iJRASET
International Journal For Research in Applied Science and Engineering Technology

INTERNATIONAL JOURNAL FOR RESEARCH

IN APPLIED SCIENCE & ENGINEERING TECHNOLOGY

Volume: 5 **Issue:** VII **Month of publication:** July 2017

DOI:

www.ijraset.com

Call: 08813907089 | E-mail ID: ijraset@gmail.com

International Journal for Research in Applied Science & Engineering Technology (IJRASET)
ISSN: 2321-9653; IC Value: 45.98; SJ Impact Factor: 6.887
Volume 5 Issue VII, July 2017- Available at www.ijraset.com

Modeling of Object Oriented Banking Database Using MDA

Shobhit Shukla[1,]
[1]*Assistant Professor, Department of Information Technology, Faculty of Computer Science and Information Technology, Dr. Shakuntala Misra National Rehabilitation University, Lucknow, India.*

Abstract: In the contemporary environment, the banking sector is altering swiftly. Numerous new services can be seen in banking sector as it is adopting new technology to advance their commercial interests and to win the client satisfaction. The banking system amass massive quantity of data which these applications are not fast enough to process and exploit it in an apt bearing. In this paper a novel design of a banking system has been suggested using the modern Model Driven Architecture (MDA) methodology of software engineering to improve the portability, robustness, and maintainability. This paper also focuses on the concept of Object Oriented Database concept which is used to design an object oriented database as an alternative of the traditional relational database which aids in better concurrency control and easier navigation.
Keywords: Model Driven Architecture (MDA); Object Oriented Database Management System (OODBMS); OTP; Computation Independent Model (CIM); Platform Independent Model (PIM); Platform Specific Model (PSM)

I. INTRODUCTION

A. Model Driven Architecture

Model Driven Architecture is a set of standards defined by the Object Management Group (OMG) which describes an approach which separates specification of system from the specification of the implementation of that system on a specific technology platform. MDA describes an architecture for models that offers a set of rules for organizing specifications stated as models [1]. MDA offers an approach for deriving value from models and architecture in support of the full life cycle of physical, organizational and I.T. systems [2]. The MDA methodology characterizes and supports various software engineering activities including requirements, modeling, implementation etc. By using MDA models, we complexity of large software systems can be managed easily and the interaction between organizations, users, engineers, hardware, software can be handled can be dealt with in a standard way [3]. MDA approach employs prevailing technologies, which support current and forthcoming OMG standards, to support model-driven development so that object models would become assets instead of overheads [4].

1) MDA can be defined as follows:

a) It is an OMG initiative to create standards based on the notion that modeling is a better foundation for developing and maintaining systems.

b) It is a benchmark for standards and systems that follow those standards.

c) It is set of tools and techniques related with those standards.

2) The various MDA technologies and standards which are available are:

a) The Meta Object Facility (MOF), which is a modeling language for describing metamodels.

b) XML Metadata Interchange (UML): It is a Definitive modeling language and an instance of the Meta Object Facility model.

c) Common Warehouse Metamodel (CWM): It is a set of modeling languages for data bundle applications. (E.g. Relational DBMS).

d) Queries/View/Transformations (QVT): Transformation definition language. It is also used for querying models.

e) XML Metadata Interchange (XMI): XMI 2.4 is a format to portray models in a structured text from. In this way UML models and Meta Object Facility (MOF) Metamodel may be swapped between various modeling tools.

The MDA Database Design Process is depicted in Fig. 1. The MDA separates the banking system database in the various level of abstraction. The first stage CIM (Computation Independent Model) represents requirements of the banking system which describes Business model. The second stage PIM (Platform Independent Model) represents the software specifications which describes the model of the system. The third stage PSM (Platform Specific Model) represents the software realization model which defines detailed

design of the system specific to the implementation platform. The end database design is implemented for an appropriate required platform from the PSM specification. In addition to the above, a model-driven approach also defines methods to integrate and update legacy systems according to new business obligations. This transformation is known as Architecture Driven Modernization (ADM) in the Object Management Group (OMG) [5].

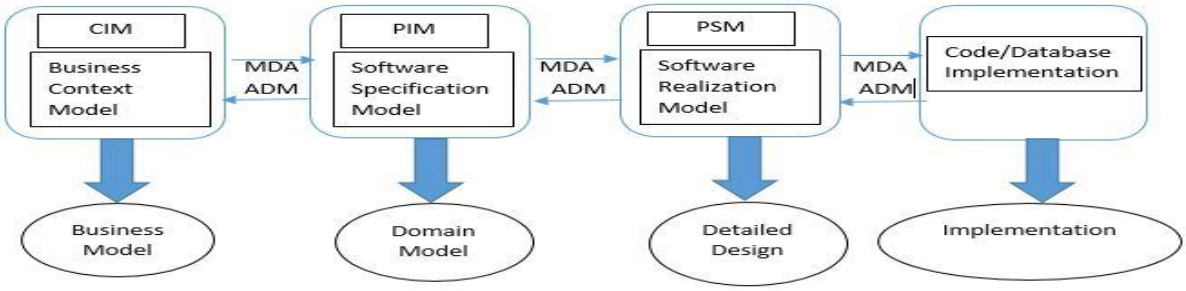


Figure 1: MDA Database Design Process

B. Object Oriented Database Management System

The progress in technology, especially the advent of internet, intranet and information technology has heralded a new paradigm for performing business transactions in banking sector. Traditionally, Relational Database Management Systems (RDBMS) formed the mainstay of dealing with central server and business transactions of Centralized Banking System (CBS). They also supported Disaster Recovery system (DRS) which used parallel queries and various data recovery techniques. In the present scenario, Object Oriented Database Management System (OODBMS) support complex queries and are able to handle large quantity of data more efficiently. An Object Oriented Database Management System is a database model which represents data in the form of objects as used in Object Oriented Programming (OOP). OODBMS are a result of combining database concepts with Object Oriented Programming techniques. An OODBMS is a database management system (DBMS) that represents data as objects and supports the modeling and creation of data as objects. They support the various OOP principles such as classes, encapsulation, inheritance, abstraction, inheritance and polymorphism.

An OODBMS must satisfy two criteria: it should be a DBMS, and it should be an Object Oriented system, i.e., it should be consistent with the existing object-oriented programming languages. The first condition characterizes five features: persistence, secondary storage management, concurrency, recovery and an ad hoc query facility. The second condition provides a set of eight features: complex objects, object identity, encapsulation, classes, inheritance, extensibility, overriding combined with late binding and computational completeness [6].

