

# **An Object Oriented Approach to model Automated Library Book Management Systems**

R.Valarmathi<sup>1</sup>, R.Thiyagaraj<sup>2</sup>, J.Jayapriya<sup>3</sup>, Prof.M.Sowmiya<sup>4</sup> and Prof.S.Balamurugan<sup>5</sup>

Department of IT, Kalaingar Karunanidhi Institute of Technology, Coimbatore, TamilNadu, India<sup>1,2,3,4,5</sup>

**ABSTRACT:** This paper examines the relationship between students and the library. It has been acknowledged that library management serves a major role for the students learning. The proposed modeling methodology is based on the principle of object orientation, which allows describing both software and functionalities explicitly. Furthermore, it is illustrated how the well-known object-oriented specification language unified modeling language can be adopted, to provided an adequate formalization of its semantics, to describe structural and behavioral aspects of online personal finance management system related to both logical and physical parts. It is needed to implement the software on the basis of Object Oriented Model developed. Flaw in modeling process can substantially contribute to the development cost and time. The operational efficiency may be affected as well. Therefore special attention should be paid to the correctness of the models that are used at all planning levels and hence Unified Modeling Language (UML) takes its impeccable role.

**KEYWORDS:** Functional Model, Behavioral Model, Object, Modeling, Operational Efficiency.

## **I. INTRODUCTION**

Most of the effort to date in object oriented community has been focused on programming language issues. Object-oriented programming languages are useful in removing restrictions due to the inflexibility of traditional programming languages. In a sense, however, this emphasis is a step backwards for software engineering by focusing excessively on implementation mechanisms, rather than the underlying thought process that they support.

The real pay-off comes from addressing front-end conceptual issues, rather than back-end implementation issues. Design flaws that surface during implementation are most costly to fix than those that are found earlier. Focusing on implementation issues too early restricts the design choices and often leads to inferior product. An object-oriented development approach encourages software developers to work and think in terms of application domain through most of the software engineering lifecycle.

Object-oriented development is a conceptual process independent of programming language until final stages. Object-oriented programming is fundamentally a new way of thinking and not a programming technique. Its greatest benefits come from helping specifiers, developers, and customers express abstracts concepts clearly and communicate them to each other. It can serve as a medium for specification, analysis, documentation and interfacing as well as programming. Unified Modeling Language (UML) is a standardized general-purpose modeling language in the field of software engineering. UML includes a set of graphical notation techniques to create abstract models of specific systems, referred to as UML model. The Unified Modeling Language (UML) is a graphical language for visualizing, specifying, constructing, and documenting the artifacts of a software-intensive system. The Unified Modeling Language offers a standard way to write a system's blueprints, including conceptual things such as business processes and system functions as well as concrete things such as programming language statements, database schemas, and reusable software components. UML is officially defined by the Object Management Group (OMG) as the UML metamodel, a Meta-Object Facility metamodel (MOF). Like other MOF-based specifications, UML has allowed software developers to concentrate more on design and architecture.

# International Journal of Innovative Research in Science, Engineering and Technology

(An ISO 3297: 2007 Certified Organization)

Vol. 4, Issue 2, February 2015

UML models may be automatically transformed to other representations (e.g. Java) by means of QVT-like transformation languages, supported by the OMG. UML is extensible, offering the following mechanisms for customization: profiles and stereotype. The semantics of extension by profiles have been improved with the UML 1.0 major revision. It is very important to distinguish between the UML model and the set of diagrams of a system. A diagram is a partial graphical representation of a system's model. The model also contains a "semantic backplane" — documentation such as written use cases that drive the model elements and diagrams. UML diagrams represent three different views of a system model:

- **Functional requirements view:** Emphasizes the functional requirements of the system from the user's point of view. And includes use case diagrams.
- **Static structural view:** Emphasizes the static structure of the system using objects, attributes, operations and relationships. And includes class diagrams and composite structure diagrams.
- **Dynamic behavior view:** Emphasizes the dynamic behavior of the system by showing collaborations among objects and changes to the internal states of objects. And includes sequence diagrams, activity diagrams and state machine diagrams.

