

Object Oriented Business Architecture on Online-exam and Assignment system

Teymur Haji-zada

Submitted to the
Institute of Graduate Studies and Research
in partial fulfillment of the requirements for the Degree of

Master of Science
in
Computer Engineering

Eastern Mediterranean University
June 2013
Gazimağusa, North Cyprus

Approval of the Institute of Graduate Studies and Research

Prof. Dr. Elvan Yılmaz
Director

I certify that this thesis satisfies the requirements as a thesis for the degree of Master of Science in Computer Engineering.

Assoc. Prof. Dr. Muhammed Salamah
Chair, Department of Computer Engineering

We certify that we have read this thesis and that in our opinion it is fully adequate in scope and quality as a thesis for the degree of Master of Science in Computer Engineering.

Prof. Dr. Erden Başar
Supervisor

Examining Committee

1. Prof. Dr. Erden Başar

2. Prof. Dr. Marifi Güler

3. Asst. Prof. Dr. Ahmet Ünveren

ABSTRACT

Business object architecture is a technology that was designed and developed during recent period. This architecture has a lot of benefits like scalability, flexibility and security. It helps create and develop maintainable, secure and reusable applications for further development. In business object architecture the logical architecture is separated into layers that give more scalability and reusability. Also using business object architecture developers must not write different programs for different platforms. Once created business objects can be accessible from different platforms.

The goal of this thesis is implementing computer assisted exam system with business object architecture. The object is to create desktop, web and mobile applications that will use the same business objects for doing operations.

The framework that is used is CSLA.NET. The library created with this framework is used by desktop, web and mobile application. For client applications only user interface is developed which uses business objects from the library. SQL Server is used for creating database. SQL Server and File server are in separate server that is accessible from all client applications. The network connection between client computers during exam is controlled by firewall. It disables all incoming and outgoing connections to the computer for holding the exam in safe environment. The implementation of the system is realized by using Microsoft .NET technology. As the development environment Microsoft Visual C# .NET is used. iPhone and iPad applications are part of the system. These applications are designed for students getting information about their grades by

mobile phones and notifications about exams or assignments from instructors. As mobile application iOS application is developed using Objective-C programming language.

Keywords: Business object architecture, computer assisted exam system, assignment system, iOS course manager application

ÖZ

İşletme Nesne Mimarisi Teknolojisi'nin icat edilmesi ve geliştirilmesi dönemi uzun bir geçmişe dayanmamaktadır. Ölçülenebilirlik, esneklik ve güvenlik gibi birçok avantajı bulunmaktadır. Bu mimari geliştirilebilir, güvenli ve yeniden kullanılabilir uygulamaların yaratılıp geliştirilmesine yardımcı olur. Daha fazla ölçülenebilirlik ve yeniden kullanılabilirlik sağlamak amacı ile Nesne İşletme Mimarisi'nin mantıksal mimarisi bir takım katmanlara ayrılmıştır. İşletme Nesne Mimarisini kullanan geliştiriciler çeşitli platformlar için program yazabilirler ve bu programlar çeşitli platformlardan erişilebilmektedir.

Bu tezin hedefi Nesne İşletme Mimarisini kullanarak bilgisayar destekli sınav sistemi geliştirmektir. Projede bir takım işlemleri yapmak için aynı işletme objeleri kullanan masaüstü, web ve mobil uygulamalar geliştirmektir.

CSLA.NET kullanılarak yaratılan kütüphaneler masaüstü, web ve mobil uygulamalar tarafından kullanılmıştır. Kullanıcılar için geliştirilmiş olan ara yüz bu kütüphanenin işletme objelerini kullanmaktadır. Veritabanı yönetim sistemi olarak SQL Server kullanılmıştır. SQL Server ve Dosya Sunucusu farklı sunucular olup kullanıcı uygulamaları tarafından erişilebilmektedir. Sınav süresince bilgisayar arasındaki ağ bağlantısı güvenlik duvarı tarafından kontrol edilmektedir. Güvenli bir ortam oluşturmak için güvenlik duvarı sayesinde gelen ve giden bağlantılar devre dışı bırakılmıştır. Bu sistemin geliştirilmesi Microsoft .NET teknoloji ile yapılmıştır. Microsoft Visual C#

.NET kullanılmıř, bununla birlikte, iOS mobil uygulamaların geliřtirilmesi için Objective-C programlama dili tercih edilmiřtir.

Anahtar Kelimeler: Nesne iřletme mimarisi, bilgisayar destekli sınav sistemi, ödev sistemi, iOS ders yöneticisi uygulaması

ACKNOWLEDGMENTS

First of all I would like thank to my supervisor Prof. Dr. Erden Başar for his support, encouragement and great guidance during the work on this thesis.

I grate to my very close friend Evelina Kudaikulova for being beside me, encouraging me and for support.

Finally, I would like thank to my parents Aygun and Kamil Hajiyev and my brother Tahir Hajizada for uninterrupted support during all my life. Special thanks to my grandmother Shargiya Aliyeva for motivating me all time.

TABLE OF CONTENTS

ABSTRACT	iii
ÖZ	v
ACKNOWLEDGMENTS	vii
LIST OF TABLES	xi
LIST OF FIGURES	xiii
1 INTRODUCTION.....	1
1.1 Background	1
1.2 Online Exam Systems.....	1
1.3 Aim and Objective.....	4
2 OBJECT-ORIENTED BUSINESS ARCHITECTURE THEORITICAL BACKGROUND	7
2.1 What is Object-oriented Business Architecture?.....	7
2.2 CSLA .NET Framework.....	7
2.3 Logical and Physical Architecture.....	8
2.4 Logical Architecture of Object-oriented Business Application	9
2.5 Advantages of Business Objects	11
3 PROJECT ANALYSIS	13
3.1 Project Scope.....	13
3.2 Project Functions	13
3.2.1 Instructor’s Functions	13
3.2.2 Student’s Functions	16
3.3 Project Requirements.....	18

3.3.1 Software Requirements.....	18
3.3.2 Hardware Requirements	19
3.4 Structure of the Conventional System and Proposes System.....	19
3.5 Database Design	23
3.5.1 Conceptual Design of the System.....	23
3.5.1.1 Entity Relationship Diagrams.....	24
3.5.1.2 Entities of the System	31
3.5.1.3 Relationship of the E.R. Diagram.....	33
3.5.2 Normalization	35
4 DESIGN AND IMPLEMENTATION.....	37
4.1 Design Issues of the Database Server.....	37
4.2 System Architecture	38
4.3 Flow-chart Design of the System	41
4.3.1 Student – Exam system Flow Chart.....	41
4.3.2 Student – Assignment System Flow Chart.....	42
4.3.3 Student – Mobile Application Flow Chart.....	43
4.4 Implementation.....	45
4.4.1 Server	45
4.4.2 Database Server	46
4.4.2.1 Tables.....	47
4.4.2.2 Usage of Tables in Implementations	52
4.4.3 Client Applications Implementation.....	52
4.4.4 Mobile Application Implementation.....	53
4.5 Client Side Design with Business Objects	58
5 CONCLUSION	61

REFERENCES.....64

APPENDICES67

 Appendix A: User Documentation for Exam Instructor’s Program.....68

 Appendix B: User Documentation for Exam Client Application.....85

 Appendix C: User Documentation for Assignment Submission System90

 Appendix D: User Documentation for iOS Application95

 Appendix E: Business Methods Region Source Code97

 Appendix F: Validation Rules Region Source Code.....98

 Appendix G: Factory Methods Region Source Code.....99

 Appendix H: Data Access Region Source Code100

LIST OF TABLES

Table 1. Compare Features of Online-exam Systems.....	3
Table 2. The Five Logical Layers and Their Roles.....	11
Table 3. Student Entity with its Attributes.....	31
Table 4. Course Entity with its Attributes.....	31
Table 5. Exam Entity with its Attributes.....	31
Table 6. Grade Entity and its Attributes.....	32
Table 7. Notification Entity with its Attributes.....	32
Table 8. Assignment Entity with its Attributes.....	32
Table 9. Laboratory Entity with its Attributes	33
Table 10. Laboratory Computers Entity with its Attributes.....	33
Table 11. Relationship between Student and Course.....	33
Table 12. Relationship between Student and Exam.....	33
Table 13. Relationship between Exam and Course.....	34
Table 14. Relationship between Student and Grade	34
Table 15. Relationship between Student and Notification.....	34
Table 16. Relationship between Assignment and Course.....	34
Table 17. Relationship between Student and Assignment.....	35
Table 18. Relationship between Exam and Laboratory	35
Table 19. Relationship between Laboratory and Computers	35
Table 20. Students Table.....	47
Table 21. Courses Table.....	47
Table 22. Student Course Table.....	48

Table 23. Assignments Table.....	48
Table 24. Assignment Submission Table.....	48
Table 25. Exams Table.....	49
Table 26. Exam Laboratories Table.....	50
Table 27. Exam Laboratory IPs Table	50
Table 28. Students' Temporary Exam Passwords Table	50
Table 29. Grades Table	51
Table 30. Notifications Table.....	51

LIST OF FIGURES

Figure 1. The 5-layer Logical Architecture.....	9
Figure 2. Connections Structure of the Previous System.....	20
Figure 3. Connections Structure in Proposed System.....	21
Figure 4. Structure of Different Platforms in Conventional Systems.....	21
Figure 5. Structure of Different Platforms that Use Business Objects	22
Figure 6. Relationship between Student Entity and Course Entity.....	24
Figure 7. Relationship between Student and Exam Entities	25
Figure 8. Relationship between Exam and Course Entities	25
Figure 9. Relationship between Student and Grade Entities.....	26
Figure 10. Relationship between Student and Notification Entities	27
Figure 11. Relationship between Assignment and Course Entities	27
Figure 12. Relationship between Student and Assignment Entities	28
Figure 13. Relationship between Exam and Laboratory Entities.....	29
Figure14. Relationship between Laboratory and Computer	29
Figure 15. Complete ER Diagram.....	30
Figure 16. Data Flow Diagram of the System	38
Figure 17. Communication between Mobile App and Server by Web Service	41
Figure 18. Student – Exam System Flow Chart.....	42
Figure 19. Student – Assignment System Flow Chart	43
Figure 20. Student – Mobile Application Flow Chart	44
Figure 21. Tables and Relations of the Database.....	46
Figure 22. iTunes Connect Login Page.....	54

Figure 23. iTunes Connect Main Page.....	55
Figure 24. Course Manager Application Managing Page on iTunes Connect.....	55
Figure 25. Course Manager Application in App Store	56
Figure 26. iPhone Application Storyboard	57
Figure 27. iPad Application Storyboard.....	57
Figure 28. ExamPro Setup Page	68
Figure 29. ExamPro Selecting Installation Folder	69
Figure 30. ExamPro Installation Process	69
Figure 31. ExamPro Configuration File.....	70
Figure 32. ExamPro Default Window.....	71
Figure 33. ExamPro Courses Window.....	72
Figure 34. ExamPro Students Window.....	73
Figure 35. ExamPro Laboratories Window	74
Figure 36. ExamPro FTP Settings Window	75
Figure 37. ExamPro Exam Definition Form.....	75
Figure 38. Exam Definition Form for Selected Exam	76
Figure 39. Students' Passwords for the Exam	77
Figure 40. Exam Control Form.....	78
Figure 41. Exam Control Form with List of Clients and Their Statuses.....	78
Figure 42. Exam Evaluation Form.....	79
Figure 43. Exam Evaluation Form with Grades.....	80
Figure 44. Exam Evaluation - Export Grades Form	80
Figure 45. Assignment Control Form	81
Figure 46. Assignment Control Form – Editing	82
Figure 47. Assignment Evaluation Form	83

Figure 49. Mobile Notifications Form	84
Figure 50. Exam Client Program Configuration File.....	85
Figure 51. ExamPro Launcher Program.....	86
Figure 52. ExamPro Client Login Window	87
Figure 53. ExamPro Client Main Form	88
Figure 54. ExamPro Client – Add Files	88
Figure 55. ExamPro Client Application with Files.....	89
Figure 56. ExamPro Client Result Form.....	89
Figure 57. Assignment Submission System – Login Page	90
Figure 58. Assignment Submission System – Assignments List Page.....	91
Figure 59. Assignment Submission System – Control and Submit Page	91
Figure 60. Assignment Submission Form – Submission Result Page	92
Figure 61. Assignment Submission System – System Prevent.....	93
Figure 62. Assignment Submission System – Password Change Page	94
Figure 63. iOS Application Greeting Page and Authentication.....	95
Figure 64. iOS Application Course Page.....	96
Figure 65. iOS app – Exam and Assignment Page	96
Figure 66. iOS Application Notification Page.....	96

Chapter 1

INTRODUCTION

1.1 Background

Computer assisted examination systems nowadays are being practiced frequently in most universities and educational centers. Because, it allows to test the performance of student accurately. Computer assisted examination systems are mostly used for programming, designing and for other courses that needs computer applications for studying and testing.

Online assignment system allows students to submit their assignments or home works without visiting professor. Submitting assignment by the internet, easy evaluation for professors are the advantages of the system.

The aim of the thesis is to create such application that will be scalable, enable high performance and secure. So, object-oriented business architecture is the one of the most suitable solutions.

1.2 Online Exam Systems

Every year the number of online exam systems increases. They are developing day by day. Developers try to add more functionalities, security fixes and etc. Before beginning working on the thesis such systems were analyzed. The results of the analyses every

system has pros and cons. After collecting the list of lacks of other systems there was developed a new system that differs from others. Here is the list of online exam systems:

Yardstick Academic Exam Solutions for Schools [5] is one of the famous services which have the functionalities creating exams, questions; deliver with the students and making reports. The question can be single and multiple choices. Students answer the questions by choosing the answers from the variants and after finishing the exam get their result.

Secureexam [6] is another type of exam system. It blocks all the functions of operating system and network and does not allow leaving the program until exam finishes. This software is good for essay type exams. Secureexam enables a text field for the student for answering each question and submit the results.

ExamSoft [7] is another tool for making exams. It allows make exams both in laboratory and online from any place. When the exam starts on exam-taker side, program downloads the exam files to client computer, then blocks operating system and disables using any of the commands of O.S. When the exam finishes, program uploads the client files to server and the computer comes to normal state. It is good for single and multiple choice test exams and essay type exams.

Also there are exam systems like Examkiosk [8], Exam Builder [9], ProPofs Quiz Maker [10] which are good for single and multiple choice exams. The features of these software solutions are shown in Table 1.

Table 1. Compare Features of Online-exam Systems

	Yardstick	SecureExam	ExamSoft	ExamKiosk, ExamBuilder, ProPofs	Developed software (ExamPro)
OS Support	Windows XP, Vista, Seven	Windows XP, Vista, Seven	Windows XP, Vista, Seven; Mac OS X	Windows XP, Vista, Seven	Windows Seven
Accessible on web	✓	✓	–	✓	–
Offline version	–	–	✓	–	–
Accessible on laboratory	–	✓	✓	–	✓
OS block during exam	✓	–	✓	–	–
Network block during exam	–	–	–	–	✓
Download questions	–	–	–	–	✓
Upload answers	–	–	–	–	✓
Courses and student data	–	–	✓	–	✓
Exam Results Report	✓	✓	✓	✓	✓
Publish results on web	–	–	–	✓	–
Send results to Iphone/Ipad	–	–	–	✓	✓
Built-in Assignment System	–	–	–	–	✓
Submit	–	–	–	–	✓

assignments from web					
Home Exams	–	–	✓	✓	–
Laboratory exams	✓	✓	✓	–	✓
Exam style	Single, multiple choice, self answer	Essay	Single, multiple choice, Essay	Single, multiple choice	Using computer programs

Existing exam systems are not suitable for computer assisted exams such as programming exams in computer laboratory since these programs are designed for essay or quiz type exams. Manually controlling such exams has difficulties like collecting answers on removable disks [16].