Information at present comprises of not only textual data but audio, video, graphs and photos which are considered as complex data types. Relational DBMS are not designed to support these complex data types. In the case of OODBMS, consistency can be maintained within one environment because both the OODBMS and the programming language will use the similar representation model [6]. On the other hand, Relational DBMS systems using complex data types would have to be separated into two separate subsystems: the database model and the application.

C. Comparison between OODBMS and Relational DBMS

The Relational DBMS differs from an Oriented Database Management System in that it does not provide the capability of modeling complex objects and of modeling inheritance relationships among various entities. A Relational DBMS does not provide tools to associate object behaviors with object definitions at schema level and does not provide mechanism to support the concept of Object Identity [8]. On the other hand, an OODBMS model supports modeling of complex objects and supports the concept of object identity. OODBMS also provide query language, allowing objects to be searched by a more declarative programming approach. It is in the area of object query language, that the main differences between products are found. An attempt at standardization was made by the ODMG with the Object Query Language, OQL. In case of OODBMS, access to data is faster as joins are not needed as opposed to tabular modeling of Relational DBMS. This is possible as an object can be retrieved in OODBMS directly without a search, by following pointers. Table 1 gives comparison between Relational DBMS and Object Oriented Database Management System [9].

S. No.	Functionality	RDBMS	OODBMS
1.	Objects	Support hierarchical complex data system.	Fetch as a singular entity.
2.	Composite data	Group similar rows of data in a tuple.	Support Classes
3.	Variable size data	Primitive data types and strings	Support user defined objects
4.	Support for legacy systems	No support for legacy systems	Supports for legacy and future system characteristics
5.	Comparable objects	No tool to model comparable representations	Supports diverse representations of the same object
6.	Schema assessment	No efficient support for schema evaluation	Support for intra class definition changes and structural changes to the class.
7.	Transactions	Returns the latest state possible.	Supports concurrent access and recovery

Table 1: Comparison between OODBMS and RDBMS

II. MDA IMPLEMENTATION OF BANKING SYSTEM DATABASE

Model Driven Architecture (MDA) consists of a collection of models, where every resulting model is developed on the basis on its preliminary model. MDA presents three types of models describing distinct abstraction levels. These are: Computation Independent Model (CIM), Platform Independent Model (PIM) and Platform Specific Model (PSM).

A. Computation Independent Model (CIM)

In MDA, a Computation Independent Model (CIM) is a model defined as a primary model. This model reveals system and software knowledge from the business perspective. The CIM may contain business knowledge about system organization, roles, functions, processes and activities, documentation, constraints etc. The CIM must contain business requirements for the software system. A CIM does not show details of the systems structure, but the overall system structure which is useful to understand the problem.

Fig. 2 shows the CIM of Banking System as a UML [9] Use Case diagram. It displays the various actors and the use cases. The actors are defined as the external users of the system and the use cases are defined as the functionality provided by the system to the actors. The use case diagram shown in Figure 2 contains 2 actors namely, Customer and Employee. A customer can open an account by accessing the Open Account functionality. The account can be either a Savings account or a Current Account. The Open Account use case includes the functionality of verifying the Permanent Account Number (PAN) and the Unique Identification Number (UID) which are uploaded by the user on the bank portal which are then cross checked with the government database. Once the verification is completed successfully, a KYC Confirmation Status message is given to the Customer through registered email and SMS on registered mobile number.

After opening an account, the Customer can deposit an amount in his account. The deposit can either be an ATM deposit or through the Cash counter in the Bank. The Employee accepts cash if the deposit is through the Cash counter in the Bank. The Customer can also perform transaction in his account. A Transaction can be either a Withdrawal or an Account Transfer. Withdrawal can be done either through an ATM or Cash Counter in the Bank. An Account Transfer can also be done either through an ATM, through cash counter in the bank or via online banking. In case of Cash counter or online banking, if the amount to be withdrawn or transferred is below 20,000 then the Employee will check the Signature of the account holder. If the amount to be withdrawn or transferred is between 20,000 and 100000 then the Employee will check the Signature as well as the Thumbprint of the account holder. If the amount to be withdrawn or transferred is above 100000 then the Employee will check the Signature, Thumbprint as well as the PAN of the account holder. I addition if the mode of transfer is online banking then an One Time Password (OTP) will be generated and sent to the registered mobile number of the Customer and the Customer has to enter this OTP in the transfer portal on the bank's website within a required amount of time period. A Customer can also close his account as well as apply for loan from the bank. The Employee will check the required and mandatory documents of the Customer and either qualify the Customer's loan or deny it.

