Aspect Oriented Strategy to model the Examination Management Systems

P.Durga[1], S.Jeevitha[2], A.Poomalai[3], Prof.M.Sowmiya[4] and Prof.S.Balamurugan[5]

Department of IT, Kalaignar Karunanidhi Institute of Technology, Coimbatore, TamilNadu, India[1,2,3,4,5]

ABSTRACT: The paper proposes a set of diagrams based on object orientation principle which describes the functionalities from various perspectives. The diagrams depicts the functional, behavioral and structural aspects of examination management system. 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 smart phone database 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. Here special attention is paid in planning phase and the same is extended to the implementation phase of the paper.

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

I. INTRODUCTION

Software is very abstract and hard to visualise. A visual modelling language, such as UML, allows software to be visualised in multiple dimension, so that a computer system can be completely designed before construction work starts. Furthermore, UML can be used to produce several models at increasing levels of detail. The overall plan of the software can quickly and easily be defined at the start of the project with a high level model allowing for accurate estimation. Increasing levels of detail can then be added to each part of the software as it is constructed, until finally the software is developed.UML is applicable for both new developed and already existing systems. It is been wrongly understood that a older system should be Re-documented for the application of newer application into it. It also improves the quality of the software engineering staffs as most of the engineers look for the company modeled with UML. The Unified Modelling Language was created by Rational Software in order to provide a standard for software modelling languages an independent organisation.

   UML is a universal language because it can be applied in many areas of software development. It includes user-defined extensions so that it can be adapted for specific environments. Industrial standard extensions now available for business modelling and web-based applications development. The Use Case Gives the nature of modelling with UML ensuring that all levels of model trace back resembles elements of the original functional requirements. This traceability between models comes without the extra effort of creating and maintaining a 'traceability matrix' which can be a complex and time consuming job. As a result of this the required changes for the document can easily be made. With the help of this UML any requirement changes can be made without any intreruption to other requirements even the lines. If any changes were been effected then that can be easily been changed without any difficulties. In many design processes, the use case diagram is the first that designers will work with when starting a project. This diagram allows for the specification of high level user goals that the system must carry out. These goals are not necessarily tasks or actions, but can be more general required functionality of the system.
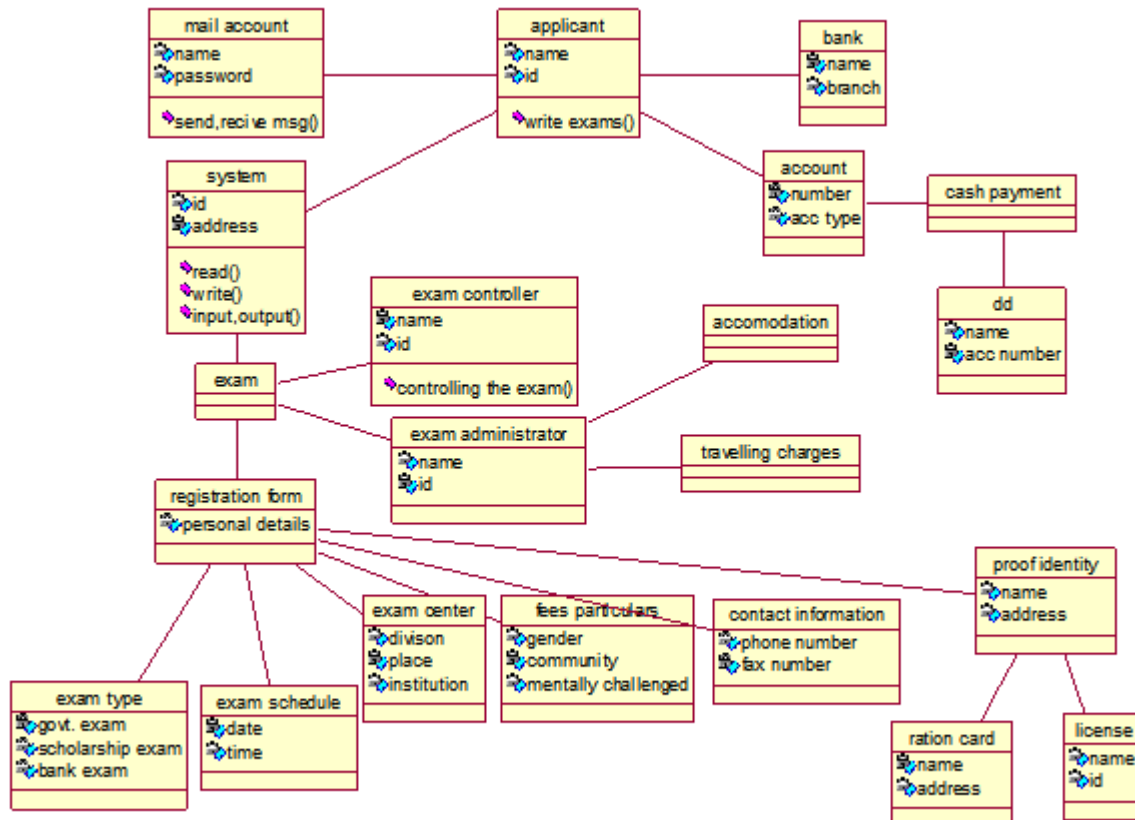
II. CLASS DIAGRAM