## II. CLASS DIAGRAM

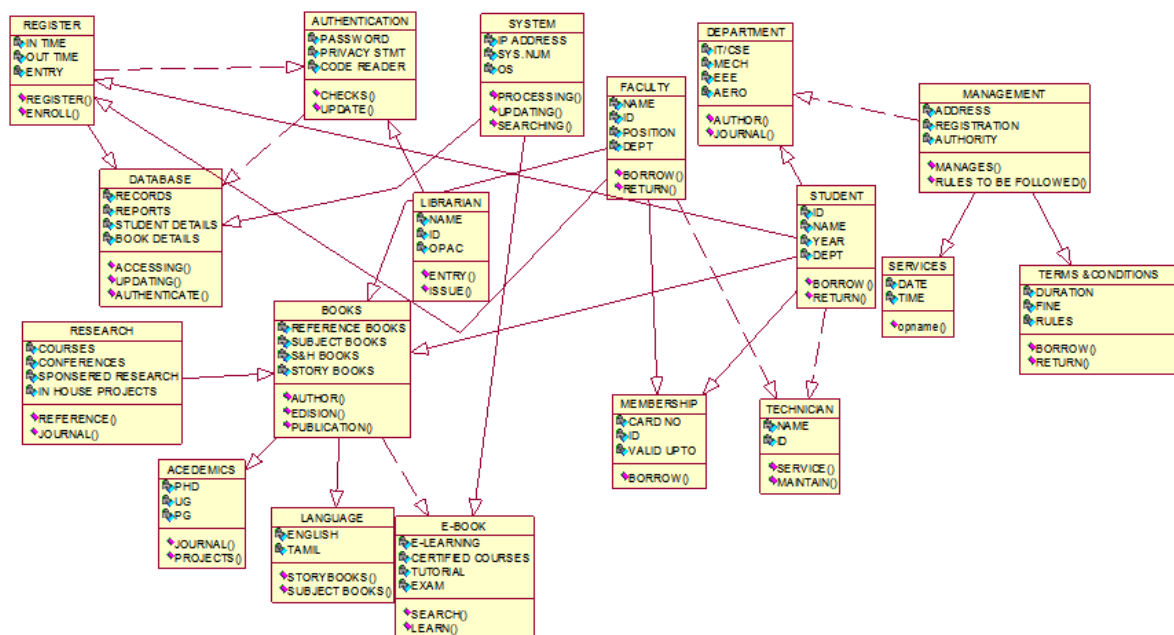


Fig Class Diagram for Book Bank Management System

## III. USE CASE DIAGRAM

Use case diagrams model the functionality of a system using actors and use cases. Use cases are services or functions provided by the system to its users. The components in a use case diagram include:

- **Actors:** Actors represent external entities of the system. These can be people or things that interact with the system that is being modeled. For example, if we are modeling an online store we have many actors that interact with the store functionality. The customer browses the catalog, chooses items to buy, and pays for those items. A stocker will look at the orders and package items for the customer. A billing system will charge the customer's credit card for the amount purchase.
- **Use Cases:** Use cases are functional parts of the system. When we say what an actor does, that's a use case. The customer "browses the catalog", "chooses items to buy", and "pays for the items". These are all use cases. Many

# International Journal of Innovative Research in Science, Engineering and Technology

(An ISO 3297: 2007 Certified Organization)

Vol. 4, Issue 2, February 2015

actors can share use cases. If we find a use case that is not associated with any actor, this may be a unnecessary functionality.

- **Associations:** Associations are shown between actors and use cases, by drawing a solid line between them. This only represents that and actor uses the use case.

There are also two kinds of relationships between use cases:

- **Includes :** Use cases that are associated with actors can be very general. Sometimes they "include" more specific functionality. For example, the "pump gas" use case that is associated with the customer includes three use cases: Choose Gas Type, Fill Tank, and Calculate Total. Includes relationship is represented by dashed arrows that point to the included functionality. Beside the arrow is <<includes>>.