1.3 Aim and Objective

The aim of this thesis is to create online-exam and assignment system based on business object architecture. The programs listed in previous chapter enable single, multiple choice tests, essay type exams but none of them was planned for the exams that need particular application or tool for holding exam. The program developed in this thesis let instructors make exam on computer laboratory. Exam-takers can use the pre-installed programs for solving exam questions. This program was developed for programming, design and other courses that need computer programs for studying and passing exam. It lets instructors to get information about student's performance; helps understand how they learned the course and the level of students more accurately than exams on paper.

The security of the program is very important. Program must prevent students copying from each other or from the internet. So the program is designed in such a way that when the exam starts first program must download the exam files and questions, after it must block all local and public network connections. Students cannot access both internet and each other. The only connection is the one between server and students. This connection is used for downloading exam files from the server to student's computer and uploading the answer to the servers by students. Also program prevents using any USB memory sticks and external hard drives. If student will try use any USB memory it will not available to open the drive.

Besides the copying, students must sit on the places that have been defined by the system before the exam. This prevent students agree before the exam and sit in such order to help each other. When the exam places are defined program sends the place and time information about the exam to students who have iPhone or iPad. Also there is opportunity for instructors printing the students with their sit place list before the exam. If the student has not such devices s/he can learn it during exam.

The exam period is defined before exam starts. When the time is up students' program will automatically close. Computer comes to normal state after restarting.

Built-in assignment system was designed to get easy submit assignments by students and evaluation process for instructors. Students used to submit the assignments by sending the solutions to the instructor's email or by visiting the instructor in his office. Now

there is a web-based assignment submission system which allows reading the conditions and the assignment description published by instructor.

Totally designed system combines the following programs for thesis:

- **Student program** is for students, which is used during exam to get exam question and submit the results in secure environment.
- **Instructor program** designed for instructors to define, make and evaluate exams and assignments, manage courses and students, configure laboratory settings, send notifications to iPhone and Ipad devices.
- **Server side.** The server consists of FTP server, MS SQL Server and IIS server. All the information is kept in database and all student files are in FTP server. IIS serves for web-based assignment submission system.
- **iPhone and iPad applications** are designed for students to get notifications from instructors and learn the grades of exams and assignments.

Chapter 2

OBJECT-ORIENTED BUSINESS ARCHITECTURE

THEORITICAL BACKGROUND

2.1 What is Object-oriented Business Architecture?

Technically, business objects combines conventional lower-level objects that implement a business process. User interfaces can be thought of as views of large-grained business objects. Databases maintain a record of the "state" of Business Objects as they change over time [2].

Compared with library-bases technology, object-oriented business architecture increases the reusability of the design and scalability, makes short the further development cycle [3].

2.2 CSLA .NET Framework

CSLA .NET framework is a powerful software development framework that helps creating scalable, maintainable business logic architecture. While other frameworks focus on data manipulation and user interface, CSLA .NET enables building and developing object-oriented business applications. The main goal of this framework is abstracting business logic within object-oriented business layer. The code pattern of CSLA .NET enables to separate the application into logical layers [4].

In order to get CSLA .NET features in the system the general classes must inherit CSLA base class. The class relies on “code region” concept. These code regions realize development pipelining and allow programmers to develop system in accordance with respective duties. The system which is developed by using CSLA .NET has clear and stable logic structure. Separating the system into logical layers by regions makes further development cycle more accurate [18].

2.3 Logical and Physical Architecture

Before beginning the definition of architecture layers it is important to define logical and physical architecture description.

N-tier applications means physical model that define different functionalities between multiple machines. These can be a client, a web server, an application server, a database server, a file server, and so on [1].

Logical n-layer architecture is different from physical n-tier architecture. Logical n-layer architecture does not interfere to work of machines, network, and etc. Rather, logical architecture defines the functionalities of the application by separating them into layers as the most commonly used interface layer, business layer and data layer [1].

Especially in big systems it is very important clearly separate these layers. This separation makes application more maintainable, scalable and flexible. Logical layer must be designed in such a way that any modification in one of the layers or in physical architecture does not impact the layers. Properly designed logical n-layer architecture

has the benefits as easier maintenance, better reuse of code and team development experience. On the other hand properly designed physical n-tier architecture can provide better performance, scalability and security [1].

The detailed explanation of logical architecture is described below.

2.4 Logical Architecture of Object-oriented Business Application

In the implementation, 5-layer logical architecture is used as object-oriented concept.

This is shown in Figure 1 [1].

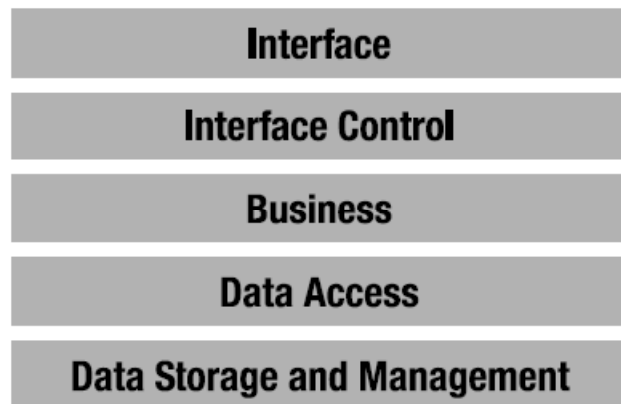


Figure 1. The 5-layer Logical Architecture

The main advantage of object-oriented business architecture is structuring the project with logical layers. Such designing gives us easy development of the project, performance increasing, scalability and security [17]. The definition of the layer is given below:

Interface layer in this thesis is consists of Windows Forms and Web Forms. This layer performs the operations like displaying data and collecting user inputs.

Interface control layer is separated from interface layer. Because the job of interface layer is getting data from user side and showing the result message to him/her. Separate interface control layer gives us to do more complex controlling of user data before sending it to business layer. Actually the main role of interface control layer is performing transition between interface and business layers [1].

Business layer is the most important part, because it is the only point of maintainability. All data validations, business and authorization rules are in business layer. It is very necessary to separate business layer from interface control layer because the business logic written into windows form application will not be available for web application and the services used in the system [1].

Data access is the layer between business layer and the database which is associates with data storage and management layer to perform select, insert, update and delete records operations. Designing data access layer and database management layer separately gives us divide the implementation into physical layers in the future. It means, the database management side can be in separate server [1].

Database storage and management: SQL Server, Oracle or other database management systems can be used as database storage and management layer [1].

Summary of the layers and their roles are shown in Table 2.

Table 2. The Five Logical Layers and Their Roles

Layer	Roles
Interface	Produces user interface and collects user inputs.
Interface Control	It is intermediate layer between the user and business logic that provides taken user inputs to the business layer and shows the result to the user.
Business Logic	Provides all business rules, validation, manipulation, processing and security for application.
Data Access	It is intermediate layer between business and data layers that contains all information about data access, database and data structures.
Data Storage and Management	Creates, retrieves, updates and deletes data physically in a data store.

2.5 Advantages of Business Objects

Development of big nonbusiness object-oriented projects sometimes is very difficult, especially if changes are done by different developers. One of the main advantages of business objects is easy understanding and developing without time waste. The main purpose of choosing business object concept for this thesis is the advantages over the other technologies and frameworks which are listed below:

- **Scalability:** Business Object Architecture supports scalability better than conventional systems. A small system can be scaled up to large systems with the large number of functionalities and users [2].

- **Reusability:** Reaching reusability on early object-oriented technologies is difficult, because searching through a class library and to understand the details of code can take a lot of time. Practice shows that sometimes such experiments end with rewriting the classes from scratch. Business object architecture solves this problem. Changes at the code are automatically synchronized and objects can be reused by modifying characteristics through inheritance [2].

Chapter 3

PROJECT ANALYSIS

3.1 Project Scope

The scope of the thesis is to design and develop application based on business objects architecture. The developed application is online-exam and assignment system. On this implementation instructors can easily manage courses, students, assignments and make exams in secure laboratory environment.

3.2 Project Functions

There are two parts of the system. The first part is for instructors and the second is for students. Instructors have a lot of functions for managing exam process and they will be described below. Students have functions as login and submitting exam answers. The functions of both side are described below.

3.2.1 Instructor's Functions

- Manage courses:

One of the first operations instructor must do is adding course. While adding new course, instructor defines course name, course code and semester. Instructor can see previous added courses in the list and s/he is able to edit and remove them.

- Manage students:

After course definition instructor can add students. For adding students first the course must be selected. The following student information must be defined while adding new one: student id, student name and surname, taken course. Also instructor has opportunity load students list from CSV file.

- Configure laboratories:

Laboratory must be defined before making any exam. Instructor can create a laboratory and add computers by IP address to the list. In laboratory computers list instructor can see all the computers with IP address and numbers (computer id).

- Set FTP settings:

The program is developed in such a way that instructor's computer does not receive anything from student's computers directly. So for configuring FTP server's settings, there is a menu for instructor. One can define ftp host, user name, and password as ftp information.

- Notifications:

Notifications menu let instructor send any news or information like exam place and time, covered topics, grades or any announcement to students. Students who has iPhone or iPad device can receive the messages that instructor sent. Also there is a list of all notifications that was sent before. There is a delete selected notifications functions which removes the notification from the list and the notification page of iOS application.

- Assignment control:

Assignments part is one of the main functions of the system. Instructor can define new assignment under the course by setting assignment name, end date and time, description. Also s/he can attach up to five additional files to the assignment. Also there is a list of all assignments ordered by end date which are available for editing of each field and deleting.

- Assignment evaluation:

Assignment evaluation part makes easy for instructors to evaluate the assignments. Instructor gets the list of students that have submitted the assignment with the submit time by selecting assignment from the list for evaluation. Instructor can easily look at the uploaded files direct from the program window and set grades for every criteria or question that was asked in the assignment description.

- Exam definition:

Exam definition part is the first part that must be ready by instructor before exam starts. Instructor creates new exam by choosing course name from list and filling the following fields: exam name, exam file folder and exam question's file. Instructor can easily change the settings of predefined exam. There is a function in this part which generates exam passwords for students and randomly chooses their place in laboratory. Instructor can print this list and automatically send notifications to student's iPhone or iPad device. Delete function is also available for instructor to remove exam.

- Exam control:

For the complete management of exam process there is exam control module for instructor. Exam control part lets instructor define exam duration, list all the laboratory computers and their statuses, start and finish exam. Instructor can easily see if there is a problem in any computer, if student has finished the exam.

- Exam evaluation:

After the exam is finished the instructor can begin to the evaluation process. S/he can easily list all the students by selecting the course. Student's exam files are accessible directly from the program window. Student's files are downloaded by single click and after this can be available in offline. To make easy the process of evaluation there is also download all function which allows download all students files at once. If instructor decides to access the files out of office and without internet connection, this is the best way to make evaluation process comfortable. Also instructor can list all students with their grades, make a report, send the results to iPhone or iPad devices and export to CSV file.

3.2.2 Student's Functions

Students have functions during exam and the submission of assignment. The functions are described in details below:

- Login to exam:

Login is the first process for starting exam. To perform the login process successfully student must fill username and password given by instructor or received by iPhone or iPad application. If student will sit to wrong place he cannot login successfully.

- Get exam questions and files:

While student performs login process the system automatically begin downloading the exam files and questions. The files are shown to student in download section. Usually student sees one file with exam questions and a folder that contains extra files for exam. During the exam the time is shown in upper right corner of the program. After getting files student can open the computer program that is needed for exam and begin to solve the questions.

- Upload exam results:

After student finishes solving the questions s/he can submit the answers. There is upload section in the bottom of the program where student browses files or whole folders and uploads them. After upload is finished student gets a message about the state of uploading process. Student can upload files only once and system prevents student's second submission.

- Login to assignment system:

Assignment part is web-based application and students access the system after login. They enter user name and password that is given by instructor. Student id is used as user name.

- Get the assignment information:

After successful login the list of available assignments is shown to student. Student gets the information about the assignment by clicking appropriate assignment. Assignment name, assignment end date, description and attached files.

- Submit the assignment:

After the solving assignment student can upload their solutions. There is opportunity select up to 5 files. If there are more than 5 files student can archive with zip format and then upload. Student must know that the submission time is also inserted to the system in submission process. Once student upload the assignment solutions system prevents student upload the second time.

- iPhone and iPad applications:

IOS application makes easy to learn grades of exams and assignments and get notifications from instructors. Students can learn their sit, exam username and password before the exam from iOS application.

3.3 Project Requirements

Several requirements need to be followed for implementation of the thesis. These requirements consist of software, hardware and financial requirements which are shown below:

3.3.1 Software Requirements

Thesis implementation was developed in .NET platform that's why .NET Framework 4.0 is required in instructor's computer and in all computers in laboratory.

The second requirement is Windows Firewall. The thesis implementation is designed for Windows 7 because built-in Windows Firewall let do a lot of stuff. Windows Firewall must be turn on for home networks and public networks.

3.3.2 Hardware Requirements

For efficient processing of the application some hardware configurations are required.

The hardware requirements are divided into two parts:

- Server side requirements:

MS Windows Server 2008 or Windows 7 is needed for functioning of the server side.

For the database MS SQL Server 2008 R2 is needed. The database must be configured in such a way that it will be possible to connect the database out of the server. Also IIS 7.0 must be configured for the functioning of assignment system and the web service that is used by iOS application. The last thing that must be installed is FTP. This is for uploading and downloading exam questions, files and solutions.

- Client side requirements:

The only thing that is required in client side is MS Windows 7 installed computer with network connection. Network connection is used for authenticating, downloading questions and for uploading the solutions.

3.4 Structure of the Conventional System and Proposed System

Previously computer aided exam software was developed for our university [14]. It will be called as previous exam system in this thesis. Previous exam system worked as follows: All student computers connected directly to the instructor's computer which acted as a server. So, in download and upload process there was jamming because everyone wants to connect to the instructor's computer and download the questions and after the exam finishes everyone sends solutions to the instructor's computer at the same time. At this point instructor's computer cannot receive the requests at the same time due

to the hardware configurations of the computer. Most of time at the end of the exam there is a bottleneck in the system. So there was no guaranty that instructor gets the solutions in the order that students sent. Also there could be missing of some solutions of the students due to this architecture. On the other hand system after some period checked the availability of the student computers and could get crashed. So this architecture is not practical. The structure of connections between student computers and instructor's is shown in Figure 2.

