

Improved Presentation and Facade Layer Operations for Software Engineering Projects

Dr. Yunus Doğan¹, Hasan Gezer² and Serdar Yılmaz³

¹Lecturer, Department of Computer Engineering, Dokuz Eylül University, TURKEY

²Student, Department of Computer Engineering, Dokuz Eylül University, TURKEY

³Student, Department of Computer Engineering, Dokuz Eylül University, TURKEY

¹Corresponding Author: yunus@cs.deu.edu.tr

ABSTRACT

Nowadays, one of the most challenging situations for software developers is the presence of a mismatch between relational database systems and programming codes. In the literature, this problem is defined as "impedance mismatch". This study is to develop a framework built on innovations based on the existing Object Relational Mapping technique to solve these problems. In the study, users can perform operations for three different database systems such as MsSQL, MySql and Oracle. In addition, these operations can be done within the framework of C# and Java programming languages. In this framework, while the developers can define database tables in the interface automatically, they can create relations between tables by defining a foreign key. When the system performs these operations, it creates tables, views, and stored procedures automatically. In addition, entity classes in C# and Java for tables and views, and operation classes for stored procedures are created automatically. The summary of the transactions can be taken as pdf file by the framework. In addition, the project can automatically create Windows Communication Foundation classes to facilitate the handling of database elements created and the interfacing operations, as well. This framework, which supports distributed systems, can be downloaded at this [link](#).

Keywords— Distributed Systems, Impedance Mismatch, N-Tier Architecture, Object Relational Mapping, Reverse Engineering

and also, their advantages and disadvantages have been researched. In this way, it has been determined how the framework should work and which features it should have.

In recent years, developments of software applications take much time because developers do repetitive operations. Developers need to create stored procedures such as insertion, deletion, updating etc. and to create views to join the tables with foreign keys. In addition, they need to create a connection between database elements and programming classes. They also need to create a class for each table and view in Entity Layer and a function for each stored procedure in Facade Layer. These operations take much time and become harder to manage when database components increase. N tier architecture can be a solution to this problem, because N tier architecture is useful in terms of creating an applied program, faster, more secure and strong-functioned according to [3]. Also, project standardization is easier with N tier architecture. As a result, it provides convenience to developers in terms of consistency and development; therefore, this study presents a solution by creating N tier architecture. Recently, software applications coordinate with a database, generally a relational one [4]. Data is stored in tables of tuples associated with each other, including attributes of values with defined types in a relational database [5][6]. With regard to [4], using objects rather than tables and tuples is much more convenient to access the database components in terms of object applications. However, one of the most challenges for software developers is the presence of a mismatch between relational paradigm and object paradigm.

In the study of Scott W. Ambler [7], this problem is defined as "impedance mismatch". Mapping objects is specified as a solution proposal in this study. In the study, Object Relational Mapping (ORM) is used as a solution for impedance mismatch problem. A connection has been provided between relational database components and programming language objects. ORM is defined in [8] as that a technique of bridging the gap between the object model and the relational model. Furthermore, ORM is simplified interaction between Object Oriented Programming (OOP) and Relational Databases (RDBs) [9].

I. INTRODUCTION

Reverse Engineering (RE) is a type of engineering which takes advantages and properties of the current created object. The main goal of RE is to create another object similar to the existing object. This technique can be used in different fields such as software, electronics etc. [1]. RE is useful to extract information from an existing piece of software or improve the value of it [2]. Also, RE can be used to understand the logic of the existing models and to find out the possible and necessary improvements. In this study, the approach of RE is used to examine existing tools and learn how they work. Popular tools like Microsoft Ado.net, Hibernate, EclipseLink etc. have been examined

According to [10], ORM layer is useful to automate the mapping process and to create an abstraction covers up technical details of the mapping application. In this proposed framework, relational database components are mapped to the objects by using ORM technique. The similar projects using ORM are still tried to improve. Some of the tools similar to the proposed framework are Microsoft ADO.Net Entity Framework, Hibernate, NHibernate, EclipseLink, Dataobjects.Net and Toplink.