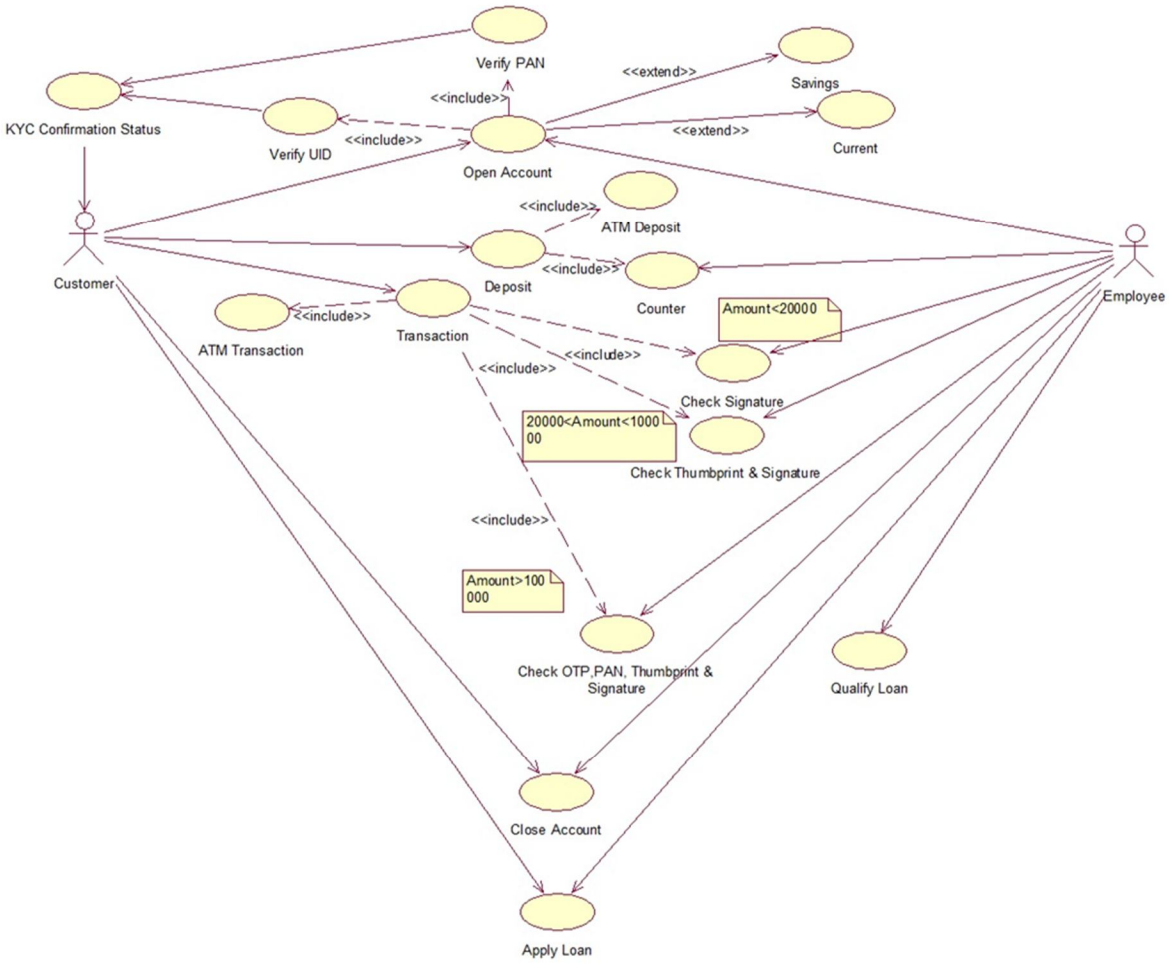


Figure 2: Use Case Diagram presented in CIM

B. Platform Independent Model (PIM)

A platform independent model of database is a view of the system from a viewpoint of platform independence. It hides detail design of database system. It is also representing platform independent information. It carries more detail than that of CIM. A Platform Independent Model (PIM) is a model of a software system which is independent of the specific technological platform used to implement it. PIM of Banking database [11] is described in the form of UML Class Diagram, State Chart Diagrams, Sequence Diagrams and Collaboration Diagrams.

1) Class Diagrams: Fig. 3 shows the Class Diagram of banking system in PIM. It consists of a class Person which has two subclasses Customer and Employee. A Person can open multiple accounts. An account can be either a Savings bank account or a Current bank account. A person can perform multiple transactions. An Account can have multiple transactions associated with it. An account can have only one UID and one PAN associated with it. A person can also apply for multiple loans.

2) State Chart Diagrams: Fig. 4 describes the State Chart Diagram for Open Account functionality. The Customer applies for new account by opening the Account opening form. The Customer then provides the registration details such as email id, mobile number, user name, login password, date of birth, address, account type etc. The Customer then uploads his UID, PAN details, Signature and Thumb Impression. The system then verifies the documents uploaded by the Customer and the registration details provided by him. After successful verification of the user details and documents, the new account is opened for the Customer and details provided to him and the system redirects to the homepage for login.