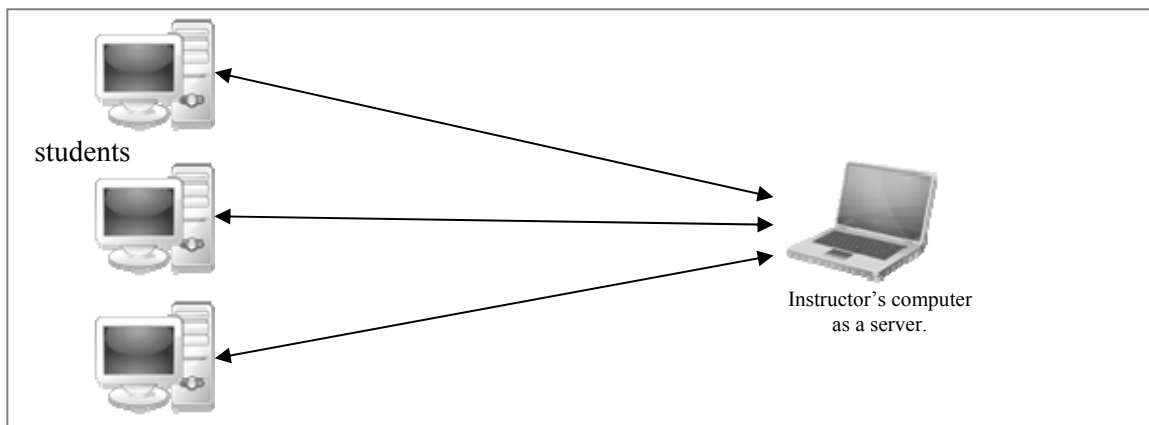


Figure 2. Connections Structure of the Previous System

The proposed system is designed in such a way that none of students' computers are dependent from instructor's computer. Both student side and instructor side are connected to the selected separate server. When instructor sends command, this command is saved in server and can be accessible to the student's computer. The structure of connections between student computers and instructor's in proposed system is shown in Figure 3.

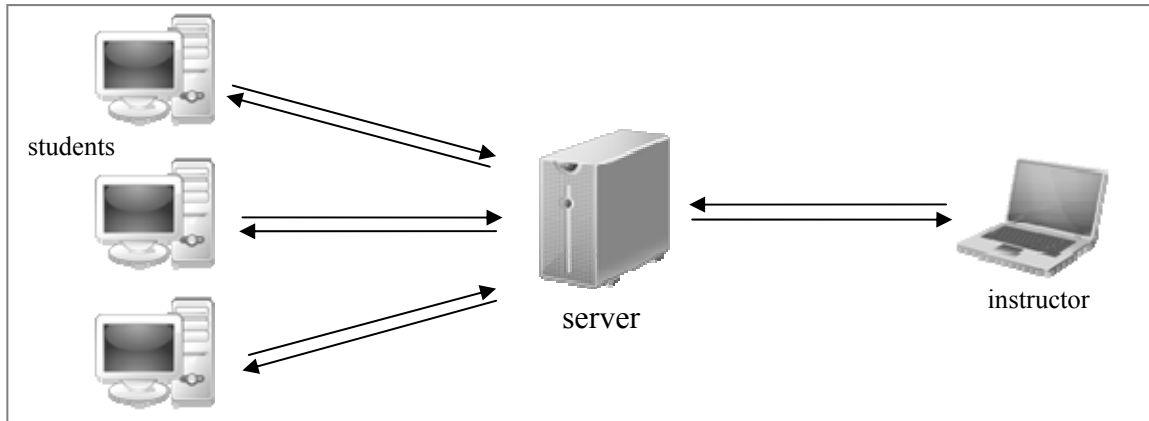


Figure 3. Connections Structure in Proposed System

Also there is one main difference between conventional systems and proposed system. This difference is that proposed system was not developed with traditional library based architecture, but business object architecture is used. The conventional architecture makes it difficult to develop application for different platforms. For every platform the program must be developed separately. This takes a lot of time for creating such system and makes difficult further changes. If there is a need to change one function in the system developers must change all programs that developed for different platforms. This systems' architecture is described in Figure 4.

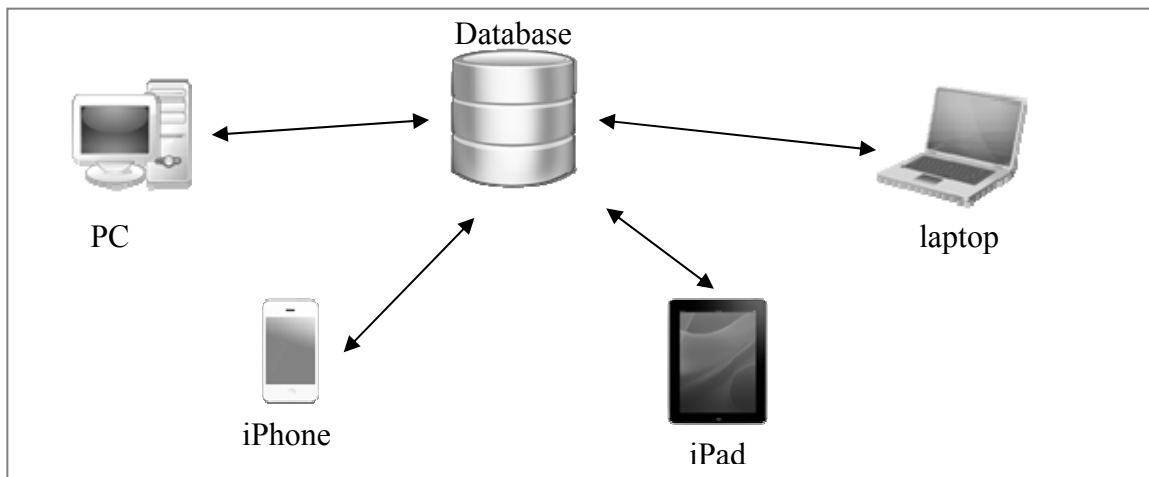


Figure 4. Structure of Different Platforms in Conventional Systems

But this problem is solved with business objects. By using business objects only user interface must be created for different platforms. The functionalities and all logic of the system are combined in business objects. Any change of the business object automatically takes place in all platforms. The best explanation for comparing these systems is described in figure 5.

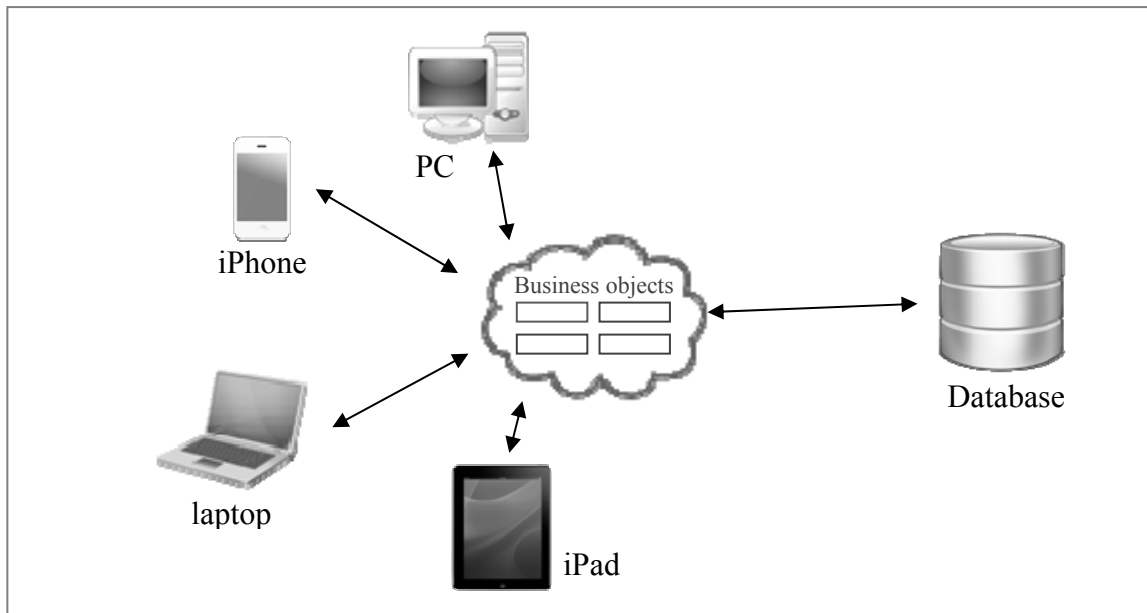


Figure 5. Structure of Different Platforms that Use Business Objects

In conventional systems business policies are hard-coded, that is, the rules are programmed directly by using programming languages. It means that business rules are merged into main program. It makes difficult for developers to differ business rules from the main program. If there is any change in rules, developers must identify and change all related source program. It makes system difficult to maintain[15].

With the development of hardware technologies, new devices come out. Designing the system with business object architecture gives fast development of the system for new

devices [19]. The part of the thesis iPhone and iPad applications is example where only user interface was created.

3.5 Database Design

In this part the data analysis is performed and pointed out the main entities, attributes and relationships between these entities. Also database normalization, entities definitions and entity relationship diagrams are described in this section. All the listed steps are explained in details below.

3.5.1 Conceptual Design of the System

Entities that are illuminated for this system are course, student, exam, assignment, assignment submission, grade, notification, exam passwords and exam results. Each of these entities has several properties which are called attributes that defines some information related with each entity. The relationship between these entities shows the action that can be done between entities. In this section the entities of the system with their attributes and the relationships between them will be illustrated in detail.

3.5.1.1 Entity Relationship Diagrams

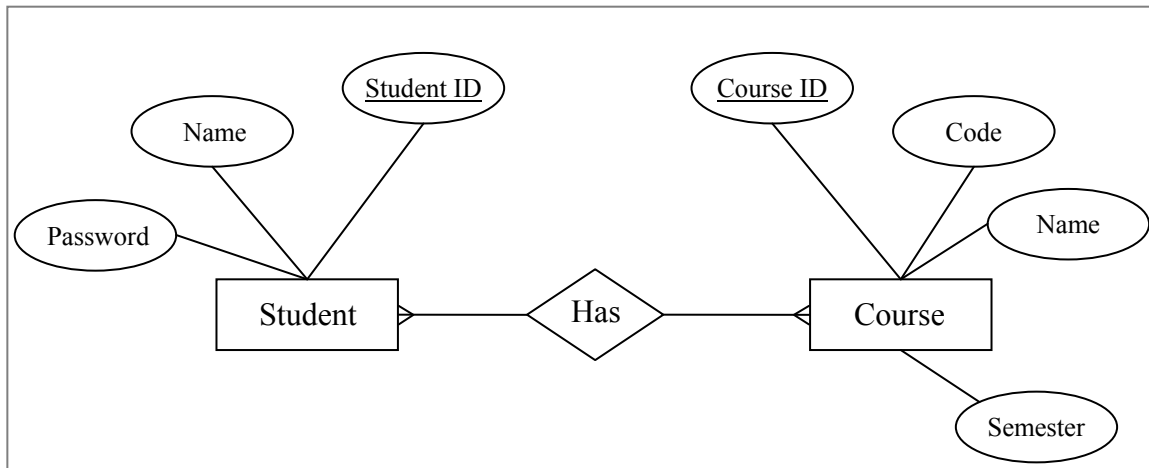


Figure 6. Relationship between Student Entity and Course Entity

As it shown in figure 6, student and course entities have many to many relationship, because every student can take more than one course.

The relationship between student and exam entities is many to many. Student can attend more than one exam. On the other hand in exam can be more than one student. The entity relation diagram is shown in figure 7.

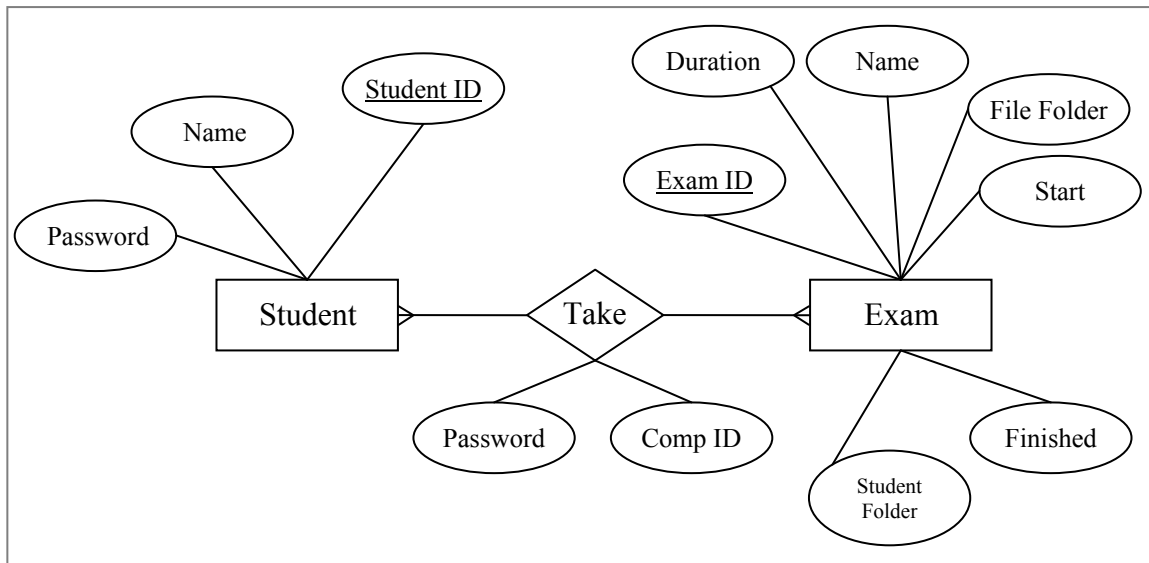


Figure 7. Relationship between Student and Exam Entities

Also there is a relationship between exam and course entities. Every course may have more than one exam. So the relation between exam and course is many to one. The entity relationship between these entities is shown in Figure 8.

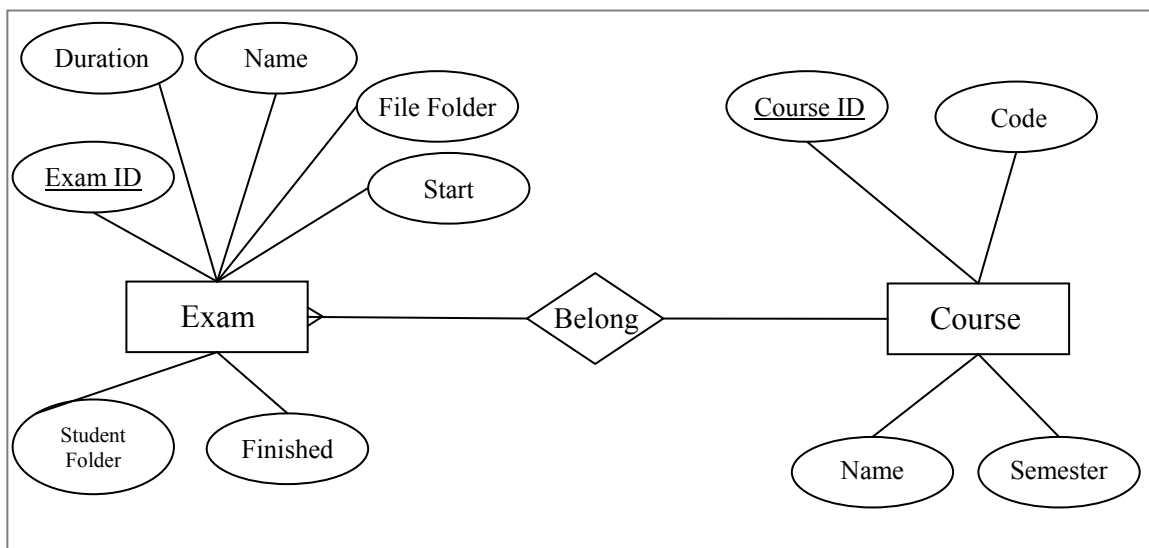


Figure 8. Relationship between Exam and Course Entities

The grades are given to the students in exam and assignment evaluation process. There is a grade entity for this process and it relates with student entity. The relationship between student and grade entities is one to many. As shown in figure 9 one student can get one than one grade from exam or assignment. Item ID attribute is added to the list of attributes of grade entity; this defines the id of the exam or assignment and can be differing by Type attribute.