Fig 1 Class Diagram for Examination Management System

III. USE CASE DIAGRAM

UML Use Case Diagrams can be used to describe the functionality of a system in a horizontal way. That is rather than representing the details of individual features of an system. UCDs can be used to show all of its available functionality. It is should be noted that a USD is differernt from the flow chart, sequence diagrams because these diagrams won't explain the details about the next execution and sub-execution steps of the system.

The UCD have 4 major elements, the use case diagram depicit:

- **Use cases**. A use case describes a sequence of actions that provide something of measurable value to an actor and it is drawn as a horizontal ellipse.
- **Actors**. An actor is a person, organization, or external system that plays a role in one or more interactions with your system. Actors are drawn as stick figures
- **Associations**. Associations between actors and use cases are indicated in use case diagrams by solid lines. An association exists whenever an actor is involved with an interaction described by a use case. Associations are modeled as lines connecting use cases and actors to one another, with an optional arrowhead on one end of the line. The arrowhead is often used to indicating the direction of the initial invocation of the relationship or to indicate the primary actor within the use case. The arrowheads are typically confused with data flow and as a result I avoid their use.
- **System boundary boxes (optional)**. You can draw a rectangle around the use cases, called the system boundary box, to indicates the scope of your system. Anything within the box represents functionality that is in scope

and anything outside the box is not. System boundary boxes are rarely used, although on occasion I have used them to identify which use cases will be delivered in each major release of a system.

- **Packages (optional)**. Packages are UML constructs that enable you to organize model elements (such as use cases) into groups. Packages are depicted as file folders and can be used on any of the UML diagrams, including both use case diagrams and class diagrams. I use packages only when my diagrams become unwieldy, which generally implies they cannot be printed on a single Page.
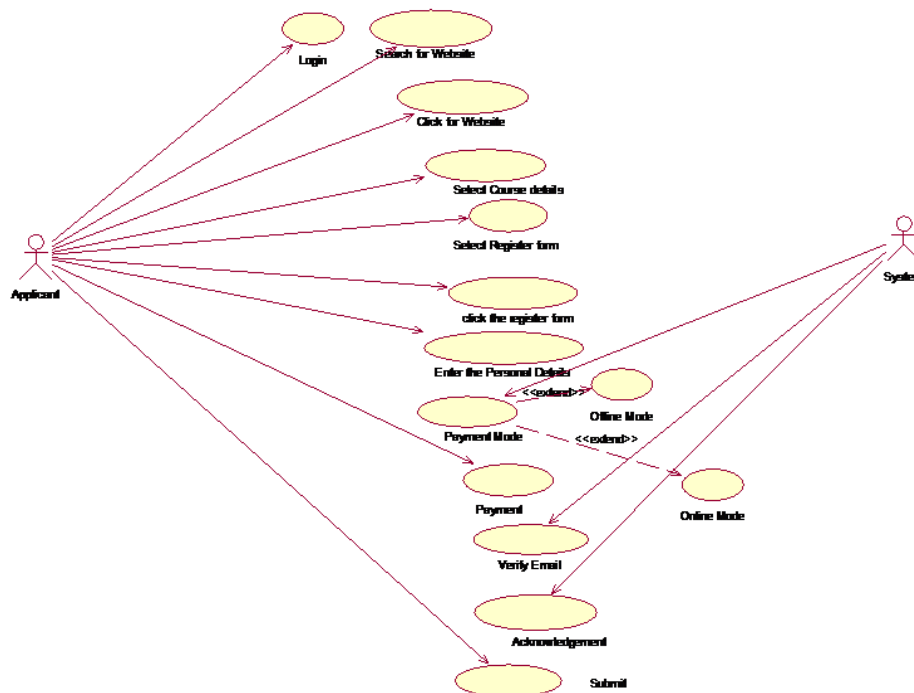

Fig2. Use Case Diagram for Examination Management Reservation System

IV. INTERACTION DIAGRAM

Interaction diagrams are models that describe how a group of objects collaborate in some behavior - typically a single use-case. The diagrams show a number of example objects and the messages that are passed between these objects within the use-case.

