one who tries to edit or eavesdrop important user information, like in Fig. 6.

Fig. 6 Customer management page

IV. CONCLUSIONS

This paper proposes an integrated software development methodology for a secure web application. To this end, security emphasized modeling language UMLsec and O-R mapping for relational database design is used. Also, to implement the analysis and design results of security, we implemented the role-based access control using DDL and DCL syntax of Java EE and SQL. Accordingly, the security application was reflected then a consistent method was to the entire system development cycle.

A software development methodology for a secure web application proposed in this paper provides a consistent analysis and design method for security that was not provided by existing OOAD methodology. Also, the association with Java EE and database that UMLsec does not provide is also provided through role-based access control. Therefore, it is possible to securely and consistently analyze and design the pre-system development cycle by firmly presenting the existing OOAD methodology, security, then the correlation between Java EE and relational database. Our methodology is verified by applying it to the development of an online banking system.

ACKNOWLEDGMENT

The Soonchunhyang University Research Fund supported this research.

REFERENCES

[1] Eduardo Fernandez-Medinaa, Juan Trujillob, Rodolfo Villarroelc, and Mario Piattinia., 2007, "Developing secure data warehouses with a UML extension," *Journal Information Systems archive*, Vol 32, No 6, pp. 826-856.

[2] G.Popp, J. Jurjens, G.Wimmel, R. Breu., 2003, "Security-Critical System Development with Extended Use Case," *Asia-Pacific Software Engineering Conference*, 5-1 self.

[3] Madan, s., 2010, "Security Standards Perspective to Fortify Web Database Applications From Code Injection Attacks," *International Conference on Intelligent Systems, Modelling and Simulation(ISMS)*, Vol. 10, pp. 226-230.

[4] lqra Basharat, Farooque Anam, Abdul Wahab Muzaffar., 2012, "Database Security and Encryption: A Survey Study," *International Journal of Computer Application*, Vol. 47, No. 12, pp28-34.

[5] David Basin, Jürgen Doser, and Torsten Lodderstedt., 2006, "Model Driven Security: from UML Models to Access Control Infrastructures," *ACM Transactions on Software Engineering and Methodology (TOSEM)*, Vol. 15 No. 1, pp39–91.

[6] Kyung-Soo Joo, Jung-Woong Woo., 2012, "A Development of the Unified Object-Oriented Analysis and Design Methodology for Security-Critical Web Application Based on Object-Relational Database –Focusing on Oracle 11g-", *Korea Society of Internet Information*, Vol 17, No 12, pp. 169-177.

[7] Byeong-Seon Jeon., 2005, *CBD WHAT&HOW*, Wowbooks Publishing Company, Seoul.

[8] Heung-Seok Chae., 2009, *Object-oriented CDB Project for UML and Java as learning*, Hanbit Media. Seoul.

[9] Mang Su, Fenghua Li, Guozhen Shi, and Li Li, "An Action-Based Access Control Moedl for Multi-level Security," *IJSIA*, 6, pp. 359-366 (2012).

[10] Allaoua Maamir, Abdelaziz Fellah, Lina A. Salem, "Fine Granularity Access Rights for Information Flow Control in Object-Oriented Systems," *IJSIA*, 2, pp. 81-92 (2008).

[11] Brett D. McLaughlin, Gary Pollice, David West., 2007, *Head First Object-Oriented Analysis & Design*, habit media, Seoul.

[12] Seung-Yun Bang, Kyung-Soo Joo., 2003, "Design Methodology for XML Schema Application based on UML," *Soonchunhyang Univ*, pp.71-75.

[13] Mang Su, Fenghua Li, Guozhen Shi, Li Li., "An Action-Based Access Control Model for Multi-level Security.", *International Journal of Security and Its Applications*, 6(2), 359-366. 2012

[14] Egbunike, Celestine, and S. Rajendran. "The implementation of the negative database as a security technique on a generic database system.", *Circuit, Power and Computing Technologies (ICCPCT), 2017 International Conference on. IEEE*, 2017.