In Table 1, some operations and properties of the popular frameworks which are appropriate for the project goal are placed into the first column. Also, operations and properties have been written for the tools. All frameworks in the table support three types of DBMS such as MsSQL,

TABLE I
COMPARISON BETWEEN THE PROJECT AND SIMILAR FRAMEWORKS IN TERMS OF OPERATIONS AND PROPERTIES

Operations/Properties	1	2	3	4	5	6
Supporting MsSQL, MySQL, Oracle	+	+	+	+	+	+
Creating Database	-	-	-	-	-	-
Creating Table	+	+	+	+	+	+
Creating View	-	-	-	-	-	-
Creating Stored Procedure	-	-	-	-	-	-
Creating Class for Table	+	-	-	-	-	+
Creating Class for Stored Procedure	+	-	-	-	-	+
Creating Class for View	+	-	-	-	-	+
Creating WCF Class	-	-	-	-	-	-

MySQL, and Oracle. However, none of them provides database creation. In Table 1, the 1st framework is Entity Framework, the 2nd is EclipseLink, the 3rd is Hiberbate, the 4th is Toplink, the 5th is NHibernate and the 6th is Dataobjects.Net.

Furthermore, database tables can be created in these frameworks. On the other hand, some database operations like to create view and to create stored procedures are supported by none of these frameworks. Some of these frameworks can be used to create Entity classes for the database tables, to create operation classes using stored procedures, and to create classes for the database views. For instance, Entity Framework and Dataobjects.Net are capable for these operations, but rest of them cannot be used for these goals.

One of these operations in the table is creating WCF classes. None of these frameworks can provide this operation. Lastly, these frameworks are supported different programming languages. For instance, Entity Framework, NHibernate, and Dataobjects.Net support C# language; EclipseLink, Hibernate, and Toplink support Java language. Also, Dataobjects.Net supports F# language as well.

The main goals of this study are to avoid from repetitive operations, to prevent complex database and object operations, to save project development time and to provide connection between relational database components and programming objects. For these objectives, N tier

architecture and ORM technique were used in the framework. This framework has been designed for three popular relational DBMSs such as Microsoft SQL Server, MySQL, and Oracle. In addition, two popular programming languages (Java and C#) are supported in the framework. The reason of choosing these DBMSs is that these systems are the three of the most popular DBMSs. With regard to [11], these DBMSs are on the top of the popularity ranking list. Oracle is the most popular system, MySQL is the second popular system and Microsoft SQL Server is in the third place. With regard to [12], the reason of preferring two programming languages (C# and Java) is that these languages are the two of the most popular programming languages. These languages are also convenient for object oriented programming and relational databases. In the list of [12], Java is the most popular programming language and C# is in the fourth place. In the framework, users can create new database connections or connect to the existing databases for three types of DBMSs separately. After connection, users can define database tables and views via the interface. The system creates tables, views, and stored procedures by only one request on the interface of the framework automatically. Furthermore, the system maps the database tables and the views to the Entity classes for selected programming language (Java or C#). Database stored procedures are also mapped to the classes in the Facade Layer by the system. It means that the system creates N tier architecture automatically by only defining a database table on the interface of the framework.

II. RELATED WORKS

In this study, a framework has been implemented by using ORM technique for processing. Therefore, previous researches by using ORM in the literature are reviewed in the following text. In Torres et al. [13], nine selected ORM solutions have been examined and compared. The goal of this survey is identifying and giving information about ORMs in order to help developers/users towards making better designing and implementing. Each ORM solutions have been assessed by using documentation and experience. Each ORM solutions have been compared in terms of coupling, mapping, identity, foreign keys, transparent association tables, and inheritance. According to this survey, JPA2 ORM framework can be called the best solution.