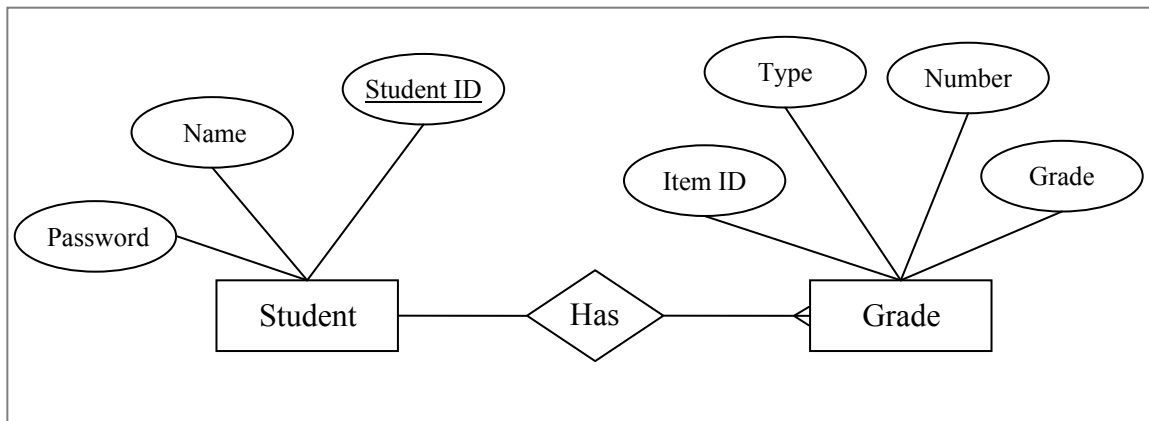


Figure 9. Relationship between Student and Grade Entities

There is one more function for a student, which is getting notifications. If there is such function then there must a relationship between these entities. The relationship between student and notification entities is one to many, as one student can get more than one notification. The entity relationship diagram is shown in Figure 10.

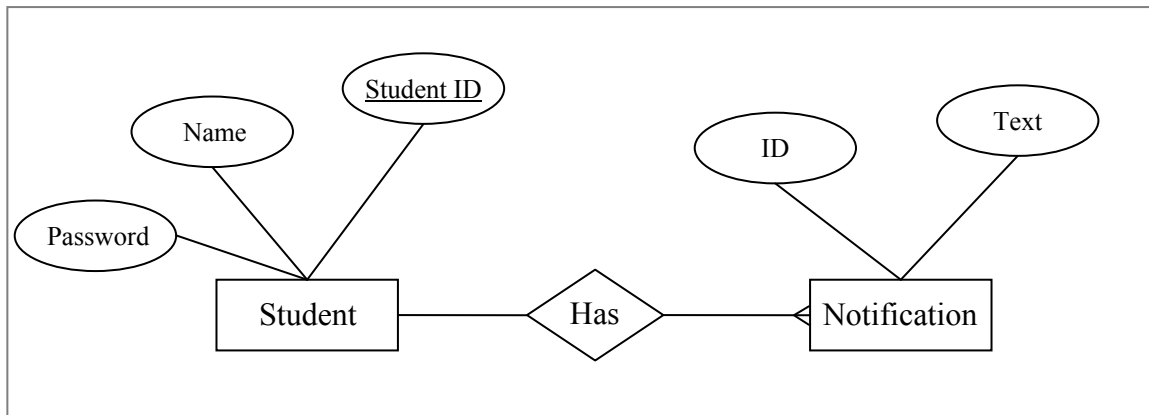


Figure 10. Relationship between Student and Notification Entities

One of the main entities of the system is assignment. Assignment entity has relationships with other entities. First there is a relationship between assignment entity and course entity. The relationship is many to one. It is clear that one course may have more than one assignment. The relationship diagram is shown in Figure 11.

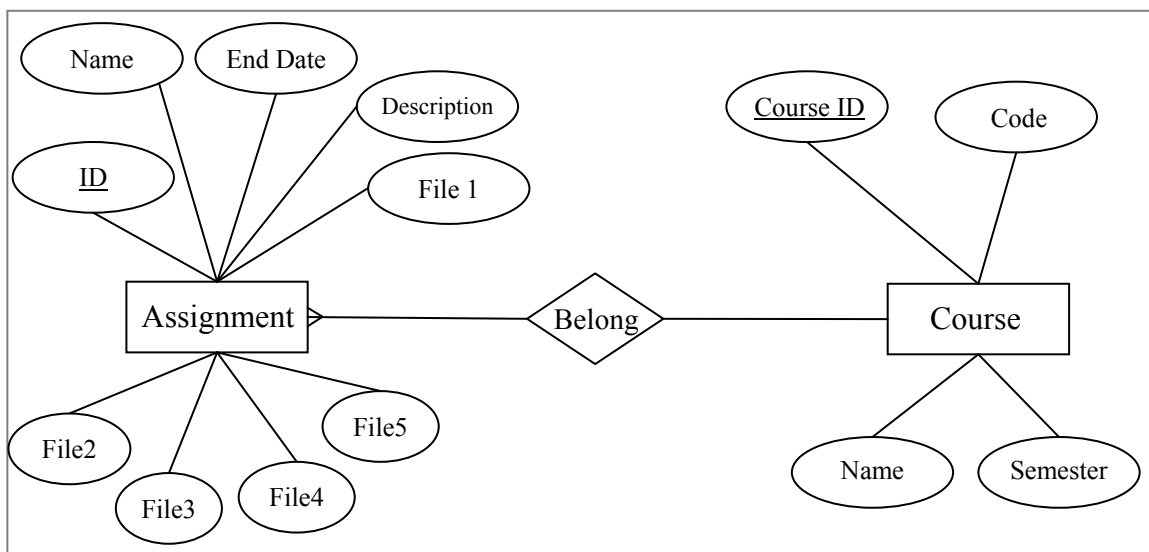


Figure 11. Relationship between Assignment and Course Entities

One of the main relationships is between student and assignment. This relation is created for submitting the solutions for the assignment by students. The relationship between

student and assignment is many to many. Because, more than one student can submit solutions for more than one assignment. The relationship diagram is shown below in Figure 12.

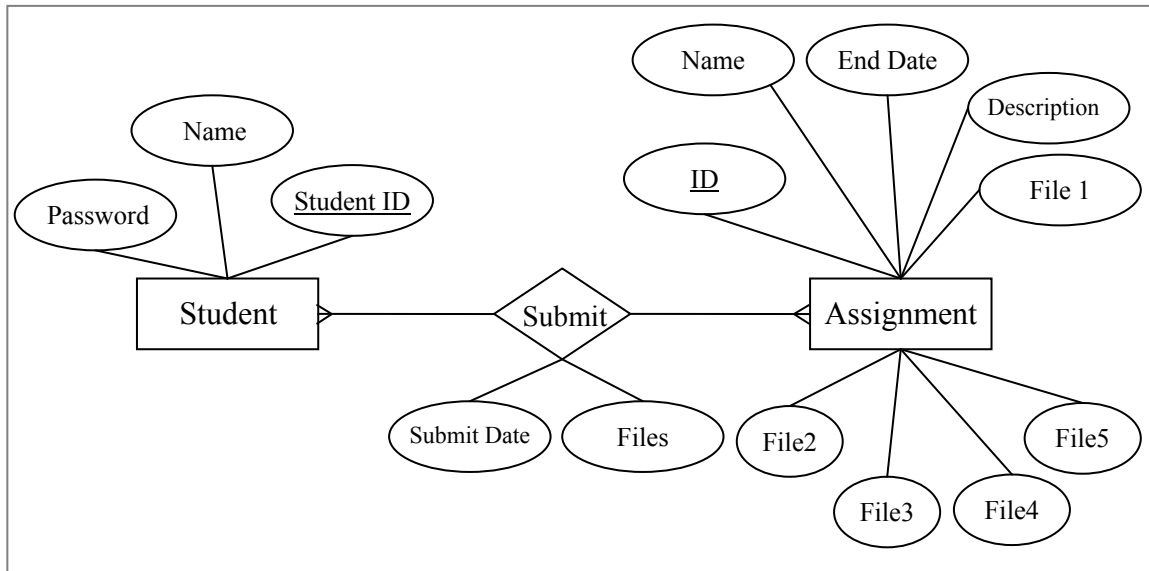


Figure 12. Relationship between Student and Assignment Entities

The last entity is laboratory entity. It has two relationships. These relationships are between exam and laboratory, and between laboratory and computer entity. Exam and laboratory relationship is many to one, while laboratory and computer relationship is one to many. This is explained as in one laboratory can hold more than one exam and in one laboratory may be more than one computer. These relationships are described respectively in Figure 13 and 14.

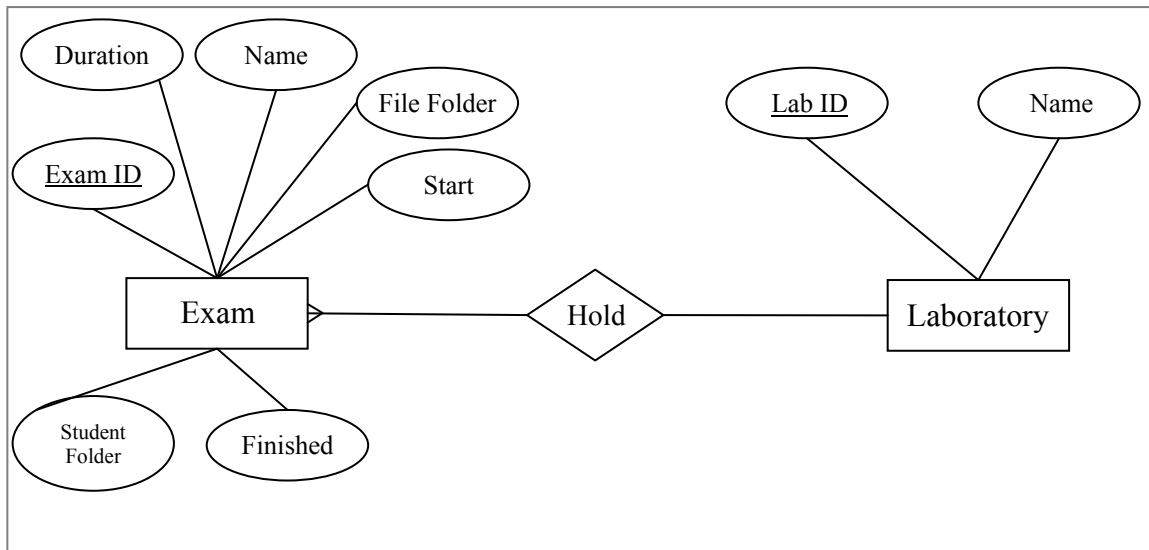


Figure 13. Relationship between Exam and Laboratory Entities

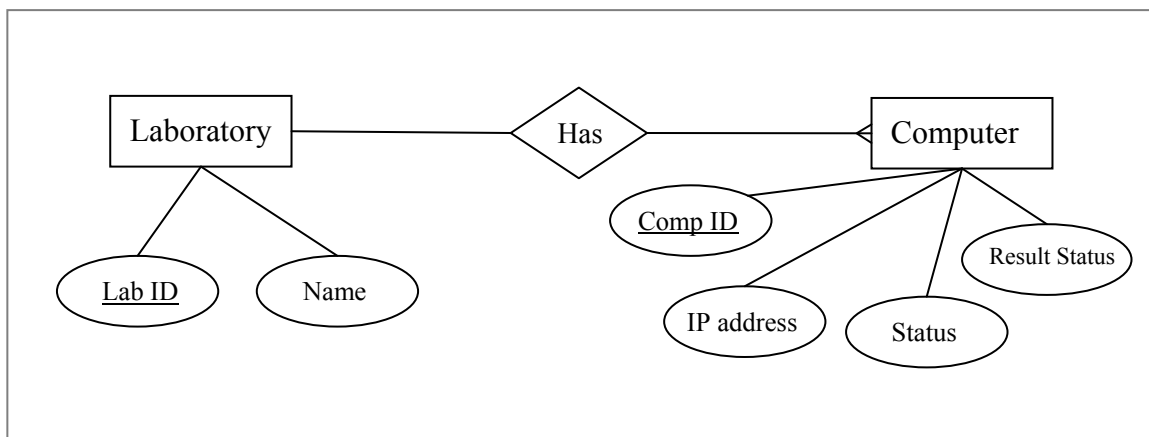


Figure14. Relationship between Laboratory and Computer

In figures 6-14 the relationships between two entities are described. Finally the whole entity relationship diagram will be shown below in figure 15.

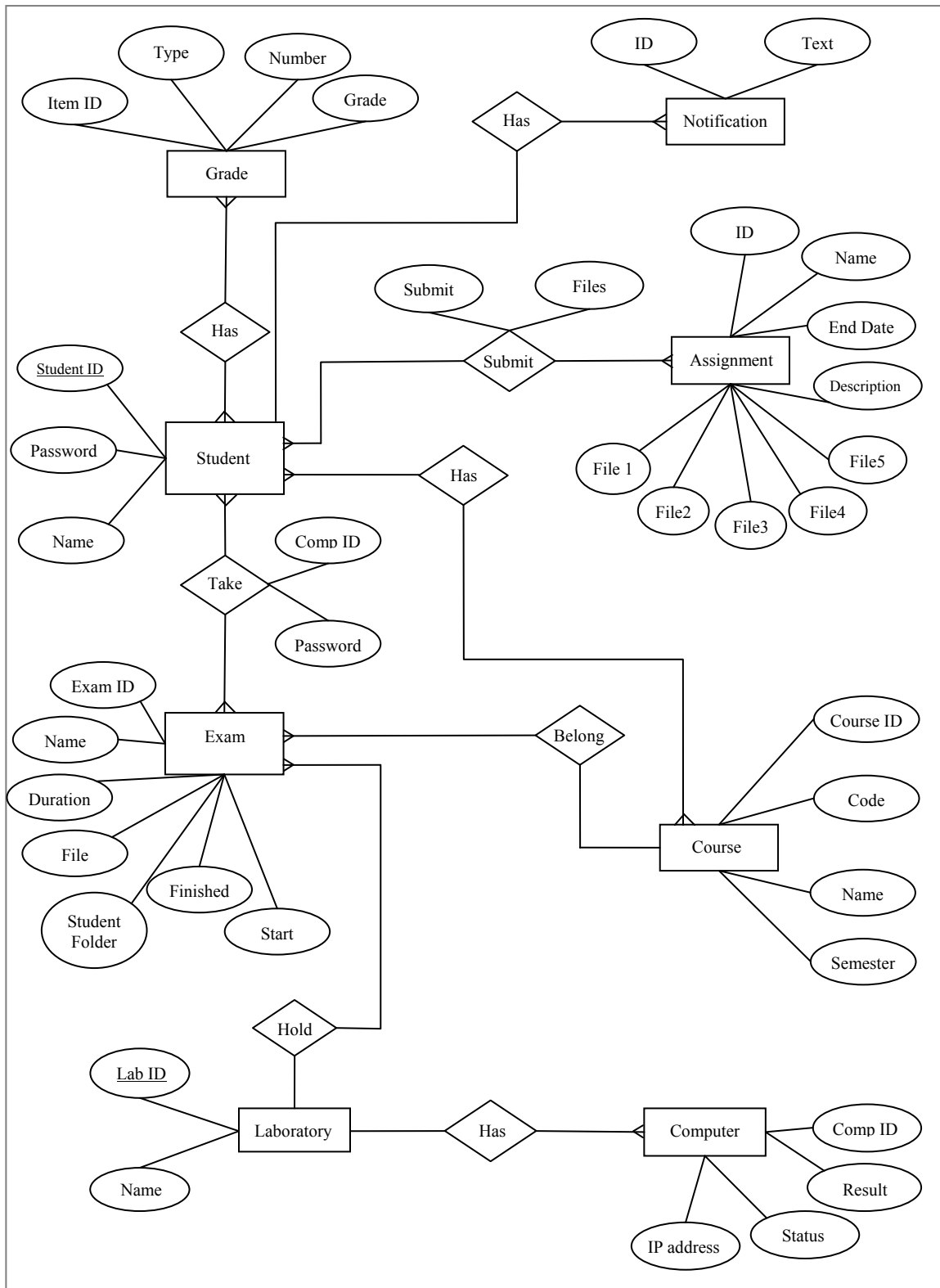


Figure 15. Complete ER Diagram

3.5.1.2 Entities of the System

Table 3. Student Entity with its Attributes

Attribute Name	Description
ID	Defines a six-digit identity of student
Name	Defines the name and surname of each student
Password	Defines the system access password for student

Table 4. Course Entity with its Attributes

Attribute Name	Description
ID	Defines identity of each course
Code	The code of the course / short name.
Name	The complete name of the course
Semester	Defines the semester when the course is opened.