Figure 3: Class Diagram presented in PIM

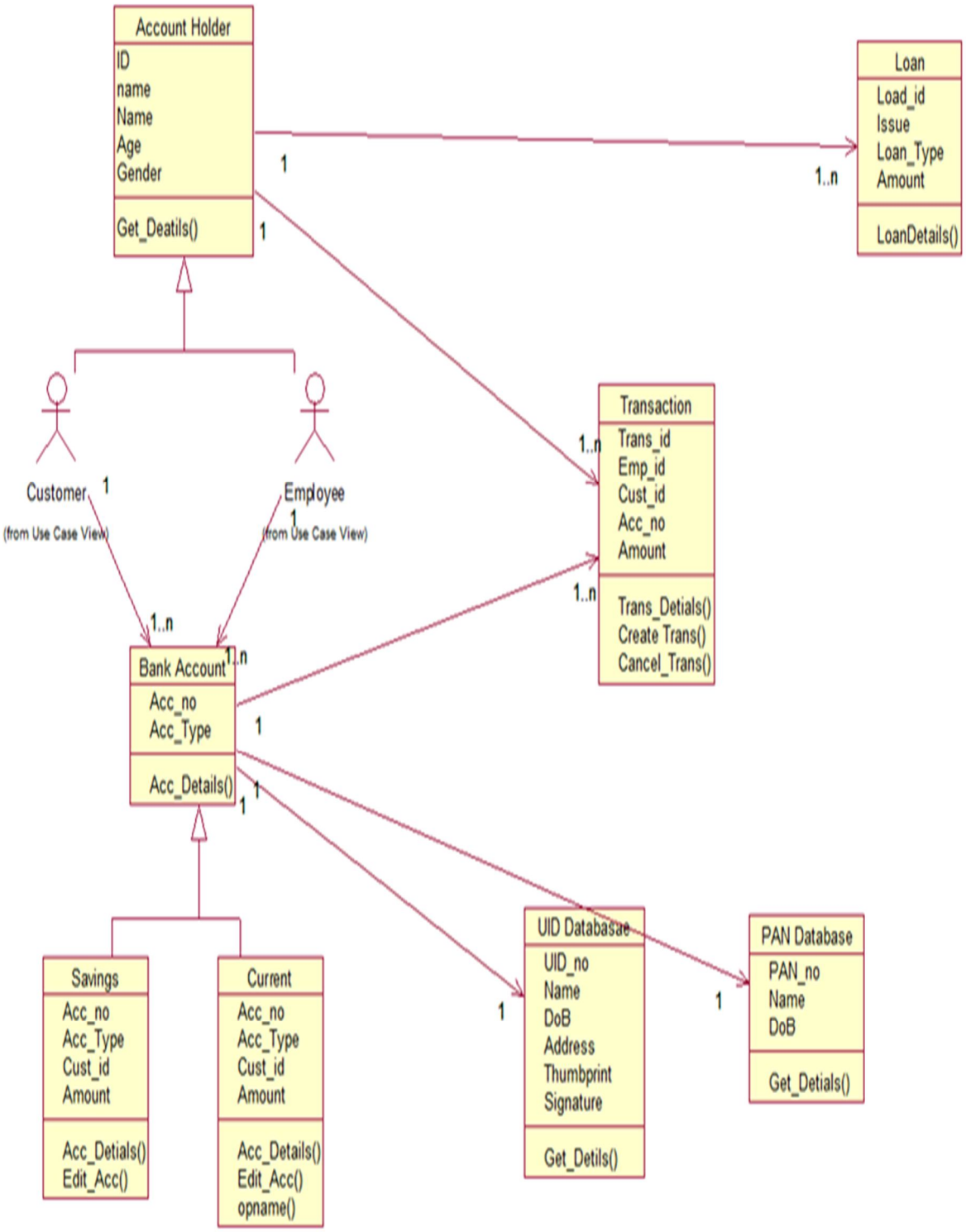


Figure 4: Open Account State Chart Diagram

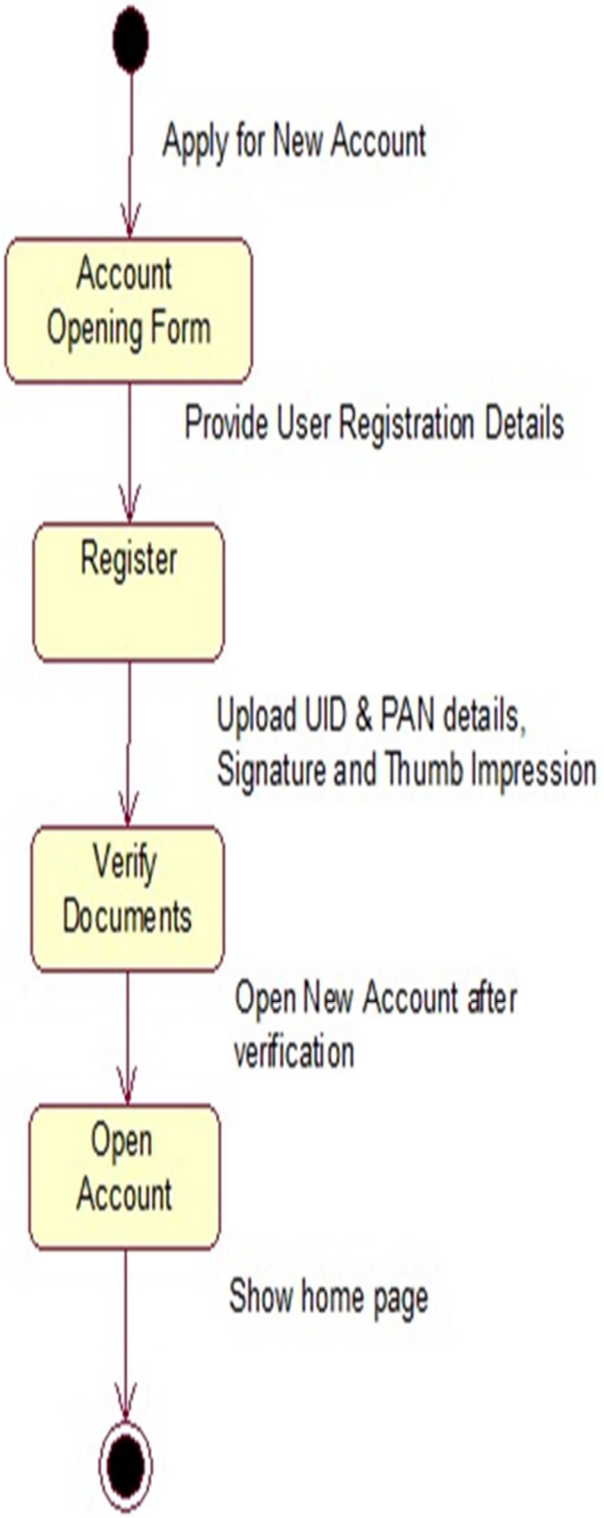


Fig. 5 describes the State Chart Diagram for Transaction functionality. A Customer first logins into his online account. He then selects the account from which the amount is to be debited and the amount which is to be debited and also provides the account number of the beneficiary. The system then checks if the selected account contains the sufficient amount so that the balance doesnot go negative after the transaction. If the sufficient amount is present then the various verifications are done dependingon the amount. After the transaction is finished successfully, the details are provided to the Customer and home page is displayed.