Interaction diagrams should be used when you want to look at the behavior of several objects within a single use case. They are good at showing the collaborations between the objects, they are not so good at precise definition of the behavior. If we want to look at the behavior of a single object across many use-cases then use of State Transition Diagram. If you want to look at behavior of many use cases or many threads, consider an activity diagram. Much like the class diagram, developers typically think sequence diagrams were meant exclusively for them. However, an organization's business staff can find sequence diagrams useful to communicate how the business currently works by showing how various business objects interact. During the requirements phase of a project, analysts can take use cases to the next level by providing a more formal level of refinement. When that occurs, use cases are often refined into one or more sequence diagrams.

- **Purpose**

The main purpose of a sequence diagram is to define event sequences that result in some desired outcome. The diagram conveys this information along the horizontal and vertical dimensions: the vertical dimension shows, top down, the time sequence of messages/calls as they occur, and the horizontal dimension shows, left to right, the object instances that the messages are sent to.

- **Lifelines**

When drawing a sequence diagram, lifeline notation elements are placed across the top of the diagram. Lifelines represent either roles or object instances that participate in the sequence being modeled. In fully modeled systems the objects (instances of classes) will also be modeled on a system's class diagram. Lifelines are drawn as a box with a dashed line descending from Messages.

- **Messages**

The first message of a sequence diagram always starts at the top and is typically located on the left side of the diagram for readability. Subsequent messages are then added to the diagram slightly lower than the previous message. At the center of the bottom edge the lifeline's name is placed inside the box.

- **Guards**

When modeling object interactions, there will be times when a condition must be met for a message to be sent to the object. Guards are used throughout UML diagrams to control flow. Here, I will discuss guards in both UML 1.x as well as UML 2.0. In UML 1.x, a guard could only be assigned to a single message. To draw a guard on a sequence diagram in UML 1.x, you placed the guard element above the message line being guarded and in front of the message name.

Except for the activity nodes the other notation elements of interaction overview diagrams are the same as for activity diagrams, such as initial, final, decision, merge, fork and join nodes. The two new elements in the interaction overview diagrams are the "interaction occurrences" and "interaction element. UML 2 has introduced significant improvements to the capabilities of sequence diagrams. Most of these improvements are based on the idea of interaction fragments which represent smaller pieces of an enclosing interaction. Multiple interaction fragments are combined to create a variety of combined fragments, which are then used to model interactions that include parallelism, conditional branches, optional interactions.
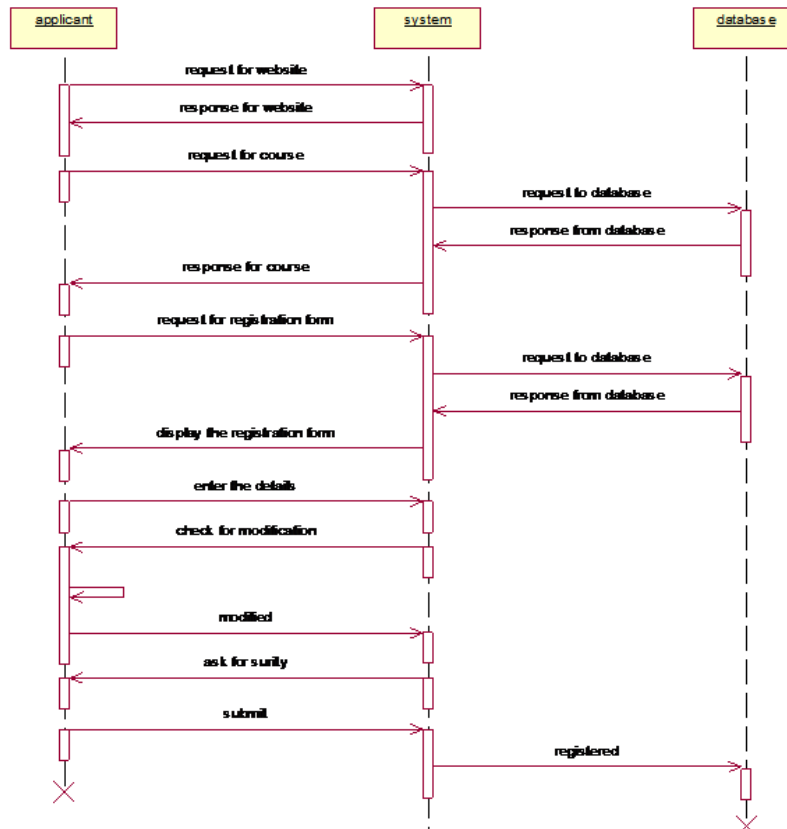