Table 5. Exam Entity with its Attributes

Attribute Name	Description
ID	Defines identity of each exam
Name	The name of the exam
File Folder	The path of the folder where exam files are stored.
Student Folder	The path of the folder where student download files.
Exam File	The main exam file with questions.
Course ID	Defines the course which exam belongs.
Lab ID	Defines the laboratory where exam will be held.
Duration	Defines the duration of exam in minutes.
Started	Shows if exam have been started or not.
Finished	Shows if exam is finished.

Table 6. Grade Entity and its Attributes

Attribute Name	Description
Student ID	Defines the identity of student who gets grade.
Item ID	Defines the identity of exam or assignment.
Grade Type	Shows if it is exam or assignment.
Grade Number	Shows the number of question.
Grade	Defines the grade of the student for question.

Table 7. Notification Entity with its Attributes

Attribute Name	Description
ID	Defines identity of each notification.
Not Text	Defines the text of the notifications.
Student ID	Defines the identity of student who receives the notification.

Table 8. Assignment Entity with its Attributes

Attribute Name	Description
ID	Defines identity of each assignment
Name	Defines the name of the assignment
End Date	Defines the date and time when the assignment finish
Description	Any explanation or information about the assignment
Course ID	Defines the course which assignment belongs.
File 1	Attached extra file path
File 2	Attached extra file path
File 3	Attached extra file path
File 4	Attached extra file path
File 5	Attached extra file path

Table 9. Laboratory Entity with its Attributes

Attribute Name	Description
ID	Defines identity of each laboratory
Name	Defines the name of the laboratory.

Table 10. Laboratory Computers Entity with its Attributes

Attribute Name	Description
Lab ID	The identity of laboratory which computer belongs.
Comp ID	Defines the identity of each computer.
Comp IP	The IP address of the computer.
Comp Status	Shows the status of the computer. Available or not.
Result Status	Shows the status of the student submission.

3.5.1.3 Relationship of the E.R. Diagram

Table 11. Relationship between Student and Course

Relationship Name	Has	
	From Entity	To Entity
	Student	Course
Cardinality	1:N	1:N

Table 12. Relationship between Student and Exam

Relationship Name	Take	
	From Entity	To Entity
	Student	Exam
Cardinality	1:N	1:N

Table 13. Relationship between Exam and Course

Relationship Name	Belong	
	From Entity	To Entity
	Exam	Course
Cardinality	1:N	1:1

Table 14. Relationship between Student and Grade

Relationship Name	Has	
	From Entity	To Entity
	Student	Grade
Cardinality	1:N	1:N

Table 15. Relationship between Student and Notification

Relationship Name	Has	
	From Entity	To Entity
	Student	Notification
Cardinality	1:1	1:N

Table 16. Relationship between Assignment and Course

Relationship Name	Belong	
	From Entity	To Entity
	Assignment	Course
Cardinality	1:N	1:1

Table 17. Relationship between Student and Assignment

Relationship Name	Submit	
	From Entity	To Entity
	Student	Assignment
Cardinality	1:N	1:N

Table 18. Relationship between Exam and Laboratory

Relationship Name	Hold	
	From Entity	To Entity
	Exam	Laboratory
Cardinality	1:N	1:1

Table 19. Relationship between Laboratory and Computers

Relationship Name	Has	
	From Entity	To Entity
	Laboratory	Computers
Cardinality	1:1	1:N

3.5.2 Normalization

Normalization is a formal process for removing anomalies with re-organization the database structure. Normalization is applied to remove duplicate attributes and relationships. There are three normal forms which are applied to the proposed system's database structure. The proposed system achieved 1NF, 2NF and 3NF and these normalizations are illustrated below:

First Normal Form (1NF):

If the table does not have multi-value attributes then it is in the first normal form [11]. By applying 1NF insertion, deletion and update anomalies was reduced. Also primary keys have been defined to identify each row in the relation. The database which is used in the system is in the first normal form due to there is no multi-value attribute in the database.

Second Normal Form (2NF):

The database become in the second normal form if each of non-key attributes is depend on the primary key. In this step a new relation is created for each primary key attribute and non-key attributes that are dependent on this primary key are moved from old relationship to the new one[11]. For example, for the table exam primary key is Exam ID and then each attribute in this table is depend on that primary key such as exam name, duration, file, start and etc.

Third Normal Form (3NF):

The database is in third normal form if there is not any transitive relation. For removing anomalies and make database become the third form, new relationship is created for each non-key attributes to serve as foreign key[11]. All tables of database are in 3NF and it can be easily seen from diagrams that illustrated above. For example in student table (student id, name, password) there is no relation between non-key attributes (name, password) and each of them related with primary key.

Chapter 4

DESIGN AND IMPLEMENTATION

In this chapter, the design and implementation issues are presented. The architecture of the system is also described. According to the analysis made, the design of the database server, the design of the student's web, desktop and mobile applications and the design of desktop instructor's application are explained.

The system consists of web server, database, file server, desktop exam application, web assignment application, mobile course manager application and instructor's administrative application. All client applications access the database, get the necessary data from the database and enter the data into the database. All modifications made on the database can be seen by all clients at the same time.

The implementation of all parts of the system is illustrated below.

4.1 Design Issues of the Database Server

According to the analysis on database of the system, the E-R diagram is obtained and illustrated in previous chapter. Smart Draw program is used to build data flow diagram and flow chart diagram for each action. Smart Draw program is one the famous program using for drawing decomposition work such as ERD, DFD, use case Diagram, and etc.

4.2 System Architecture

Data flow diagram represents the architecture of the system; therefore building this diagram gives clear idea about action sequences between client and server. The architecture of the system is explained in details below.

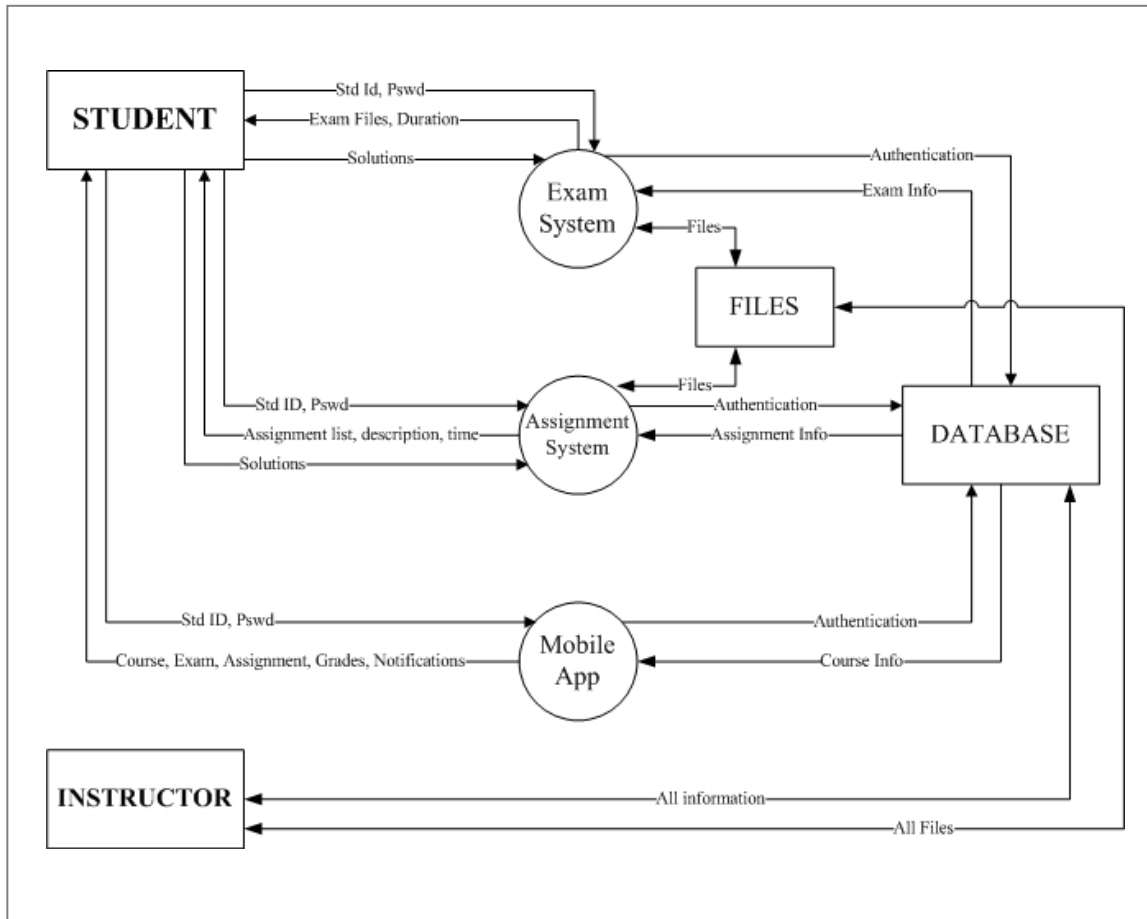


Figure 16. Data Flow Diagram of the System

As shown in data flow diagram student and instructor are external parts; exam system, assignment system, mobile application, database and file server are internal parts. To indicate the data and the direction of the flow of data labeled directed arcs are used.

Whole system consists of exam system, assignment system and mobile application those are managed by instructor. Instructor has access to all parts and can modify any information in the system. S/he has privileges like an administrator of any system, so in the current system s/he can manage assignments, held exams and be in contact with students from mobile application.

Student's can access to exam system, assignment system and mobile application only with authentication. Access to unauthorized parties is prevented by the system. Students get two types of passwords: temporary and private. For the exam system they use temporary password which is changed every exam, but for assignment and mobile application they can access with their private password. As shown in data flow diagram the first thing that student must do for accessing the exam system is login. Student enters his or her student id and the temporary password that is given by instructor. Students can get this password with mobile application and during the exam. After successful authentication system sends exam files, questions and exam duration time to the student computer. This point is the starting of the exam. After these steps is done successfully student can open the necessary program for solving the exam questions. The last step is sending solutions to the exam system by students. After solving all questions they can submit the answers and get response about the status of submission from the system. On the other hand exam system has connection with database and file server. Exam system accesses to database for checking the authentication information and for getting the exam information. Also system accesses to file server for retrieving the exam files and for uploading student files. When the system gets files it uploads the files directly to the file server and changes the status information in the database.

Student can access the assignment system from any internet browser by entering student id and private password. The assignment system is connected to the database and file server, too. The system accesses to database for authentication process, then gets the list of available assignments and shows them to the student. Student can see information like assignment name, description, deadline time, extra attachments and etc. Then student submits the solutions of assignment to the system and system upload the files to the file server and responds the status of the submission.

As the assignment system student authenticates to the mobile application with student id and private password. Application checks the authentication information from the database and shows the student the list of courses. Students access to exams, assignments and grades by selecting the course from the list. Also they get notifications from status bar.

There is no connection driver of SQL Server for iOS applications. So the best way that is approved by practice is selected. It is web service. The mobile application part is not so complicated, that's why HTTP requests with JSON are used. The scheme of connections between mobile application, web service and database is shown in Figure 17.

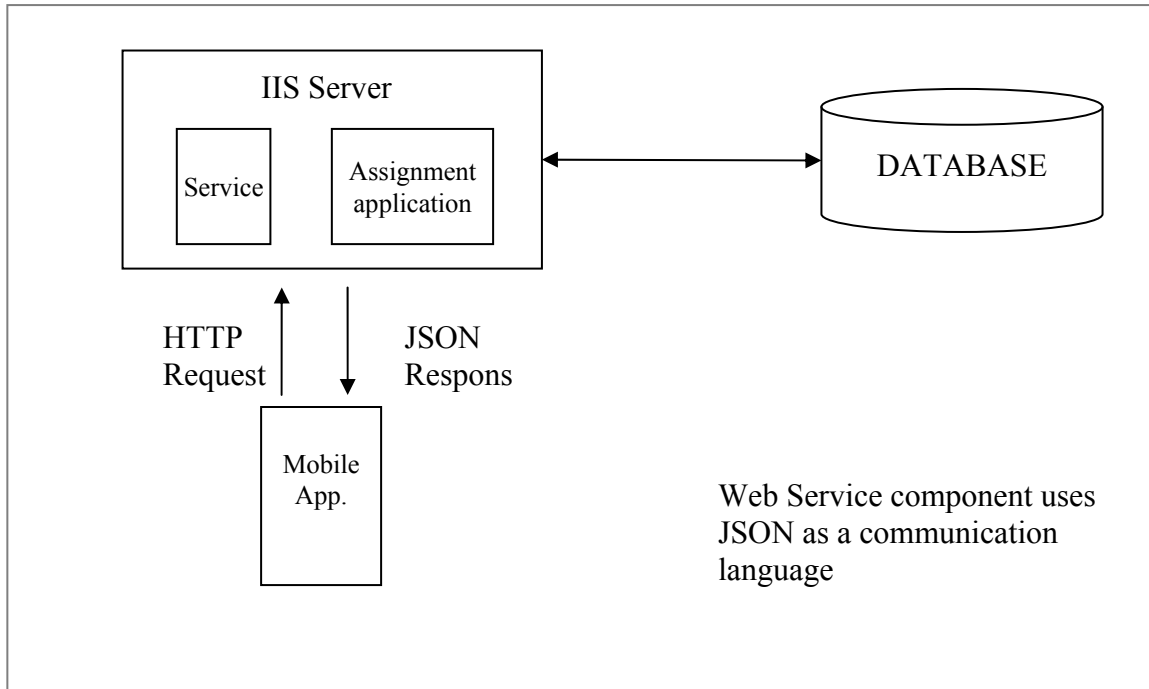


Figure 17. Communication between Mobile App and Server by Web Service

4.3 Flow-chart Design of the System

There are several actions that can be done by student and instructor. These actions are described before in previous chapter. In this section the flow chart details of these actions will be illustrated.

4.3.1 Student – Exam system Flow Chart

For making the system more secure and define the student's sit the first thing that student must do is login. As shown in figure 18 student enter student id and password. System checks the validity of the user credentials and the sit place. If student enters true student id and password to another computer s/he will not access the system. Because, s/he can access only the computer that is defined before the exam. Then system

accordingly downloads the exam files, starts the timer. Student can upload the solution files before timer ends.