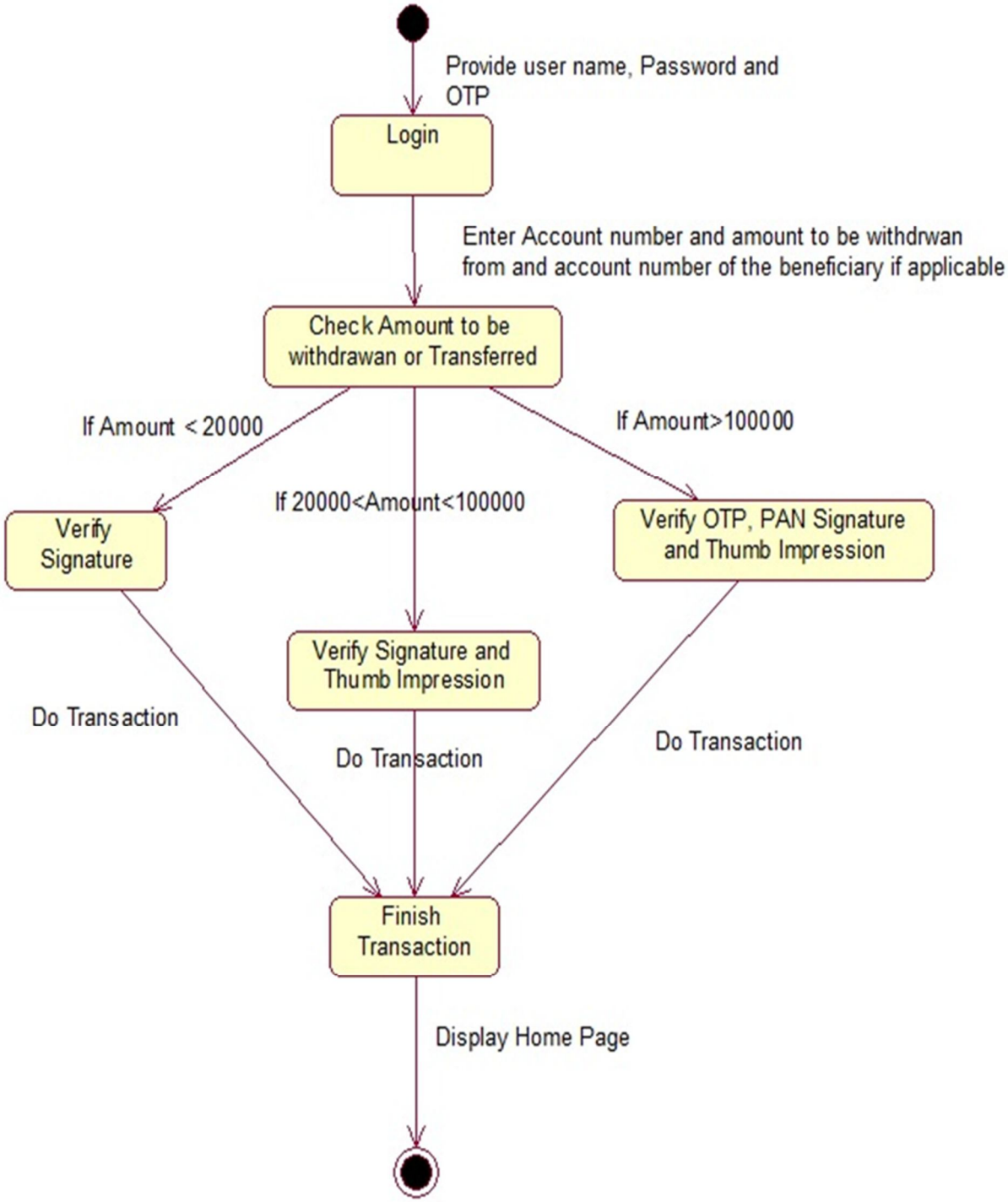


Figure 5: Transaction State Chart Diagram

3) Sequence Diagrams: A sequence diagram tells how various objects interact with each other. This diagram has two axes: The vertical axis shows time and the horizontal axis shows the objects. Fig. 6 describes the sequence diagram for Open Account operations in the Banking

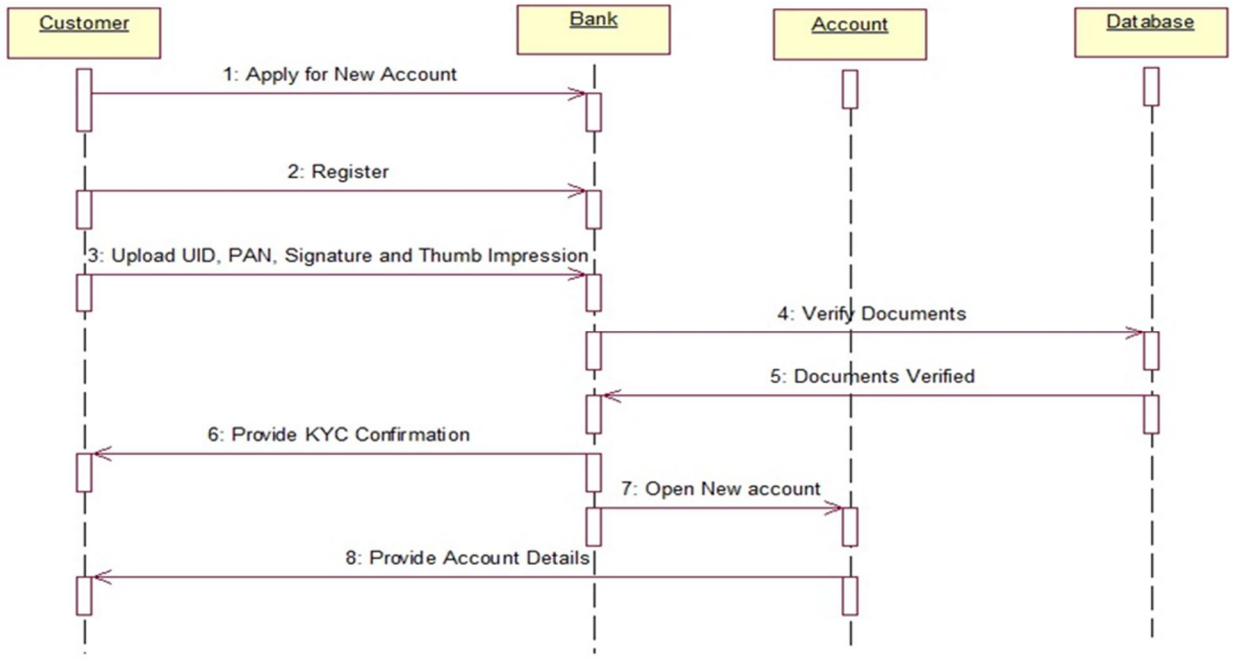
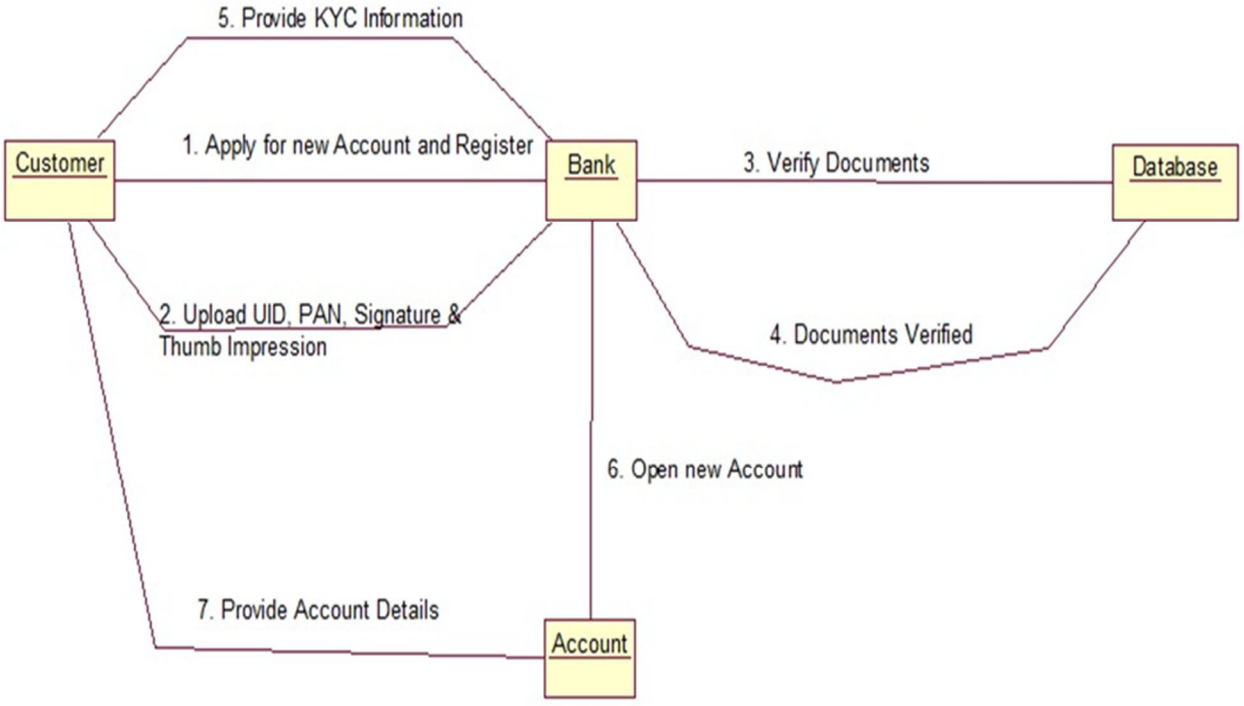


Figure 6: Open Account Sequence Diagram

Fig. 7 describes the sequence diagram for Transaction operations in the Banking System.