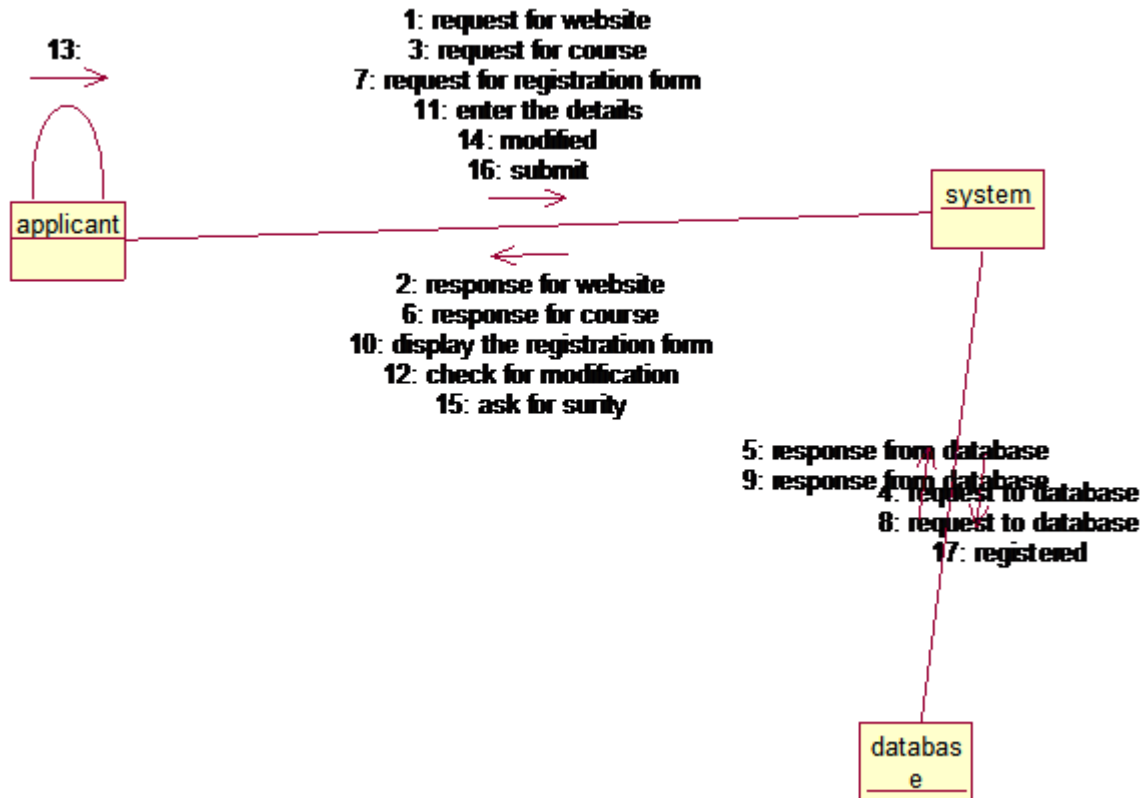Fig 3. Sequence Diagram for Examination Management Reservation System

Fig 4. Collaboration Diagram for Examination Management Reservation System

V. CONCLUSION AND FUTURE WORK

 The paper proposes a set of diagrams based on object orientation principle which describes the functionalities from various perspectives. The diagrams depicts the functional, behavioral and structural aspects of examination management system. 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 smart phone database 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. Here special attention is paid in planning phase and the same is extended to the implementation phase of the paper.

REFERENCES

1. B.Powmya,NikitaMaryAblett,V.MohanaPriya,S.Balamurugan,"An Object Oriented Approach to model the secure Healthcare Database system", In the proceedings of international conference on compter, 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 Lorensen "Object Oriented Modeling and Design", Prentice-Hall of India Private Limited.

International Journal of Innovative Research in Science, Engineering and Technology

(An ISO 3297: 2007 Certified Organization)

Vol. 4, Issue 2, February 2015

5. Grady Booch, "Object-Oriented Analysis & Design with Applications", Pearson Education
6. Rogert.S.Pressman, "Software Engineering:A Practioner'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
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.Chranyaa, 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.
15. Charanyaa, S., et. al., Certain Investigations on Approaches forProtecting 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, Decermber 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