- **Extends :** An extension use case is an insertion to the base use case. For example, some stores may allow for different payment options like credit card, debit card, or cash on delivery. These specific functionalities are extension of the general "pay for items." Extends relationship is represented by dashed arrows that point to the base functionality. Beside the arrow is <<extends>>

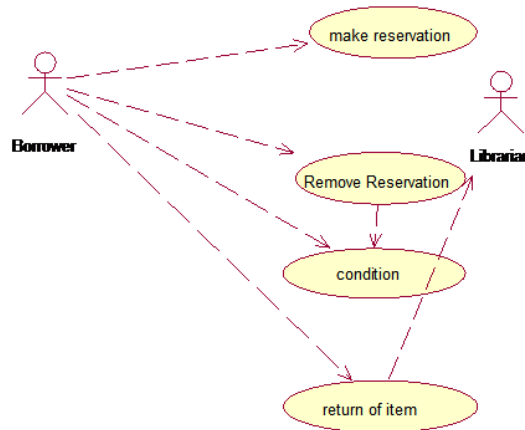


Fig2. Use Case Diagram for Borrower Reservation System

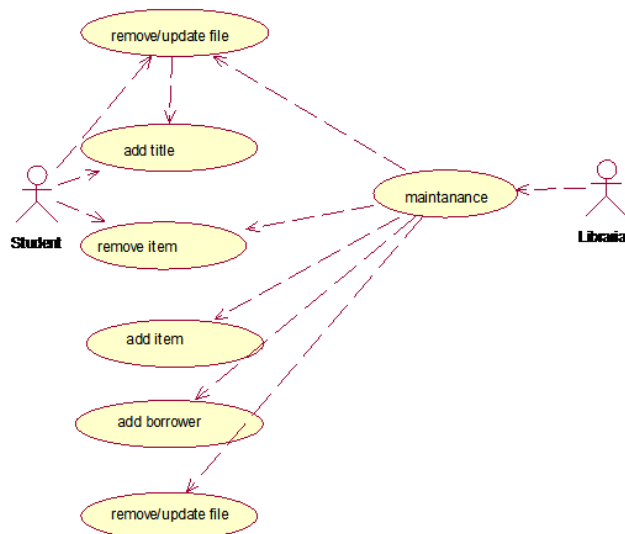


Fig 3. Use Case Diagram for Library Management System

# International Journal of Innovative Research in Science, Engineering and Technology

(An ISO 3297: 2007 Certified Organization)

Vol. 4, Issue 2, February 2015

## IV. INTERACTION DIAGRAM

Interaction diagrams are used to model the behaviour of the use cases by describing the way in which the group of objects interact to the complete task. There are two types of interaction diagram sequence and collaboration diagram. In order to model the behaviour of the several objects in a use case diagram interaction diagrams are used. This diagram is to demonstrate how the object supports for the behavior. Meanwhile the collaboration diagram does not give the deep representation of the behaviour. The interaction of object in the use case is done by sequence diagram, collaboration diagram or by both diagrams. The sequence diagrams are used to represent the sequence of events that occurs. Collaboration diagrams are used to represent how the objects are connected.

The interactions which has occurred are shown by the sequence diagram in the sequential order.

Diagram Elements:

- **Object.** Each of the objects that participate in the sequence diagram. It is drawn across the top. Note that objects are used in this diagram while classes are used in use cases, class diagrams.
- **Lifeline.** A dotted line is dropped from each object in the sequence diagram. Arrows terminating on the lifeline indicate messages (commands) sent to the object. Arrows originating on the lifeline indicate messages sent from this object to another object. Time flows from top to bottom on a sequence diagram.
- **Active.** To indicate that an object is executing, i.e., it has control of the CPU, the lifeline is drawn as a thin rectangle.
- **Message.** A horizontal arrow represents a message (command) sent from one object to another. Note that parameters can be passed as part of the message and can (optionally) be noted on the diagram.
- **Return.** When one object commands another, a value is often returned. This may be a value computed by the object as a result of the command or a return code indicating whether the object completed processing the command successfully. In some instances the object may not be able to return this information immediately. This indicates the flow of information was based on a previous request.
- **Iteration.** Square brackets preceded by an asterisk (\*) indicate iteration. The message is sent multiple times. The expression within the brackets describes the iteration rule.
- **Deletion.** An X is used to indicate the termination (deletion) of an object

The below diagram shows the sequence of operations that are being performed when a student approaches the college administration for admission. Once if a particular student is admitted he has to pay fees to conform his admission. Then the administration also appoints the staffs for each departments according to their qualification.