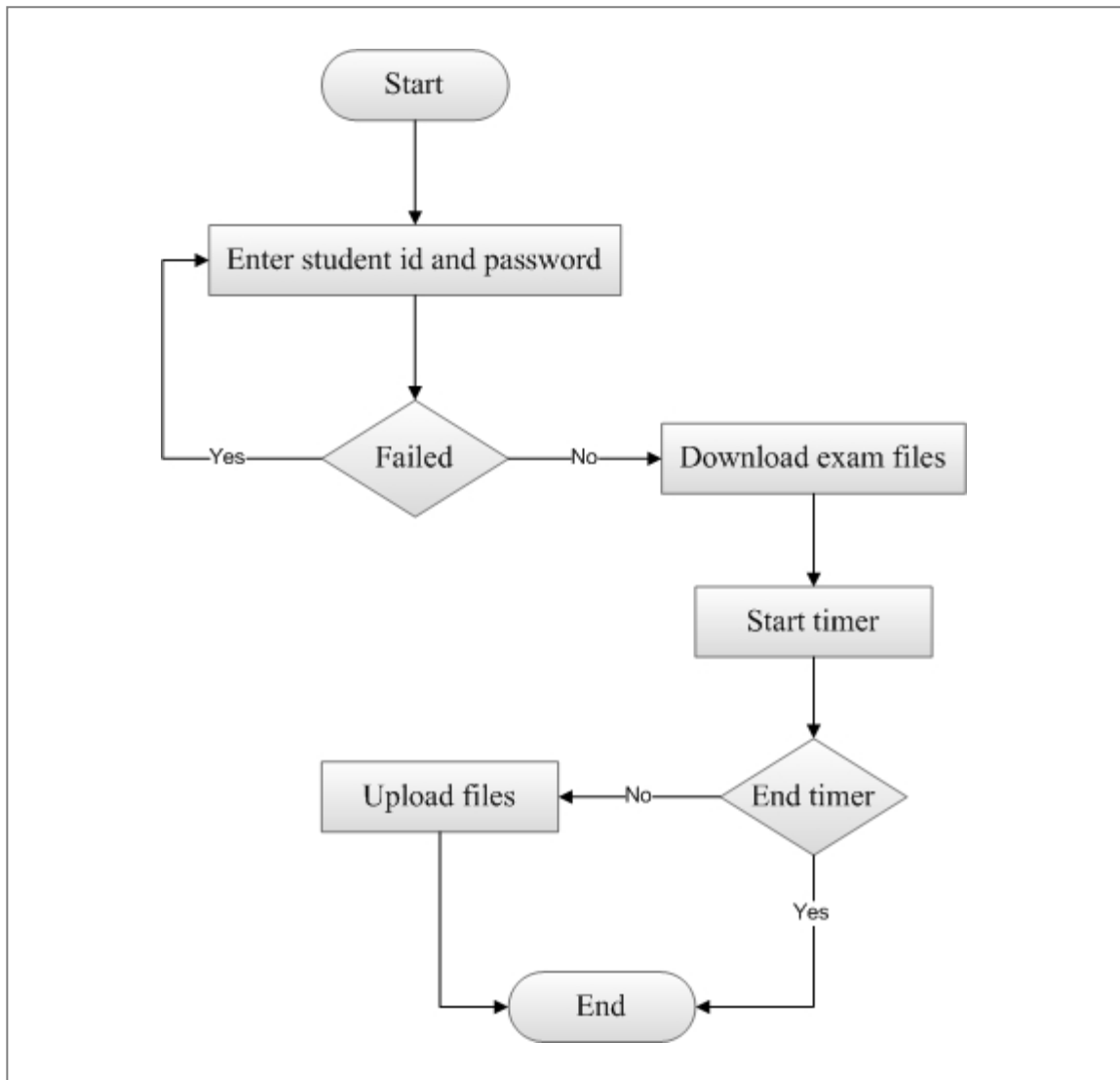


Figure 18. Student – Exam System Flow Chart

4.3.2 Student – Assignment System Flow Chart

The first step is authentication, so student enters student id and password. Then system accordingly shows the list of assignments and detailed information about the

assignment. Detailed information contains assignment name, description, end date time and extra attachment files. After review students can submit their solutions to the system. The flow chart is shown below in Figure 19:

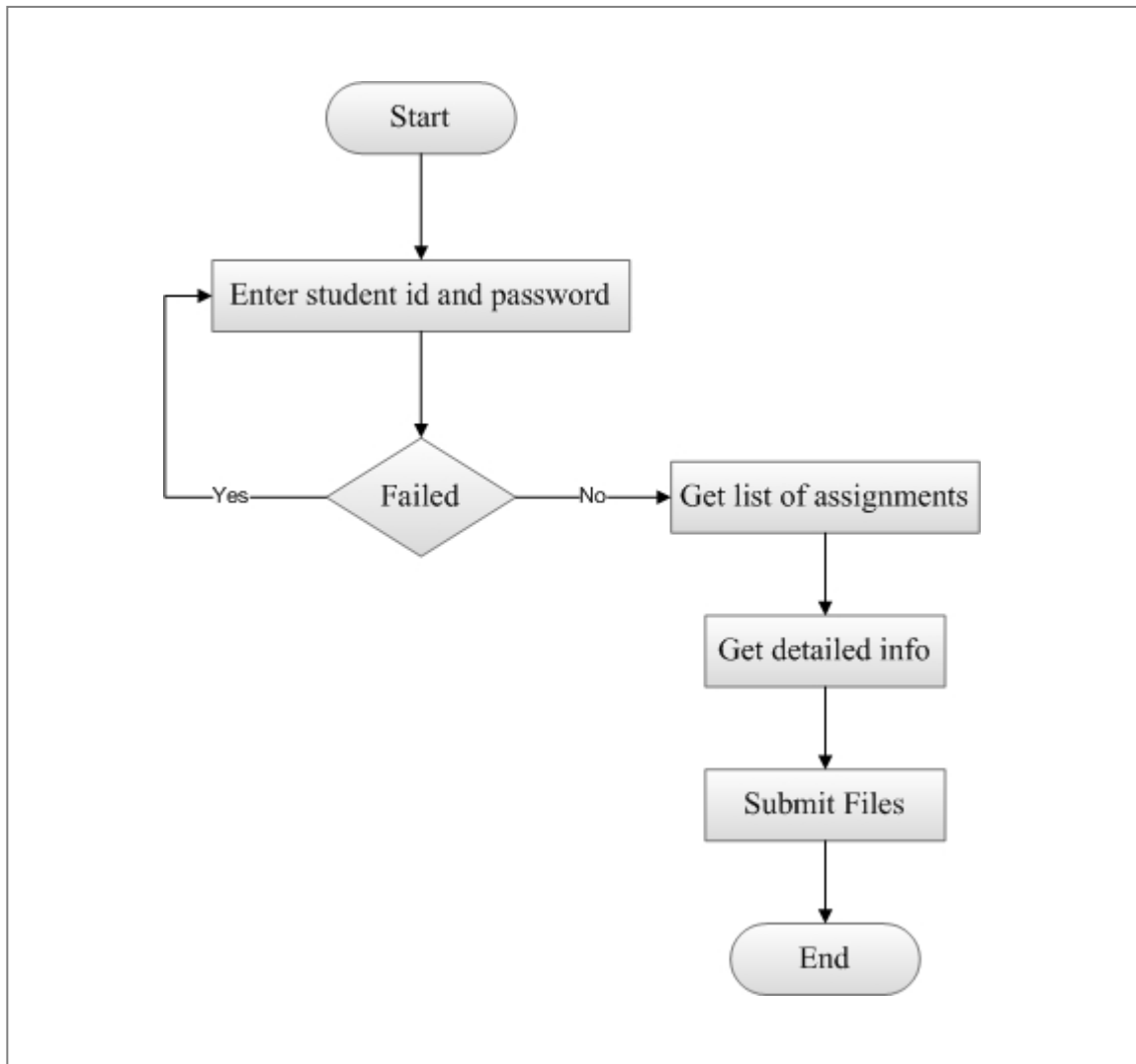


Figure 19. Student – Assignment System Flow Chart

4.3.3 Student – Mobile Application Flow Chart

As in other systems mobile application is accessible with authentication. The steps which student can do are login, select course, select assignment or exam and see the

grades. Also student can get notifications from instructor. All these steps are illustrated in flow chart below in Figure 20:

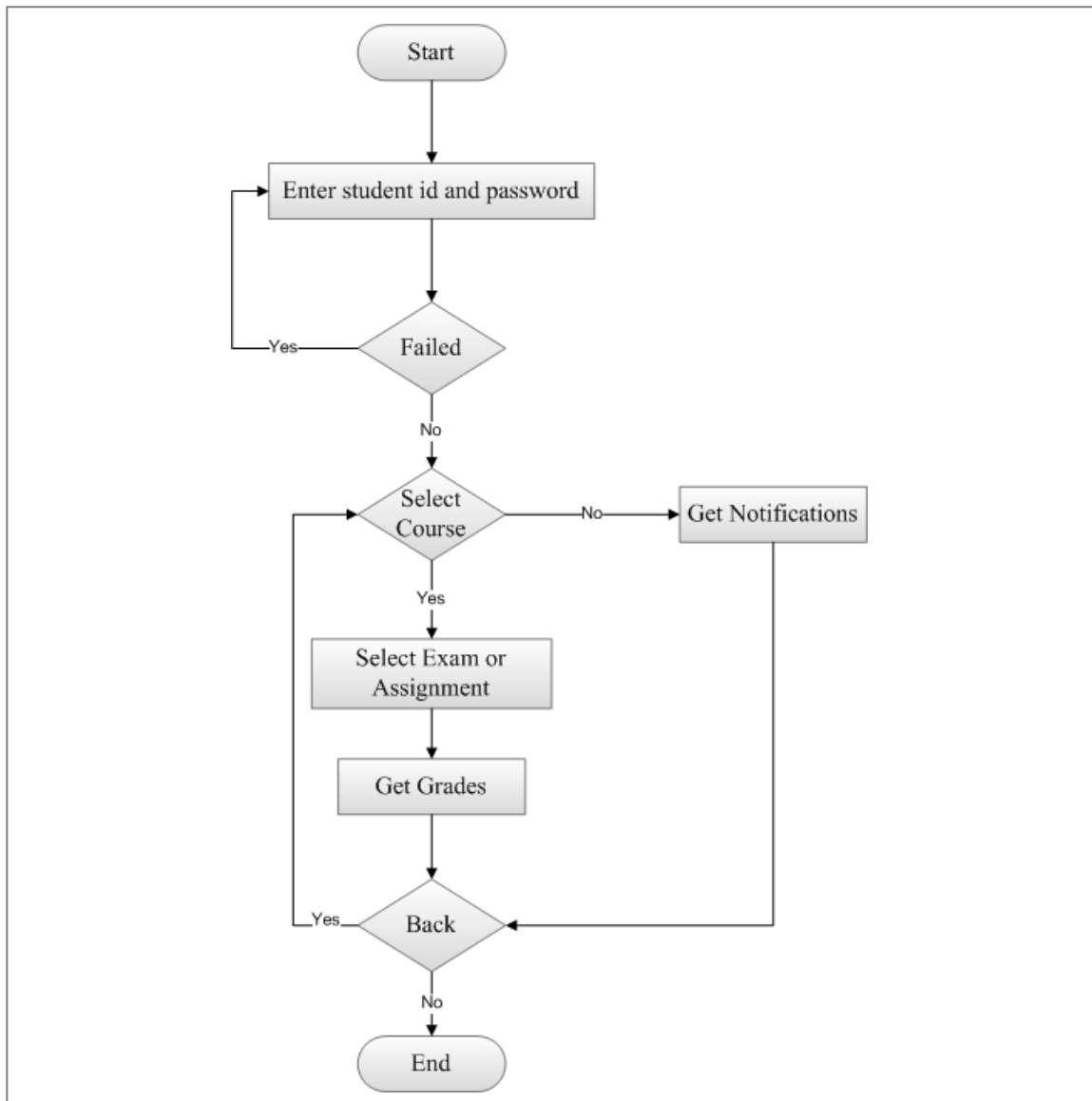


Figure 20. Student – Mobile Application Flow Chart

4.4 Implementation

Implementation of the server side database is realized by SQL Server 2008. All tables, relations and all SQL queries like select, insert, update, delete are implemented in this database. Also ASP.NET is used in server side for web application.

For the implementation of client side, different technologies are used depending on platform. For the exam system Microsoft Visual C# .NET is used as development environment. It is desktop application, so it is Windows Form application. MS Visual C#.NET is a powerful programming environment for the developers who create applications on Microsoft .NET platform [12]. The assignment system is Web Application and is developed in ASP.NET and C# platforms.

Also there is an iOS application for iPhone and iPad, Objective-C programming language is used. Objective-C is object-oriented programming language which allows create native applications for OS X platform. Also iOS SDK is used for developing the application which allows develop application for any iOS device [13].

4.4.1 Server

The server consists of IIS web server, File Server and SQL Server. All client applications use the same database. The server exchanges data with clients. IIS serves for assignment system and for web service for mobile application. File server is the common place where all files are downloading and uploading.

4.4.2 Database Server

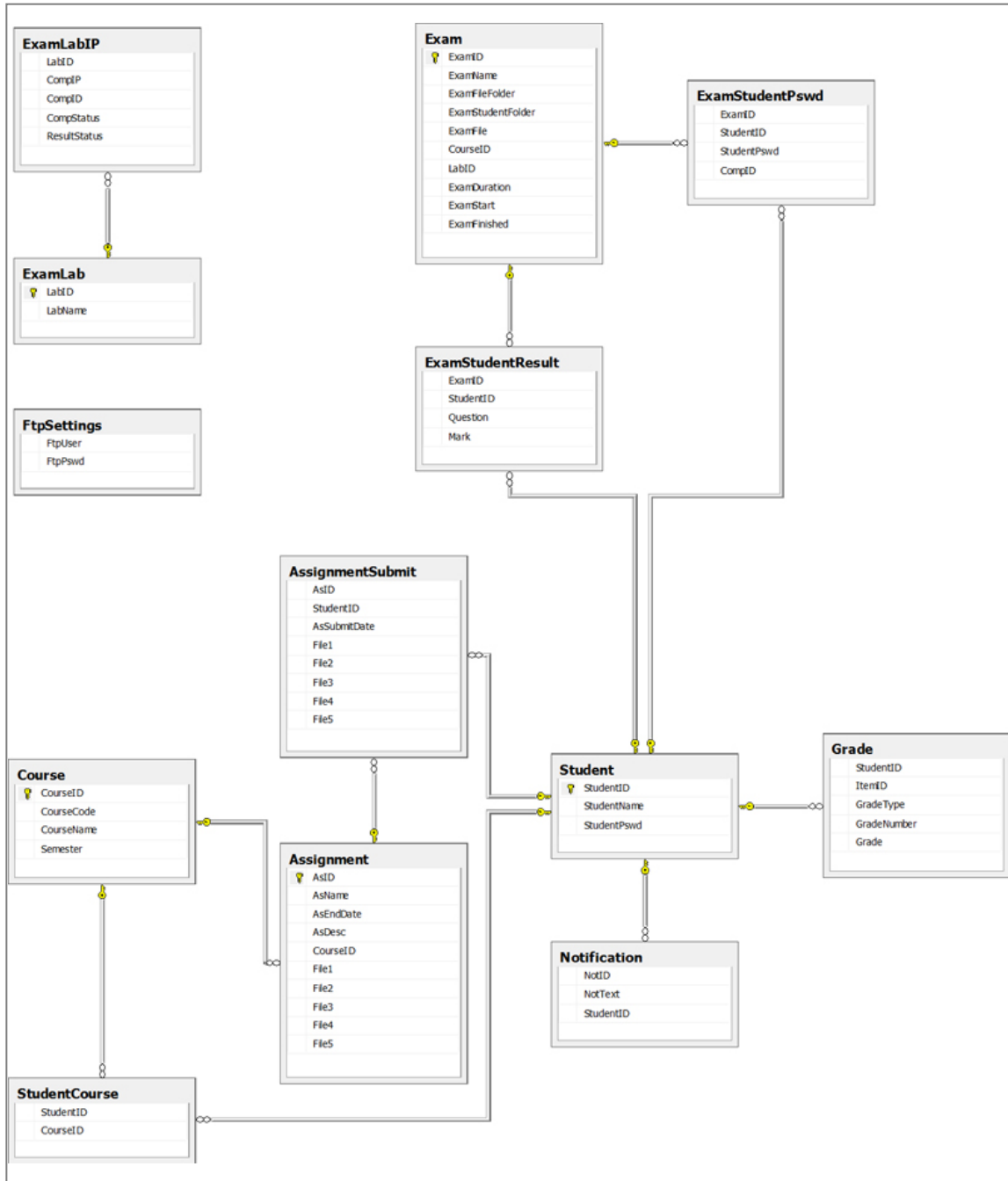


Figure 21. Tables and Relations of the Database

According to the database analysis and E-R diagram, the database system is implemented in Microsoft SQL Server 2008. The automatically generated SQL queries

by Microsoft SQL Server Management System tool are used to create the tables and relations in SQL Server. The relational diagram of the database is shown in figure 21.