Lukyanchikov et al. [14] aimed at finding a solution for generating of the general principles of creating effective hybrid cloud systems. A technology system design must be selected to apply the principles of database. The design provides quick changes in the course of the experiments, automation for distributed hybrid cloud data. This research proposes ORM to provide these operations.

This solution has been chosen to provide a communication between relational data objects and developed software.

Cho and Kang [15] improved a framework of mobile application development tool using ORM solution in front-end domain layer. They have been tested the framework through mobile and web application projects to prove the usability and economic efficiency of the solution in mobile database-driven application development. Hule and Shaikh [16] built a tool which has been developed as a class library. They aimed to create a new project efficiently and usefully for developer like NHibernate, Hibernate. This project is based on ORM technique. The library project has been developed for C# programming language, so only C# classes and query methods have been created via an interface by using ORM technique.

Cvetković et al. [17] explained a tool which has been created for ORM specification based on NHibernate ORM framework. This tool has been created to automatically switch to provided database schema and domain class library to be matched. In addition, this tool is one of the schema mapping tools. It has been created for semi-automatic object-to-database mapping by using ORM solution and it has been designed for Geographic Information System (GIS). Blakeley et al. [18] researched to describe ADO.NET Entity Framework [19]. According to this research, the impedance mismatch problem and data services such as reporting, analysis can be reduced by using this platform and this platform can increase the level of abstraction from the relational to the conceptual level. This framework has been used as mapping engine to convert conceptual (entity) level to the logical (relational) level. According to this research, higher-level conceptual models based on entities and relationships need to be created instead of relational models. As a result, this research proposes ADO.NET Entity Framework to avoid from impedance mismatch problem and to reduce data services.

III. SYSTEM OVERVIEW

Figure 1 shows the relationship between the related component of the project such as Client, Server, Framework, and User machine.

The user can manage these operations by using the interface on the Client unit. Database operations are generated on the Framework unit. Also, C# and Java class operations are realized on the Framework unit. The request of file operations is sent to the User Machine unit according to selected programming language and selected DBMS. After that, related file operations are created on the User Machine unit. Furthermore, the request for the selected database type is sent to the server unit. Therefore, database operations are performed on the Server unit according to the selected DBMS After that, the server unit responds to

the Framework unit. Lastly, Framework unit sends result information to the Client unit to display.

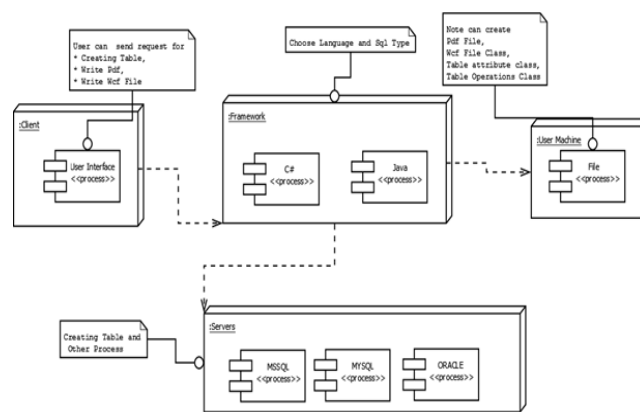


Figure 1: Deployment diagram of the system

IV. OUR APPROACH

After entering table attributes, the system manages some operations such as creating table, creating stored procedure, creating entity layer classes and creating facade layer classes. A class that is used to read related datagridview values is called by the system according to the selected sql type. The query strings to create table and to create five determined types of stored procedures to insert, delete, update, search, and select are defined. Each tuple element (attribute name and type) of the datagridview is concatenated to these defined strings. Further, if there is any defined relation between tables, a view query is created by the system. When the user wants to create log table, insertion query for log table is concatenated to every table stored procedure creation string to be able to record to the log table as well. In addition, the system generates five lists in the form of a class that has two parameters as attribute name and type for determined stored procedures.