# International Journal of Innovative Research in Science, Engineering and Technology

(An ISO 3297: 2007 Certified Organization)

Vol. 4, Issue 2, February 2015

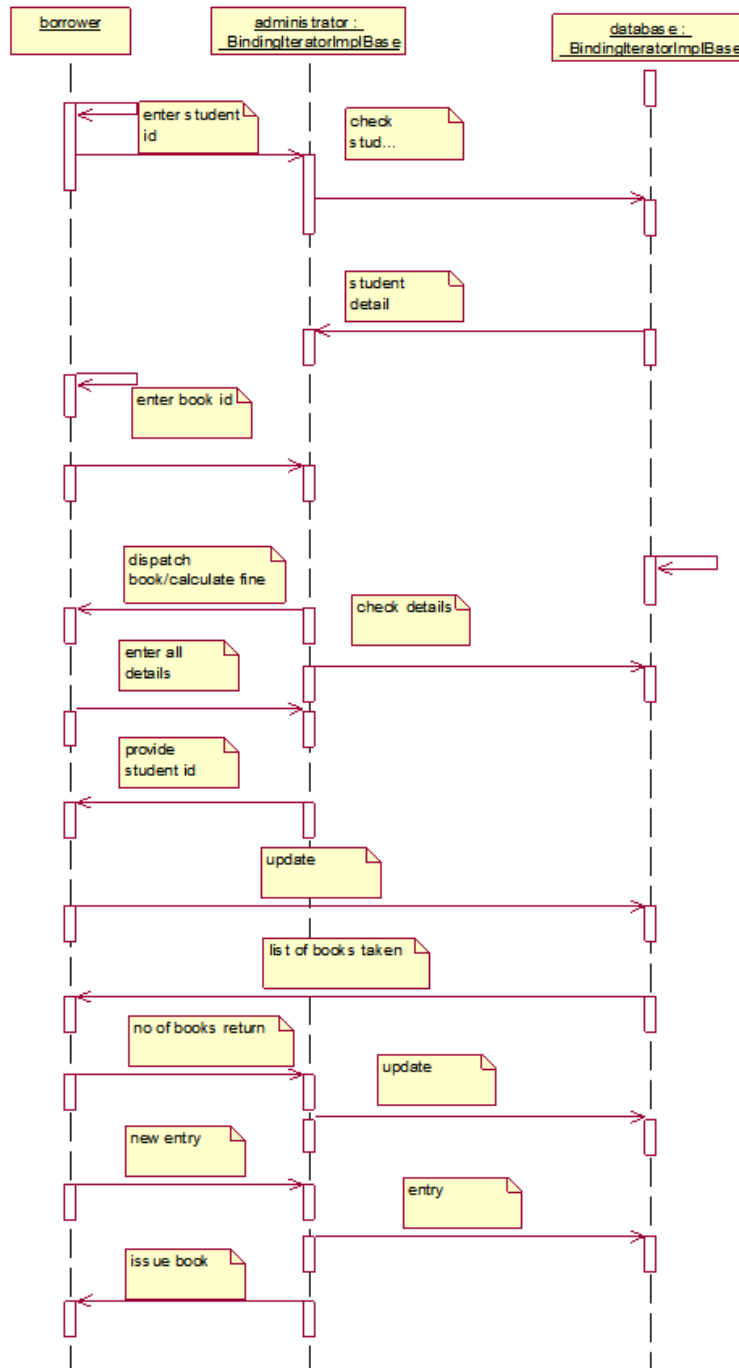


Fig 4. Sequence Diagram for Issue of Book

# International Journal of Innovative Research in Science, Engineering and Technology

(An ISO 3297: 2007 Certified Organization)