4.4.2.1 Tables

The description of the columns of tables is illustrated in previous chapter. In this section tables that database contains, columns, data types and etc is shown:

Table 20. Students Table

Table Name	Student		
Column Name	Data Type	Required	Primary key
StudentID	Varchar (7)	Yes	Yes
StudentName	Varchar (100)	Yes	No
StudentPswd	Varchar (128)	No	No

Table 21. Courses Table

Table Name	Course		
Column Name	Data Type	Required	Primary key
CourseID	Int	Yes	Yes
CourseCode	Varchar (20)	Yes	No
CourseName	Varchar (100)	Yes	No
Semester	Varchar (20)	Yes	No

Table 22. Student Course Table

Table Name	StudentCourse		
Column Name	Data Type	Required	Primary key
StudentID	Varchar (7)	Yes	No
CourseID	Int	Yes	No

Table 23. Assignments Table

Table Name	Assignment		
Column Name	Data Type	Required	Primary key
AsID	Int	Yes	Yes
AsName	Varchar (100)	Yes	No
AsEndDate	Datetime	Yes	No
AsDesc	Text	Yes	No
CourseID	Int	Yes	No
File1	Varchar (300)	Yes	No
File2	Varchar (300)	Yes	No
File3	Varchar (300)	Yes	No
File4	Varchar (300)	Yes	No
File5	Varchar (300)	Yes	No

Table 24. Assignment Submission Table

Table Name	AssignmentSubmit		
Column Name	Data Type	Required	Primary key
AsID	Int	Yes	No

StudentID	Varchar (100)	Yes	No
AsSubmitDate	Datetime	Yes	No
File1	Varchar (300)	Yes	No
File2	Varchar (300)	Yes	No
File3	Varchar (300)	Yes	No
File4	Varchar (300)	Yes	No
File5	Varchar (300)	Yes	No

Table 25. Exams Table

Table Name	Exam		
Column Name	Data Type	Required	Primary key
ExamID	Int	Yes	Yes
ExamName	Varchar (50)	Yes	No
ExamFileFolder	Varchar (100)	Yes	No
ExamStudentFolder	Varchar (100)	Yes	No
ExamFile	Varchar (100)	No	No
CourseID	Int	Yes	No
LabID	Int	No	No
ExamDuration	Int	Yes	No
ExamStart	Int	Yes	No
ExamFinished	Int	Yes	No

Table 26. Exam Laboratories Table

Table Name	ExamLab		
Column Name	Data Type	Required	Primary key
LabID	Int	Yes	Yes
LabName	Varchar (50)	Yes	No

Table 27. Exam Laboratory IPs Table

Table Name	ExamLabIP		
Column Name	Data Type	Required	Primary key
LabID	Int	Yes	No
CompIP	Varchar (15)	Yes	No
CompID	Int	Yes	No
CompStatus	Varchar (10)	No	No
ResultStatus	Varchar (10)	No	No

Table 28. Students' Temporary Exam Passwords Table

Table Name	ExamStudentPswd		
Column Name	Data Type	Required	Primary key
ExamID	Int	Yes	No
StudentID	Varchar (7)	Yes	No
StudentPswd	Varchar (256)	Yes	No
CompID	Int	Yes	No

ExamStudentPswd table is not only for temporary password. This table also keeps the information about the sit place of the student in the exam. CompID column is filled when the password is generated. This column gets random integer value which is less than the number of computers in the laboratory. Student passwords are encrypted so from the database they are unreadable.

Table 29. Grades Table

Table Name	Grade		
Column Name	Data Type	Required	Primary key
StudentID	Varchar (7)	Yes	No
ItemID	Int	Yes	No
GradeType	Varchar (20)	Yes	No
GradeNumber	Int	No	No
Grade	Varchar (10)	Yes	No

Table 30. Notifications Table

Table Name	Notification		
Column Name	Data Type	Required	Primary key
NotID	Int	Yes	Yes
NotText	Text	Yes	No
StudentID	Varchar (7)	Yes	No

4.4.2.2 Usage of Tables in Implementations

There are four different programs implemented to take exam, to submit assignment, to learn grades and administrative control. Among those systems, the Exam system uses Student, Exam, ExamStudentPswd, ExamLab tables. Assignment system uses Student, Assignment and AssignmentSubmit tables. Mobile application uses Student, Course, Exam, Grade and Notification tables. Student table contains all information about the student for authentication process, that's why all of the applications use this table for checking shown student id and password by the student. Administrative application which is designed for instructors uses all tables depending on operation, because this application can manage all other applications and administrator related processes.

4.4.3 Client Applications Implementation

Client side consists of four different applications for exam taking, for assignment submission, for getting results and for controlling and managing all the system. Each of the systems is implemented in different platforms.

Exam system is a desktop application which is implemented in Microsoft Visual C# .NET programming environment. Also built-in libraries are used to control the firewall of the computer for blocking the network access.

Assignment system is web application and implemented in MS Visual C# .NET programming environment, too. The ASP.NET platform is used to develop the application. The assignment system works in IIS 7 web server.

Mobile application is called “Course Manager”. It is implemented in Objective-C programming environment. It is designed for iOS devices such as iPhone and iPad. Course Manager application is available for downloading in Apple App Store. The user interface of the application is designed in user-friendly manner by selecting from the list step by step. All list items are described and separated in readable manner. The navigation between pages make user never be lost in application

Instructor’s administrative application is implemented in MS Visual C# .NET programming environment and it is desktop application. It is designed to control whole system. All the functionalities are grouped in menus and submenus. Also forms are designed in such a way that the order of the functionalities is the same with the action buttons or fields change actions.

4.4.4 Mobile Application Implementation

Mobile part of the application is developed for iOS devices like iPhone, iPad and is implemented in Mac OS X environment with Objective-C programming language. As IDE tool Xcode is used. User interface is designed, classes that do main job is written and connectors between user interfaces and classes are created. Also screen resolutions for different versions of devices, vertical and horizontal positions are configured. Application is published in Apple App Store and can be modified or controlled by iTunes Connect portal. Application name is Course Manager and how it can be managed from Apple account is shown below:

iTunes Connect can be access from the <http://itunesconnect.apple.com>. The login page is shown below in Figure 22:

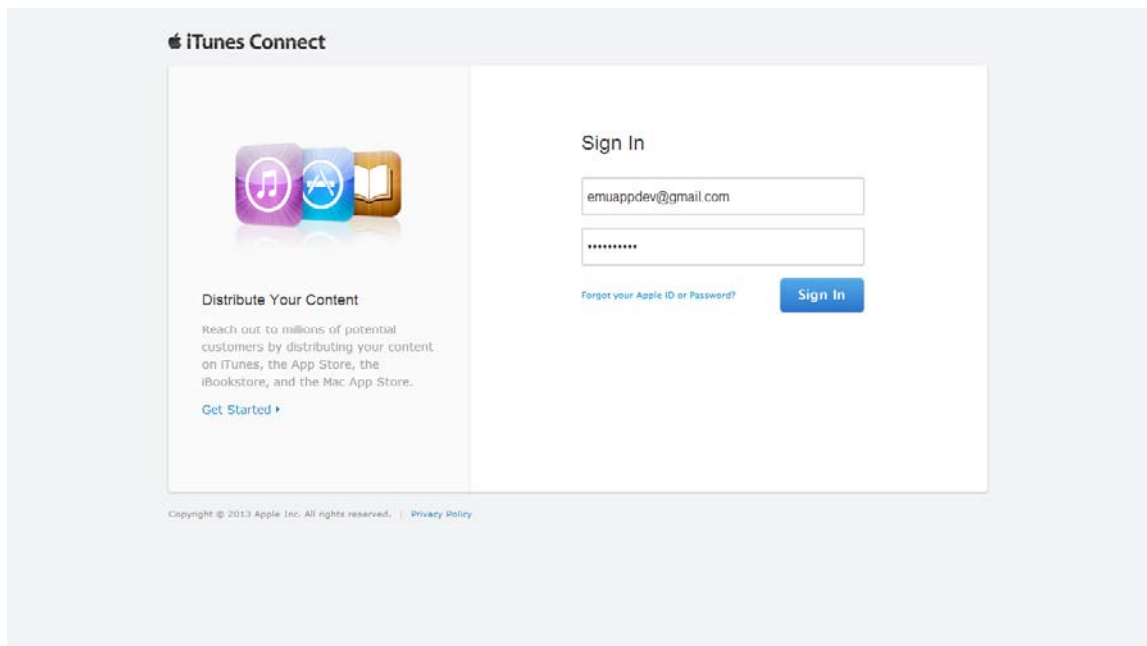


Figure 22. iTunes Connect Login Page

User name is `emuappdev@gmail.com` and password is `cmpei0sDev`. After login the list of operations are shown for managing applications, follow sales and trends, get reports about downloads and etc. This page is shown in figure 23.

Manage Your Apps link allows add, view, manage App Store applications. After clicking this link our application is shown with some details. There is information about version, creation date, and status. Here it is possible to add new versions of application, edit metadata, screenshots and information about the application. This page is illustrated in Figure 24. The current status of the application is ready for sale.

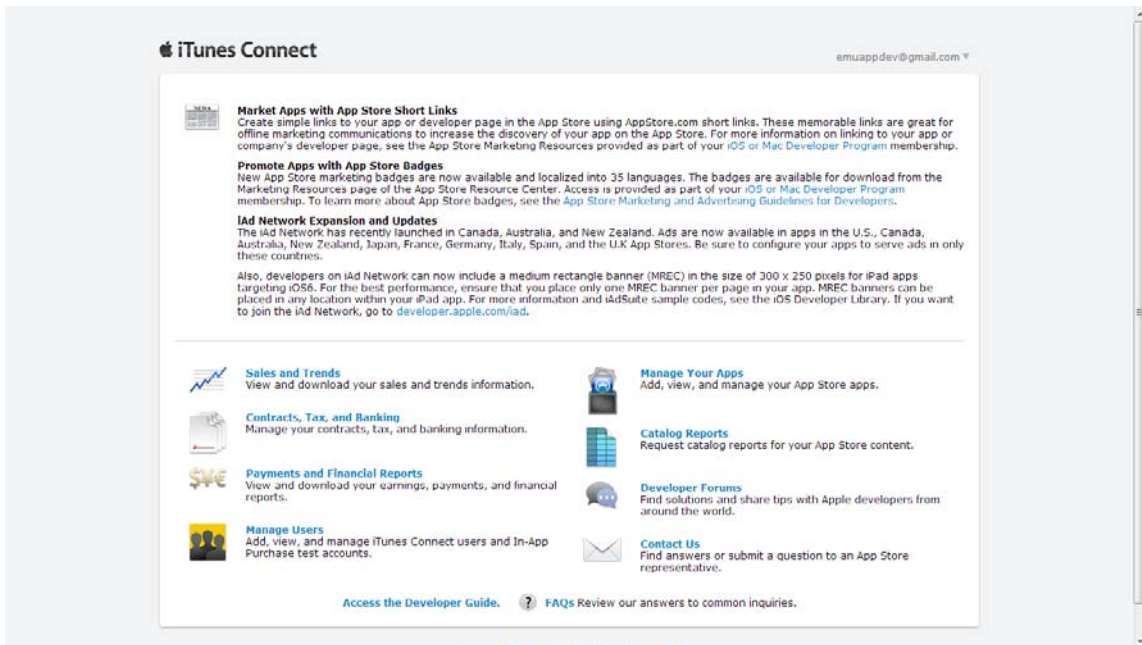


Figure 23. iTunes Connect Main Page

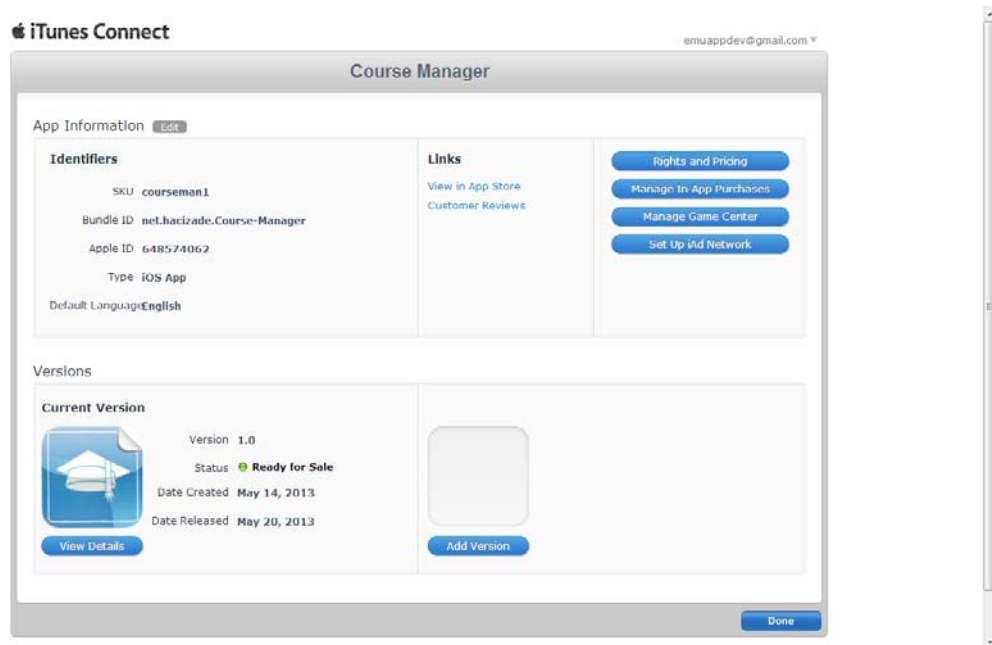


Figure 24. Course Manager Application Managing Page on iTunes Connect

The application is available for downloading from App Store and it can be viewed from <https://itunes.apple.com/us/app/course-manager/id648574062?ls=1&mt=8>. This page is illustrated in the following figure. It is compatible with all versions of iPhone, iPad and iPod. It is free and available for everyone.