When reading the datagridview values, attribute names and types are mapped to the type of the selected programming language. Getter and setter methods are created for Entity classes and attribute name and types are inserted to the lists created for stored procedures. After datagridview reading, the queries of creation table, view, and stored procedure are executed by the system. Furthermore, creating operation of Entity class is managed. If there is any view definition, then the system creates Entity class for the defined view as well. In addition, the system operates creating of facade layer classes by using defined five lists for the determined stored procedures.

A) Creating of entity layer classes:

Four different strings are generated named as header, namespace, content, and footer. Related libraries of

the selected programming language are concatenated to the header string.

In addition, namespace and class values taken by the user are appended to the namespace string. Further, the string content is used to keep attribute name and type taken by the user. These attribute types are converted to the programming language types and getter-setter methods for each attribute are concatenated to the content string. After that, closing parenthesis is appended to the footer string. Finally, this class structure is written as a file. File name is generated as "EntityClass_" plus entered table name. Figure 2 shows the flows of the sub-steps to generate the "EntityClass_".

B) Creating of operation classes:

Five different strings for queries of stored procedure call functions are defined besides header, namespace, content, footer to create methods in the operation classes (contentdelete, contentinsert, contentupdate, contentsearch, contentselect). The function of header, namespace, and footer strings is like the Entity class. On the other hand, content string is created by using five different methods. Firstly, attribute names and types are read from the datagridview and they are assigned to a generic list. In addition, five different generic lists are defined for each stored procedure.

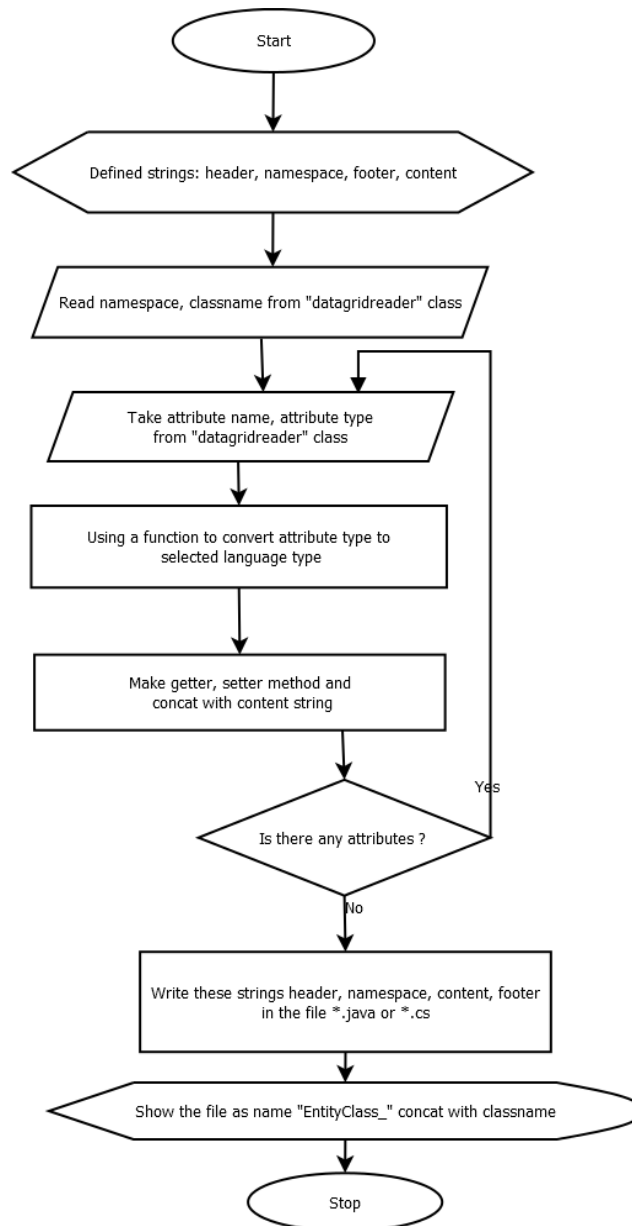


Figure 2: Entity class creation steps

Each list is sent to the related methods to be created call functions of stored procedures. In these methods, lists are searched and associated call functions of stored procedures are created by using list values according to the selected programming language and sql type syntax. After that, created structures are concatenated to the content string. Before concatenation operation, attribute types in the lists are converted to the programming language type. Finally, all strings are written as a file in the selected path by the user. Figure 3 shows the flows of the sub-steps to generate the “_OperationsClass”.