Figure 7: Transaction Sequence DiagramCollaboration Diagrams



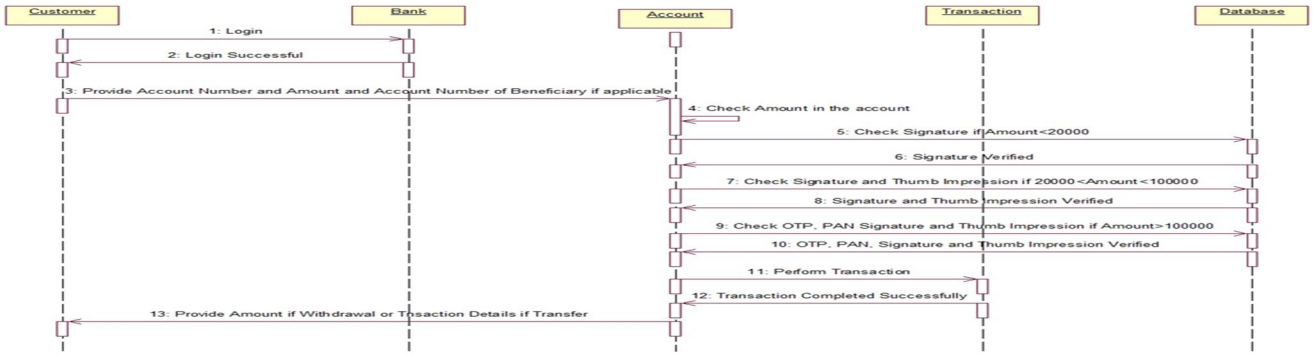
Collaboration diagrams convey the same information as sequence diagrams but they focus more on object roles in an overall system architecture. Fig. 8 shows the collaboration diagram for Open Account functionality.

C. Platform Independent Model (PIM)

A PSM combines the specifications in the PIM with the details required to specify how a system uses a specific kind of platform. An MDA mapping offers specifications for how to transform a PIM into a particular PSM [12]. The target platform model determines the nature of the mapping. The general pattern is:

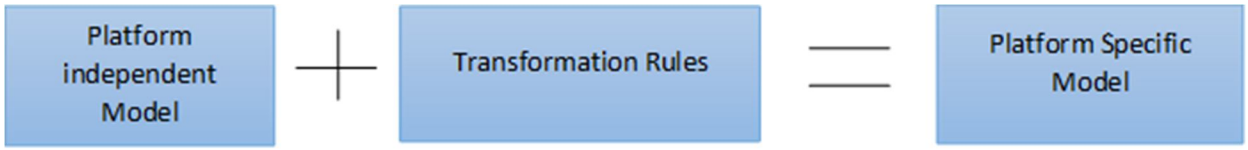
A model transformation mapping must be stated using some language which can be either a natural language or a dedicated mapping language. Two models PIM and PSM describe the same in MDA. To get a PSM from a PIM, different artifacts of the system are mapped from one model to another. Hence, it is essential to specify a set of transformation mappings that permits conversion of a source model e.g. PIM into the target model e.g. PSM. In the Figure 10 [11], PIM transformation into PSM using

Figure8: Transaction Collaboration Diagram



transformation mapping is shown. There are some transformation tools [12], which can perform model-to-model transformations and model to code transformation for given mapping rules.

Figure 9 describes the collaboration diagram for Transaction operations in the Banking System.



III. BIOMETRIC MATCHING

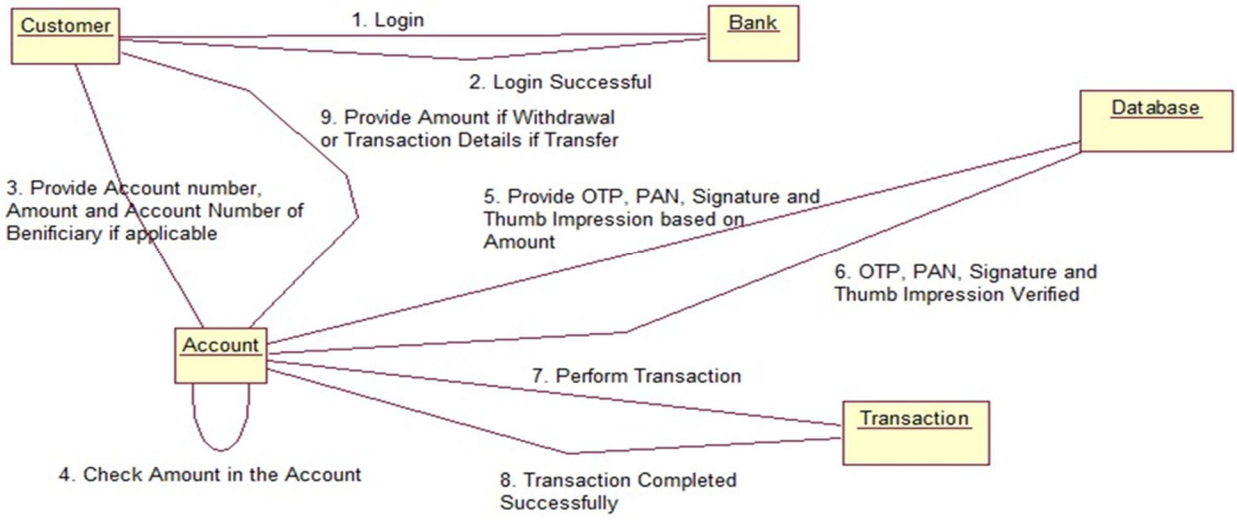
To implement the biometric matching of customers from the database we can employ various pattern matching algorithms using the concept of Artificial Intelligence. The main aim of Artificial Intelligence (AI) is to formulate the theory and develop computer systems able to perform tasks normally requiring human intelligence, such as visual perception, speech recognition, decision-making, and translation between languages. It is an academic field of study directed to create algorithms enabled by constraints exposed by representations that support models targeted at thinking, perception and action. Various Soft Computing based algorithms exist for pattern matching of biometric data. For Fingerprint Recognition mainly three techniques are used which are: a) correlation-based approaches, b) minutiae-based approaches and c) ridge feature-based approaches [13]. For Iris Recognition we can use Daugman algorithm [14], Boashash and Boles algorithm [15] based on zero-crossings or Multi-resolution Independent Component Identification (M-ICA) algorithm [16].