Vol. 4, Issue 2, February 2015

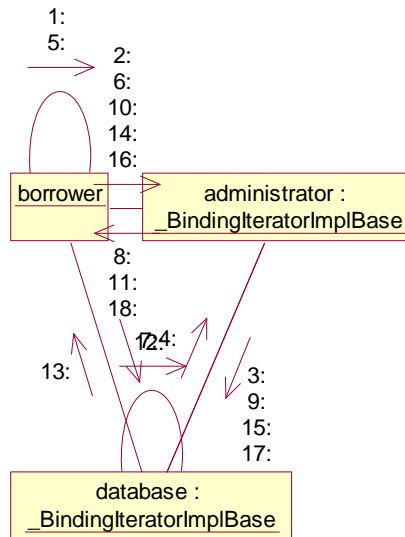


Fig 5. Collaboration Diagram for Issue of Book

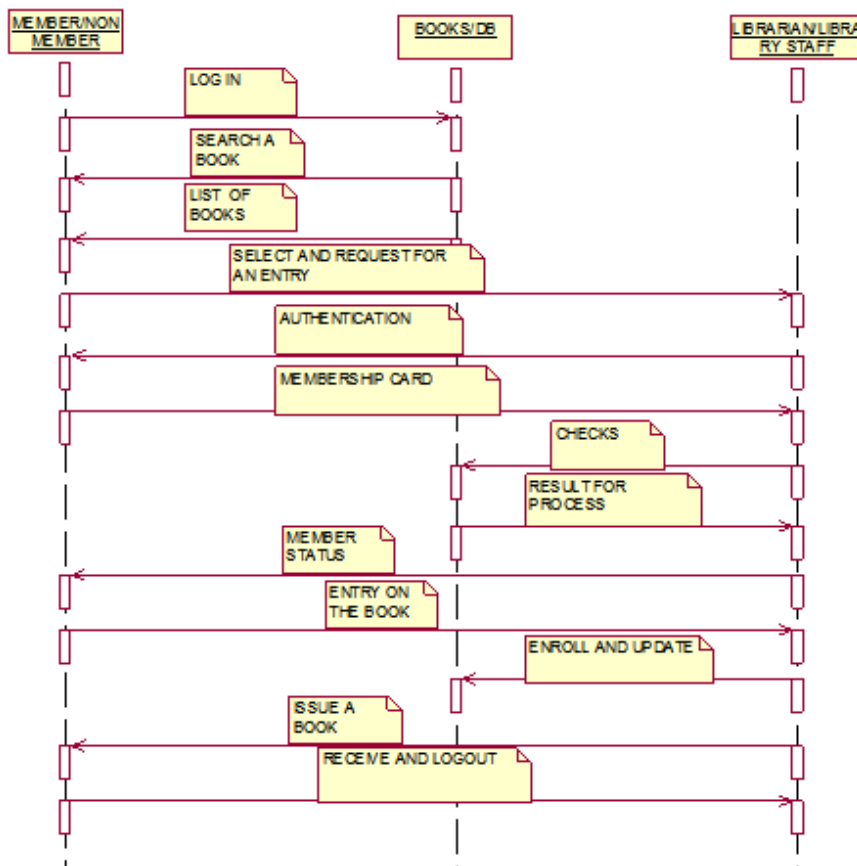


Fig 6. Sequence Diagram for Searching of Book

# International Journal of Innovative Research in Science, Engineering and Technology

(An ISO 3297: 2007 Certified Organization)