function names of stored procedures, entity class structure of tables and views, and closing parenthesis. Firstly, table, view, and stored procedure elements of the connected database are taken. These components are kept in the lists. Table and view lists are appended to a string to convert to the class structure. Further, all stored procedures in the database are kept in the lists and these elements are appended to the related string to create call functions of stored procedures.

Lastly, all strings are written as a file named as “IService.cs”.

As it is described, another WCF class named as “Service1.svc.cs” is created by the system. Some strings are defined to keep header, namespace, content, and closing parenthesis. Furthermore, content string is used to keep call functions of stored procedures for the selected database system. Attribute names and types of each stored procedure are collected in the lists and these lists are sent to the methods to create call functions of stored procedure. The syntax of them is determined according to the sql and programming language types. The generated structure is concatenated to the content string. Lastly, footer string is built up from closing parenthesis. Finally, all strings are written as a file named as “Service1.svc.cs” in the path selected by the user.

V. APPLICATION PROPERTIES

In the framework, there are some fields to take inputs such as sql type, programming language type, and package or namespace name at the start form. The user can select three different DBMSs such as MsSQL, MySQL, and Oracle via a list element as shown in Figure 4. Furthermore, the user can select two different programming language types like Java and C# over another list element. According to these inputs, different form fields can be displayed by the system for three different DBMSs separately. These forms have special input areas in terms of database creation or connecting to the database.

The user can create a new database by entering related fields like database name, server address etc. or can connect to the existing database by selecting any element in the database list. After clicking connect button, the system displays the main form for selected a type of DBMSs. In the main form, there are some buttons to create table, to delete table, to create pdf log report, to create pdf database log report, to create WCF classes, and to exit the system. In addition, there are two result areas in the main form. One of them is used to display process achievement and another one is used to display operation queries. If any error occurs, the user can get these query strings to correct them.