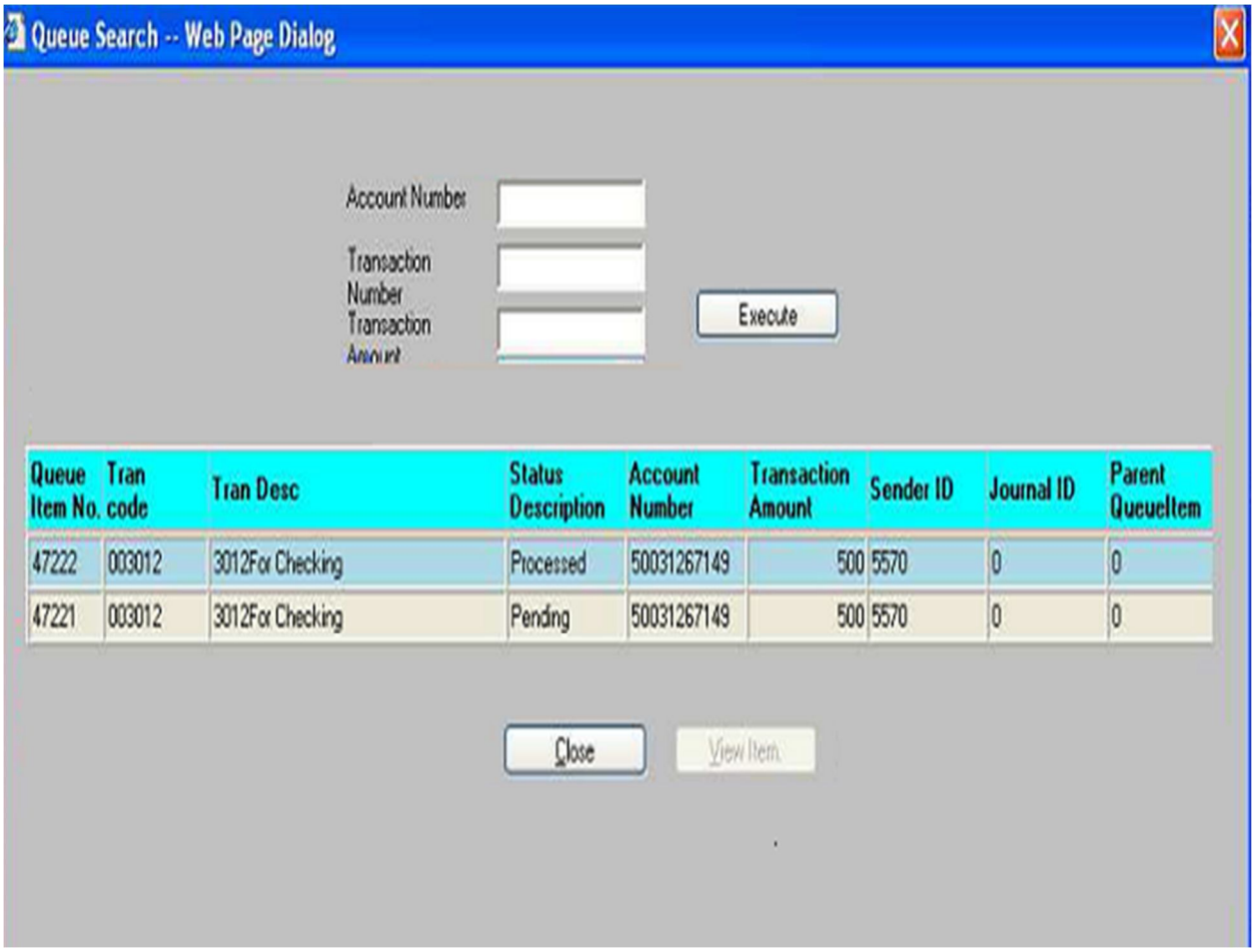
IV. EXPERIMENTAL RESULTS

The online banking transaction system using object oriented concepts and MDA has been implemented. The system uses an Intel Core i5 Processor with 3 GB DDR3 RAM. Windows 7 or higher version of operating system can be used and SQL server 2005 or

higher version can be used as a database. Rational Rose Software is used for the designing of MDA diagrams. The proposed system is implemented and the experimental results are shown in Figure 11:

Figure 11: Screenshot of Database Search Query





V. CONCLUSION

In this paper Model Driven Architecture has been applied on an Object Oriented database based banking systems. MDA provides system development a novel thought. Using MDA, system specifications can be represented in a way which separates the functionality from implementation thus it reacts promptly to the changing requirements. MDA also provides the benefit of faster development time as all the requirements of the system can be modeled at different abstraction levels using CIM, PIM and PSM before system development begins which also results in easier system validation and verification processes. Since the PIM produced is not system specific thus the same PIM model can be used to produce different PSMs for different platforms thus increasing design portability and making the system more robust to change in requirements. With the use of Object Oriented Database, the database system can perform faster operations and is also able to handle non-textual data easily. The model presented here also proposes the idea of a novel way of banking in which the customer does not have to physically visit a bank to perform account opening and transactions as he can upload his identification documents (UID, PAN, Thumb Impression and Signature) which can be verified using government databases thus ascertaining his identity.

VI. FUTURE SCOPE

The above work can be extended for the design of banking system for handling mobile banking and stock trading. It can also be used to model the system which can be useful for physically challenged people where they can use banking system without any hassle. The above system can also be extended by using the concept of clustering which can expedite the process of querying the database and performing transactions and also be useful in data mining.

REFERENCES

[1] M. Guttman and J. Parodi, Real-Life MDA: Solving Business Dilemma with Model Driven Architecture, San Francisco, CA: Morgan Kaufmann Publishers, 2007

[2] MDA Foundation Model, OMG document ormsc/10-09-06.