Vol. 4, Issue 2, February 2015

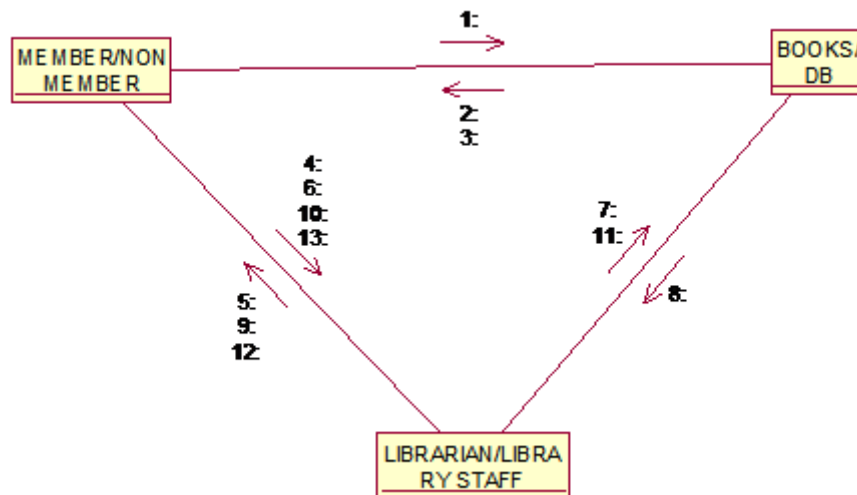


Fig 7.Collaboration Diagram for Searching of Book

## V. CONCLUSION AND FUTURE WORK

The paper has focused on the essential and extended Object Oriented features of a library book management system. Various models of UML have been employed to analyze the static, dynamic and functional view of the system. In spite there are lot of functionalities rendered by mobile phones using recent emerging technologies, the mobile phones advancement deserves a major breakthrough. In a few years from now it is expected that mobile phones would replace most of the handheld systems, incorporating modern operating systems such as android 2.3 gingerbread, meego and windows phone and its emergent higher versions. In order to further enhance the features provided by the current day smartphones, an object oriented approach would add in the developmental growth. Our further work is focused on developing an ER modeling approach to implement with JAVA as a frontend and MY SQL, the online database as a backend and by establishing JDBC connectivity. The same is planned to be tested using Rational Test Manager.

## REFERENCES

1. B.Powmya, Nikita Mary Ablett, V. Mohana Priya, S. Balamurugan, "An Object Oriented Approach to model the secure Healthcare Database system", In the proceedings of international conference on computer, communication and signal processing (IC3SP), in Association with IETE Students Forum and The Society of Digital Information and Wireless communication SDIWC, 2011, pp.2-3.
2. Conrad Bock, U.S. National Institute of Standards and Technology, "UML 2 Composition Model", Journal of Object Technology, Vol.3, No.10, 2007
3. Thomas M. Atwood, "The Case for Object Oriented Databases" IEEE Spectrum, 0018-9235/91/ 0002-004, pp 44-48, 2007
4. James Rumbaugh, Michael Blaha, William Premerlani, Frederick Eddy, William Lorenson "Object Oriented Modeling and Design", Prentice-Hall of India Private Limited.
5. Grady Booch, "Object-Oriented Analysis & Design with Applications", Pearson Education
6. Robert S. Pressman, "Software Engineering: A Practitioner's Approach" Fifth Edition, McGraw-Hill Higher Education.
7. Armour, Frank, and Granville Miller. *Advanced Use Case Modeling: Software Systems*. Addison-Wesley, 2000.
8. [http://atlas.kennesaw.edu/~dbraun/csis4650/A&D/UML\\_tutorial/state.htm](http://atlas.kennesaw.edu/~dbraun/csis4650/A&D/UML_tutorial/state.htm)
9. <http://visualcase.com/tutorials/interactions-diagram.htm>
10. <http://developer.com/design/article.php/2206791>
11. Balamurugan Shanmugam, Visalakshi Palaniswami, "Modified Partitioning Algorithm for Privacy Preservation in Microdata Publishing with Full Functional Dependencies", Australian Journal of Basic and Applied Sciences, 7(8): pp.316-323, July 2013
12. Balamurugan Shanmugam, Visalakshi Palaniswami, R. Santhya, R.S. Venkatesh "Strategies for Privacy Preserving Publishing of Functionally Dependent Sensitive Data: A State-of-the-Art-Survey", Australian Journal of Basic and Applied Sciences, 8(15) September 2014.
13. S. Balamurugan, P. Visalakshi, V.M. Prabhakaran, S. Chranayaa, S. Sankaranarayanan, "Strategies for Solving the NP-Hard Workflow Scheduling Problems in Cloud Computing Environments", Australian Journal of Basic and Applied Sciences, 8(15) October 2014.
14. Charanyaa, S., et. al., "A Survey on Attack Prevention and Handling Strategies in Graph Based Data Anonymization". International Journal of Advanced Research in Computer and Communication Engineering, 2(10): 5722-5728, 2013.