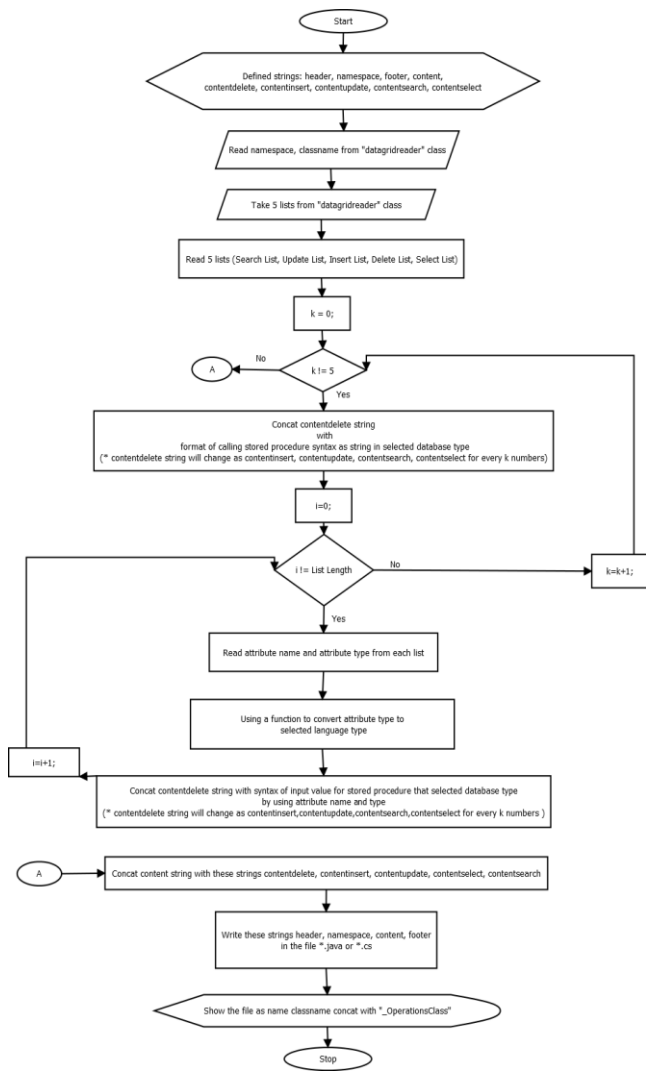


Figure 3: Operation class creation steps

C) Creating of operation classes:

Two WCF classes of “IService1.cs” and “Service1.svc.cs” are created by the system. Some strings are defined for “IService1.cs” class. These strings are used to keep libraries of class, namespace information, call

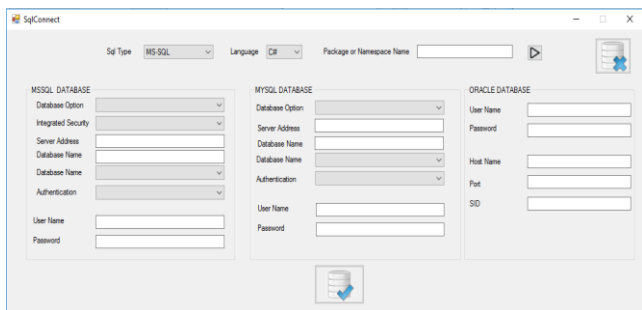


Figure 4: Database connection panel

As it is described, the main form is customized for the selected DBMS. Therefore, the datagridview component has special database elements for the selected database system. For example, in MsSQL datagridview, there is not any text area for attribute type size, but this element is placed in MySQL datagridview. The user can enter table components via the interface. There are two special columns in the datagridview. One of them is the “search for” column. The system creates “select” stored procedure by using table id parameter as default. If this column is checked, then the system inserts the checked attributes as parameter to the “select” stored procedure. Another special column is the “foreignkey” column. When the user checks this column, the system displays a window where there are two list components. In the first one, connected database tables are listed. The primary key of the selected table from the first list is filled in the second list by the system. Thus, the user can create a connection with another table in the database system. After this connection, a datagridview component is displayed in the top-right of the form. In this datagridview, foreign key and primary key values are displayed.

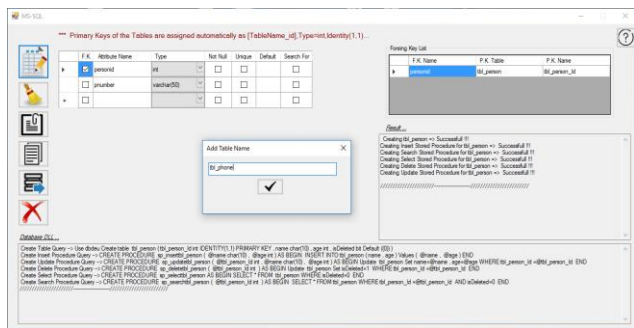


Figure 5: Table creation screen

When the user clicks on the “Create Table” button, the system asks whether the user has the log table request. If the user wants to create log table, then a form that has a datagridview component is displayed to define a user table. After creating user table, the system creates a log table by

using created user table. In log table, some columns are placed such as “userId”, “process date”, “user machine ip address”, and “process detail”. As a result, the user can create a database table by clicking “Create Table” button as shown in Figure 5, and create a log table as optional.