[3] OMG, "OMG Model Driven Architecture", Object Management Group (OMG). http://www.omg.org/mda.

[4] M. Asadi, M. Ravakhah, R. Ramsin,Second.2008. Asia International Conference on Modelling & Simulation “An MDAbased System Development Lifecycle”,IEEE..

[5] OMG, "OMG Model Driven Architecture", Object Management Group (OMG). www.omg.org/mda.

[6] Prabhu C. S. R. Object – oriented Database Systems (Approaches and Architectures), Third Edition, PHI Publication, 2012.

[7] Elisa Bertino , Lorenzo Martino, Object-Oriented Database Management Systems: Concepts and Issues, Computer, v.24 n.4, p.33-47, April 1991 [doi>10.1109/2.76261]

[8] Hurson A. R. And Pakzad H. Object Oriented Database Management System: Evaluation and Performance Issues, IEEE Publication, 1993.

[9] Rumbaugh, James, Ivar Jacobson, and Grady Booch (2005) The Unified Modeling.

[10] Singh, Y., Sood, M.2009. "Models and Revolutions in MDA," cicsyn, pp.253-258, 2009 First International Conference on Computational Intelligence, Communication Systems and Networks, IEEE computer society.

[11] Mishra, Suman Kumar et.al., “MDA based approach towards Design of Database for Online HealthCare System”, (IJCSIT) International Journal of Computer Science and Information Technologies, Vol. 5 (6) , 2014, 8182-8187.

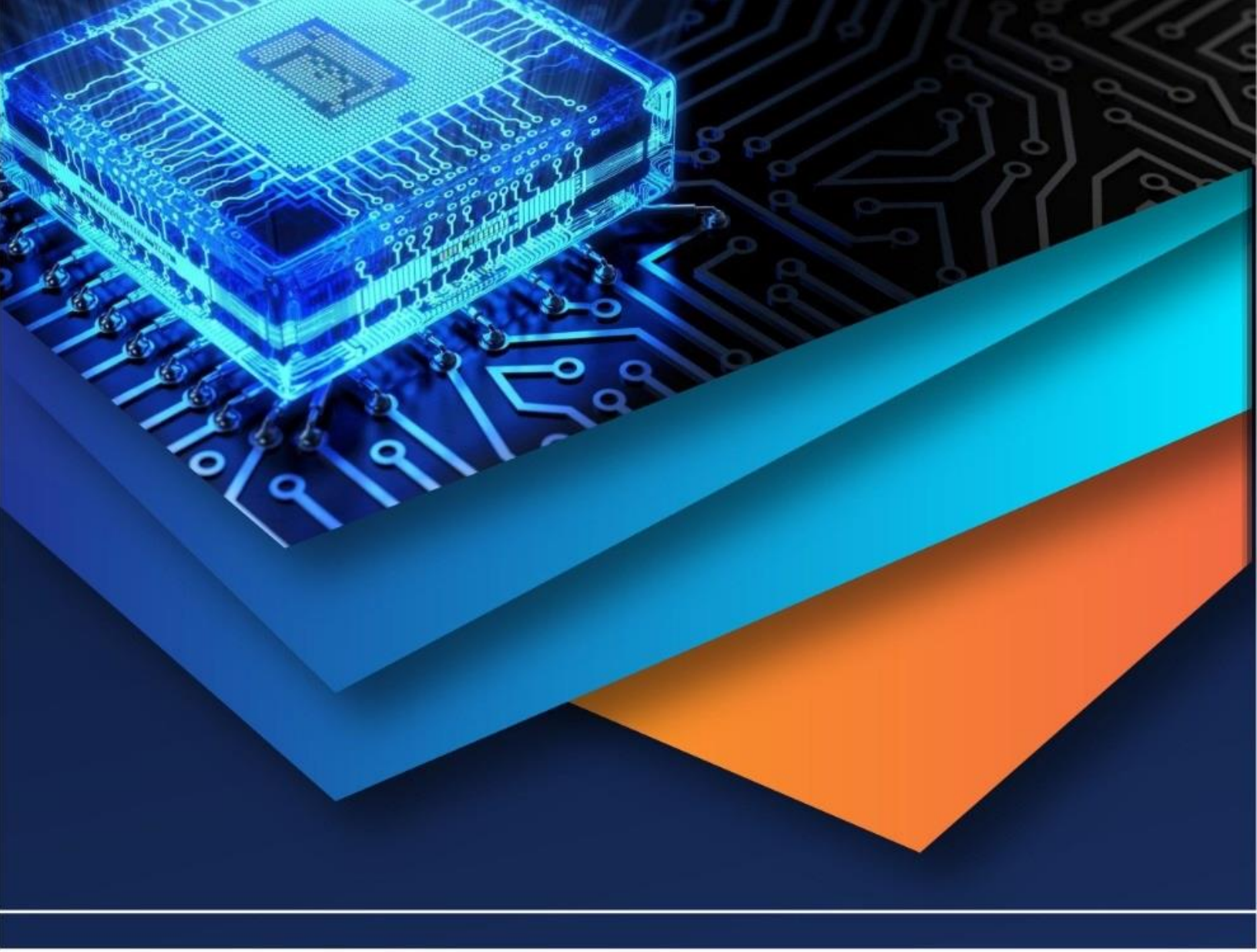
[12] ISO/IEC 9126-2:2002(E), Software engineering – Product quality –Part 2: External metrics.

[13] Maltoni, D., Maio, D., Jain, A., Prabhakar, S.: Handbook of Fingerprint Recognition. Springer, New York (2003).

[14] Daugman J. G., High coifidence visual recognition of persons by a test of statistical independence IEEE Transactions on Pattern Analysis and Machine Intelligence, Volume: 15, No. 1 I, (l993), pp. 1148-1 161.

[15] Boles W.W, Boashash B., A human identification technique using inzages of the iris and wavelet transform, IEEE Transactions on Signal Processing, Vol. 46, NO. 4, (1998), pp. 1185-1188.

[16] Seung-In Noh, Kwanghuk Pae, Chulhy Lee, Jaihie Kim, Multiresolution Independent Component Identification, Proceedings of the 2002 International Technical Conference on CircuitdSystems, Computers and Communications, Phuket, Thailand, (2002).



10.22214/IJRASET

45.98

IMPACT FACTOR: 7.129

IMPACT FACTOR: 7.429

INTERNATIONAL JOURNAL FOR RESEARCH

IN APPLIED SCIENCE & ENGINEERING TECHNOLOGY