# International Journal of Innovative Research in Science, Engineering and Technology

(An ISO 3297: 2007 Certified Organization)

**Vol. 4, Issue 2, February 2015**

15. Charanyaa, S., et. al., Certain Investigations on Approaches for Protecting Graph Privacy in Data Anonymization. International Journal of Advanced Research in Computer and Communication Engineering, 1(8): 5722-5728, 2013.
16. Charanyaa, S., et. al., Proposing a Novel Synergized K-Degree L-Diversity T-Closeness Model for Graph Based Data Anonymization. International Journal of Innovative Research in Computer and Communication Engineering, 2(3): 3554-3561, 2014.
17. Charanyaa, S., et. al., Strategies for Knowledge Based Attack Detection in Graphical Data Anonymization. International Journal of Advanced Research in Computer and Communication Engineering, 3(2): 5722-5728, 2014.
18. Charanyaa, S., et. al., Term Frequency Based Sequence Generation Algorithm for Graph Based Data Anonymization International Journal of Innovative Research in Computer and Communication Engineering, 2(2): 3033-3040, 2014.
19. V.M.Prabhakaran, Prof.S.Balamurugan, S.Charanyaa, "Certain Investigations on Strategies for Protecting Medical Data in Cloud", International Journal of Innovative Research in Computer and Communication Engineering Vol 2, Issue 10, October 2014
20. V.M.Prabhakaran, Prof.S.Balamurugan, S.Charanyaa, "Investigations on Remote Virtual Machine to Secure Lifetime PHR in Cloud ", International Journal of Innovative Research in Computer and Communication Engineering Vol 2, Issue 10, October 2014
21. V.M.Prabhakaran, Prof.S.Balamurugan, S.Charanyaa, " Privacy Preserving Personal Health Care Data in Cloud" , International Advanced Research Journal in Science, Engineering and Technology Vol 1, Issue 2, October 2014
22. P.Andrew, J.Anish Kumar, R.Santhya, Prof.S.Balamurugan, S.Charanyaa, "Investigations on Evolution of Strategies to Preserve Privacy of Moving Data Objects" International Journal of Innovative Research in Computer and Communication Engineering, 2(2): 3033-3040, 2014.
23. P.Andrew, J.Anish Kumar, R.Santhya, Prof.S.Balamurugan, S.Charanyaa, " Certain Investigations on Securing Moving Data Objects" International Journal of Innovative Research in Computer and Communication Engineering, 2(2): 3033-3040, 2014.
24. P.Andrew, J.Anish Kumar, R.Santhya, Prof.S.Balamurugan, S.Charanyaa, " Survey on Approaches Developed for Preserving Privacy of Data Objects" International Advanced Research Journal in Science, Engineering and Technology Vol 1, Issue 2, October 2014
25. S.Jeevitha, R.Santhya, Prof.S.Balamurugan, S.Charanyaa, " Privacy Preserving Personal Health Care Data in Cloud" International Advanced Research Journal in Science, Engineering and Technology Vol 1, Issue 2, October 2014.
26. K.Deepika, P.Andrew, R.Santhya, S.Balamurugan, S.Charanyaa, "Investigations on Methods Evolved for Protecting Sensitive Data", International Advanced Research Journal in Science, Engineering and Technology Vol 1, Issue 4, December 2014.
27. K.Deepika, P.Andrew, R.Santhya, S.Balamurugan, S.Charanyaa, "A Survey on Approaches Developed for Data Anonymization", International Advanced Research Journal in Science, Engineering and Technology Vol 1, Issue 4, December 2014.
28. S.Balamurugan, S.Charanyaa, "Principles of Social Network Data Security" LAP Verlag, Germany, ISBN: 978-3-659-61207-7, 2014
29. S.Balamurugan, M.Sowmiya and S.Charanyaa, "Principles of Scheduling in Cloud Computing" Scholars' Press, Germany., ISBN: 978-3-639-66950-3, 2014
30. S.Balamurugan, S.Charanyaa, "Principles of Database Security" Scholars' Press, Germany, ISBN: 978-3-639-76030-9, 2014