There are two types of log report creation buttons in the main form. One of them is used to create a pdf file that includes run time database operations. When the user clicks on this button, the system collects all operations that are managed by the user at the run time. The collected information is extracted as a pdf file into the determined path by the user. Furthermore, another button is used to create a pdf file that includes information of database components containing table names, attribute names and types of the tables, stored procedure names and their attributes, view names and their attributes etc. The collected information is extracted as a pdf file into the determined path by the user.

There is a button in the main form to create WCF classes. When the user clicks on this button, the system creates two WCF classes named as “IService1.cs” and “Service1.svc.cs”. These classes are created according to the connected database components. To manage this operation, all database elements are taken from the connected database system. After that, the structure of WCF classes is created by the system. The user can use these classes by importing into the existing WCF project.

Some test cases have been determined to control the result of the framework performance. These test cases were generated from basic functions of the framework. As it is described in previous chapters, the framework has been run for three popular DBMSs such as MsSQL, MySQL, and Oracle. Further, the system has been developed for two popular programming languages such as Java and C#. There are test results for test cases in Table 2. In some cases, there are star marks. If the case has three stars (***) , it means that the case is tested for three database types such as MySQL, MsSQL, and Oracle. If the case has two stars (**), it means that the case is tested for two different programming languages such as C# and Java. The framework has been tested for 10 software engineering projects being completed the requirement analysis phase. Furthermore, these projects at medical, academic and industrial areas have requirements with different sizes. As a result, it has been observed that the operations for each project have taken between 1 day and 5 days by only one person, and programmer-mistakes in the development process have been reduced approximately 100%. In addition, unlike other professional frameworks, database operations could be implemented as independent of types of DBMSs and operations in the facade layer could be implemented as independent of types of programming languages.

VI. CONCLUSION

As it is described, there are similar studies to the proposed framework such as Microsoft Ado.net, Hibernate, NHibernate, EclipseLink, Dataobjects.net, and Toplink. The functions and goals of these frameworks are similar to the proposed framework. On the other hand, our framework has lots of advantages upon other similar ones. Firstly, any of these frameworks does not work for Java and C# together. They are structured for either Java or C#. On the other hand, the framework is appropriate for these languages. Furthermore, these frameworks cannot be used to create database, view, and stored procedure. The framework is capable to realize these operations. In addition, these frameworks except Ado.net and Dataobjects.net are not used to create entity and operation classes. Furthermore, the framework serves some user optional operations like creation of WCF classes, creation of database and run time summaries. The framework is easy to use and also, database and object operations are fast and easy to manage, because all processes are automated. In summary, an approach in terms of ORM technique is proposed to solve "impedance mismatch" problem. Further, a framework is proposed that

TABLE II
RESULTS FOR THE TEST CASES.

Test Cases	Successful (for each of the 10 projects)
Creating MySQL, MsSQL or Oracle	+
Creating Table ***, View *** or Stored Procedure ***	+
Creating Log Table	+
Creating Log File	+
Creating Operation Class for Log Table **	+
Creating WCF File	+
Creating Database Connection Class **	+
Creating Entity Classes **	+
Creating Operation Classes **	+
Creating View Classes **	+
Creating PDF Summary Files ***	+

is appropriate for n-tier architecture to manage database and programming operations user-friendly in parallel. In the study of [20], it has been shown that using ORM tools increase the performance of the process for the implementation of a software application. As a result, the aim of the study has been to create a framework that has built on ORM technique.

The problem has been determined as that database operations become more complex, repetitive, and time-consuming processes as projects grow. A solution has been claimed to solve impedance mismatch problem between relational database systems and programming languages by using ORM technique. The similar tools and projects have been researched and determined their advantages and disadvantages versus our project. The requirements have

been decided and related functions have been defined. For the future works, this framework can be improved by addition of new DBMSs and programming languages for facade layer.

ACKNOWLEDGEMENT

This study was supported by Sema for Technology Limited Liability Company in Izmir in Turkey. We thank our colleagues from this company to give a chance to test our framework with the samples of their projects.

REFERENCES

- [1] Paulic M., Irgolic T., Balic J., Cus F., Cupar A., Brajlilj T., & Drstvensek I. (2014). Reverse engineering of parts with optical scanning and additive manufacturing. *Procedia Engineering*, 69, 795-803.
- [2] T.A. Nguyen. (2015). *Tools for program understanding and reverse-engineering of mobile applications*. Available at: https://rc.library.uta.edu/uta-ir/bitstream/handle/10106/26733/Nguyen_uta_2502D_1326_0.pdf?sequence=1.
- [3] Jing L.U. (2006). Comment on N tier architecture based on. net. *Computer Engineering and Design*, 27, 3467-3469.
- [4] Pluciennik-Psota E. & Pluciennik T. (2013). Using object database as a cache for relational database-performance and usage considerations. *Studia Informatica*, 34,145-60.
- [5] H. Garcia-Molina, J.D. Ullman, & J. Widom. (2000). *Database system implementation*. Upper Saddle River, New Jersey, USA: Prentice Hall.
- [6] J.D. Ullman & J. Widom. (2007). *A first course in database systems*. (3rd ed.). Upper Saddle River, New Jersey, USA: Prentice Hall.
- [7] Ambler S.W. (2000). *Mapping objects to relational databases*. Available at: <http://tiopf.sourceforge.net/Doc/Concepts/mappingObjects.pdf>.
- [8] M. Keith & M. Schincariol. (2013). *Introduction*. In: *Pro JPA 2*. California, USA: Apress, 1-14.
- [9] Alghamdi A., Owda M. & Crockett K. (2017). Natural language interface to relational database through object relational mapping. *Advances in Computational Intelligence Systems, Lancaster*, 449-464.
- [10] Lorenz M., Rudolph J.P., Hesse G., Uflacker M., & Plattner H. (2017). Object-relational mapping revisited-a quantitative study on the impact of database technology on O/R mapping strategies. *The 50th Hawaii International Conference on System Sciences, Hawaii*, 4877-4886.
- [11] Lieponienė J. & Šlenderis L. (2016). Computer sciences. *The 58th International Scientific Conference of Daugavpils University, Daugavpils*, 189-198.

- [12] Hamilton D. & Pape P. (2017). 20 Years after the Mandate. *CrossTalk*, 14-16.
- [13] Torres A., Galante R., Pimenta M.S., & Martins, A.J.B. (2017). Twenty years of object-relational mapping: A survey on patterns, solutions, and their implications on application design. *Information and Software Technology*, 82, 1-18.
- [14] Lukyanchikov O., Payain S., Pluzhnik E., & Nikulchev E. (2014). Using object-relational mapping to create the distributed databases in a hybrid cloud infrastructure. *IJACSA: International Journal of Advanced Computer Science and Applications*, 5, 61-64.
- [15] Cho J., Kang H., & Kim S. (2015). A mobile application development tool based on object relational mapping solution. *Indian Journal of Science and Technology*, 8, 1-5.
- [16] Hule K. & Shaikh Z. (2014). Object relational mapping tool for c#.net framework. *IJIRSET: International Journal of Innovative Research in Science, Engineering and Technology*, 3, 15185-15191.
- [17] Cvetković S.S., Bogdanović M.D. & Stoimenov L.V. (2011). NHibernate Mapper – A tool for rapid development of data access layer for interoperable GIS solutions. *Faculty of Electrical Engineering University of Banja Luka*, 15, 62-66.
- [18] Blakeley A.J., Campbell D., Muralidhar S., & Nori A. (2006). The ADO.NET entity framework: Making conceptual level real. *ACM SIGMOD Record*, 35, 31-38.
- [19] Papa J. (2007). Data points- ADO.NET entity framework overview. *MSDN Magazine-Louisville*, 25-34.
- [20] Borys M. & Panczyk B. (2015). Improving data processing performance for web applications using object-relational mapping. *Actual Problems in Economics*, 1, 387-395.