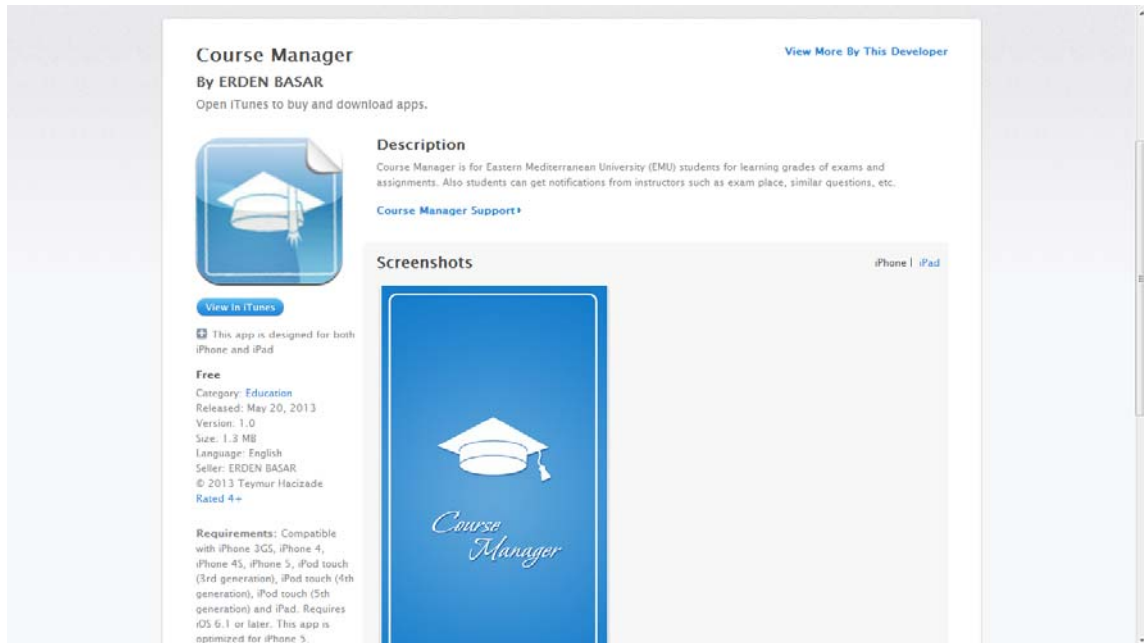


Figure 25. Course Manager Application in App Store

The latest versions of Xcode tool allow design the user interface of the application with story boards. It means that it is possible to see all views (pages) of the application in the same window and the connectors between these views. So Course Manager application is implemented with story board style and it is illustrated in the Figure 26.



Figure 26. iPhone Application Storyboard

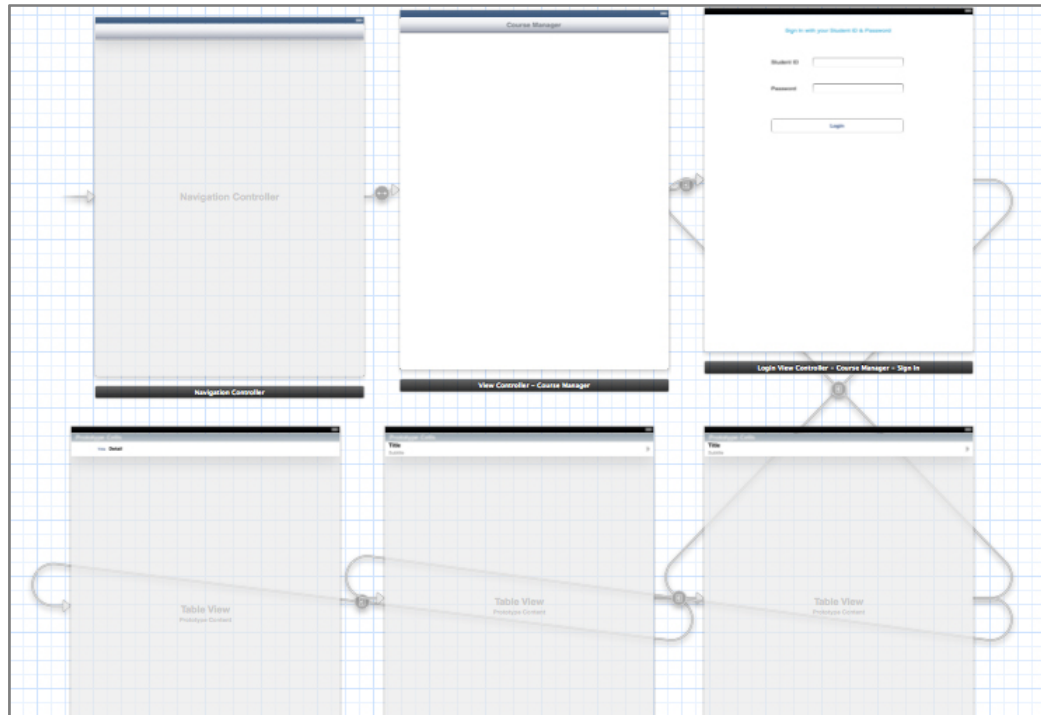


Figure 27. iPad Application Storyboard

For publishing the application iOS developer certificate was needed. It's price is \$99 for year.

4.5 Client Side Design with Business Objects

In previous chapters business objects are described in details. In this chapter it is shown how business objects are implemented in proposed system.

Client side classes are designed only with business objects. In proposed system the main business base objects are Student, Course, Exam, Assignment, AsssignmentSubmit, Grade, Lab, LabIP, Notification, ExamStudentPswd, FtpSettings classes. Also business business base list objects are exist. These object functions like list of objects. For example, StudentList object is the list of Student object.

Business object consists of 5 parts: business methods, validation rules, authorization rules, factory methods and data access. These parts are called regions in the code and it is shown below:

```
[Serializable]  
public class Course : Csla.BusinessBase<Course>  
{  
    #region Business Methods  
  
    #endregion  
  
    #region Business and Validation Rules  
  
    #endregion
```

```
#region Authorization Rules  
  
#endregion  
#region Factory Methods  
  
#endregion  
  
#region Data Access  
  
#endregion  
}
```

For explaining all regions in details let's see Course class as example. First thing has been done is describing business methods. The business methods region contains all member or instance field declaration, along with any business specific properties and methods. Business methods region of Course class is shown in Appendix E.

The Validation Rules region, overrides the `AddBusinessRules()` method in which `ValidationRules.AddRule()` is called to connect rule methods with properties. Custom rule methods for rules that aren't already available in `Csla.Validation.CommonRules` may also be included in the validation rules region or in our own library of rule methods. Validation Rules of the Course class is shown in Appendix F.

The Authorization Rules allow specifying the roles that can function with the object itself. One can define the roles that field allowed to create, get, edit, and delete the object.

In the Factory Methods region, static factory methods create, retrieve, and delete the object. For sure, these are just examples that one can edit as appropriate. It is important

to stitch the accepted parameters to match the identifying criteria for the particular business object. The factory methods region of the Course class is shown in Appendix G.

Finally, the Data Access area consists of DataPortal_XYZ methods. These methods must include the code to load defaults, retrieve object data, and insert, update, and delete object data, as appropriate. The DataPortal XYZ methods are DataPortal_Create, DataPortal_Fetch, DataPortal_Insert, DataPortal_Update, DataPortal_DeleteSelf, DataPortal_Delete with are called in factory methods. The data access region of course class is shown in Appendix H.

All these regions together make a whole business object class. The course class is called by all four client applications. Any change in this class automatically takes place in all applications, and there is no need change the user interface.

Chapter 5

CONCLUSION

Computer assisted examination systems become important field in education nowadays. It is frequently used in most universities, schools and educational centers. Computer assisted systems helps to test the knowledge of student more accurately that relational paper based exams.

Nowadays a lot of frameworks are developing for making flexible, secure, scalable applications. The aim of the new frameworks is to develop such system that is more usable than the systems before done. Business object architecture is one of the powerful technologies that developed in this field. The difference between business object architecture and relational architectures is its possibility to separate the system into layers. The practice shows that the applications that are developed with business object architecture are more scalable, secure and maintainable. It makes created system more reusable for further development. CSLA.NET framework is used to implement business object architecture.

The aim of the thesis is to show how business object architecture is implemented to computer assisted exam systems. Also one of the main purposes is to show how different platforms can use the same objects without any changes. All of the desktop, web and

mobile applications use the same business objects, make some operations and any change in these business object takes place in all platforms.

For the implementation of the whole system MS Visual C#, ASP.NET, Objective-C are used. The proposed system is tested and ready for use.

There are some challenges was faced to implement this application. Security is one of the main features in the proposed system.

First problem was disabling network and internet connection between student computers. They must not have any connection with each other and worldwide. Also system must prevent students to restore internet connection or network connection.

Second problem was faced with USB storages. Students are not allowed use any USB storage in the exam. The problem was that the Windows operating system disables only pre-installed devices and automatically enables all USB ports if detects one device.

Third challenge was related with mobile application. It must be iOS application. iOS application can be developed only in Mac OS X environment. Macbook Pro was bought for the development and developer certificate for publishing the application to App Store.

As a future work, Android and Windows Phone applications can be developed. The web service for mobile applications is already in use. So, only user interface and web service reader must be created for other mobile platforms.

Web application can be improved to be useful not only for assignment submission, also for learning grades and feedback to instructor.

Client interface of exam system can enhanced to support drag and drop for uploading and downloading. Server can log the operations like login, send, update, delete, insert that students do. Now proposed system is designed for laboratory exams. In future it can be developed for distance exams, home exams and etc.

REFERENCES

- [1] Rockford Lhotka, "Expert C# 2008 Business Objects". USA: Apress, 2009. p10-14
- [2] D. Patel, J. V. Sutherland, and J. Miller, Business Object Design and Implementation II: OOPSLA'96, OOPSLA'97 and OOPSLA'98 Workshop Proceedings. London: Springer-Verlag, 1998.
- [3] Fan Zhang, Yanbin Cui "Design of Distributed Warehouse Management System based on CSLA.NET" IEEE 2010 Second Pacific-Asia Conference on Circuits, Communications and System (PACCS), 2010. p293-294
- [4] "What is CSLA.NET ?" <https://github.com/MarimerLLC/csla/wiki/What-is-CSLA-.NET%3F> (Accessed on 23/04/2013)
- [5] <http://getyardstick.com/solutions/academic-exam-solutions-for-schools/>
- [6] <http://www.softwaresecure.com/Main.aspx>
- [7] <http://learn.examsoft.com/>
- [8] <http://examkiosk.in/index.html>
- [9] <http://www.exambuilder.com/index.htm>

[10] <http://www.proprofs.com/quiz-school/>

[11] Jeffrey A. Hoffer, V. Ramesh, Heikki Topi “Modern Database Management”, Tenth edition, Prentice Hall, 2011. p178-190.

[12] John Sharp, Jon Jagger, “Microsoft Visual C# .NET step by step”, Microsoft Corporation, 2002

[13] Stephen G. Kochan, “Programming in Objective-C”, Fourth edition, Addison-Wesley, 2012.

[14] Bülent Genç, “Design and Implementation of Software for Computer Aided Exams” MS Thesis, EMU, 2005.

[15] Xiaojian Liu, Xuejun Liu, Jianxin Li, “A Hybrid Language Combining Business Rules with Object-Orientation”, International Conference Computational Intelligence and Security, CIS’08, 2008, p101-106.

[16] E. Basar, B. Genc, “A Software System for Secure Computer Aided Exams”, Conference on Frontiers in Education: CS and CE, FECS’06, 2006, p24-30.

[17] A. Lodhi, G. Kassem, C. Rautenstrauch, “Modeling and Analysis of Business Processes using Business Objects”, 2nd International Conference Computer, Control and Communication, 2009, p1-6.

[18] Huang Qing, Cai Hou-xin, “An Application of the Code Specification based on CSLA”, 2nd IEEE International Conference Information Management and Engineering (ICIME), 2010, p509-512.

[19] An Hongxin, Xu Yusheng, Ma Zhixin, Liu Li, “Integrating User Interfaces by Business Object States”, 2nd International Conference Information Science and Engineering (ICISE), 2010, p2900-2903.

APPENDICES

Appendix A: User Documentation for Exam Instructor's Program

In this section the way of installing, using and detailed user documentation of the system is shown.

For installing the program for instructors, setup file called ExamProSetup.msi is available. The installer is usual one step by step with next buttons and finish at the end.

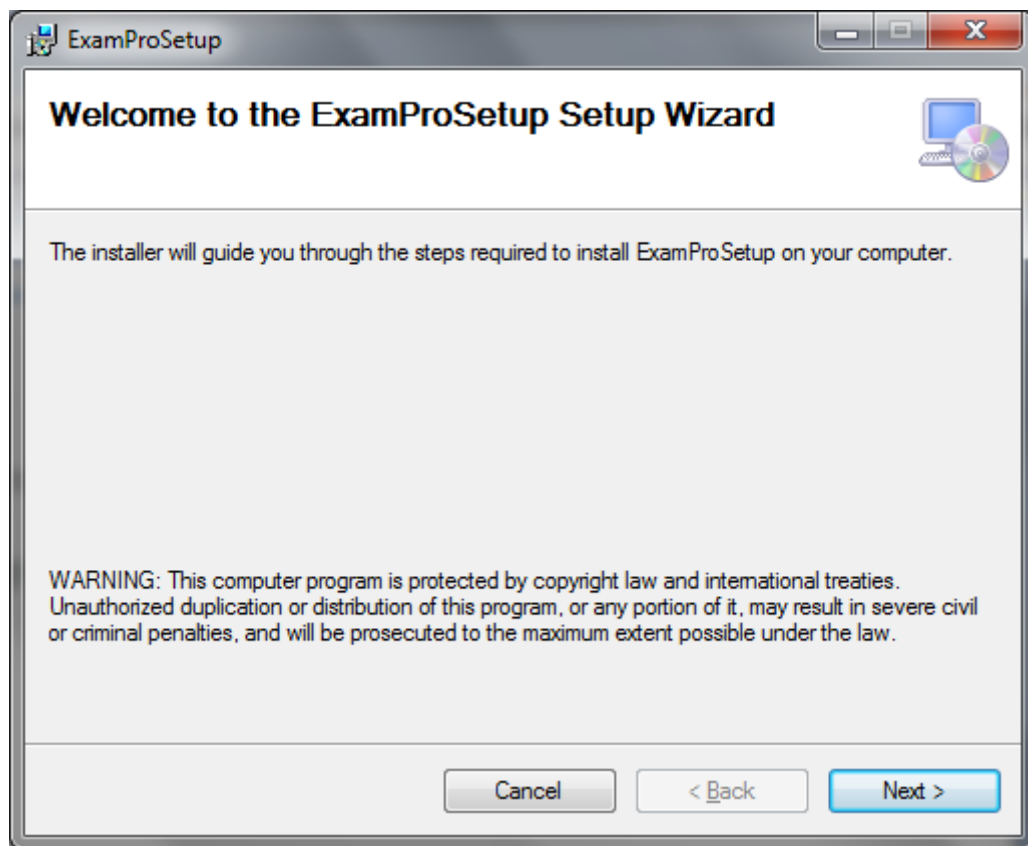


Figure 28. ExamPro Setup Page

It is recommended not change the installation path of the program. But if you decide to install it another place after installation you must change configuration file.

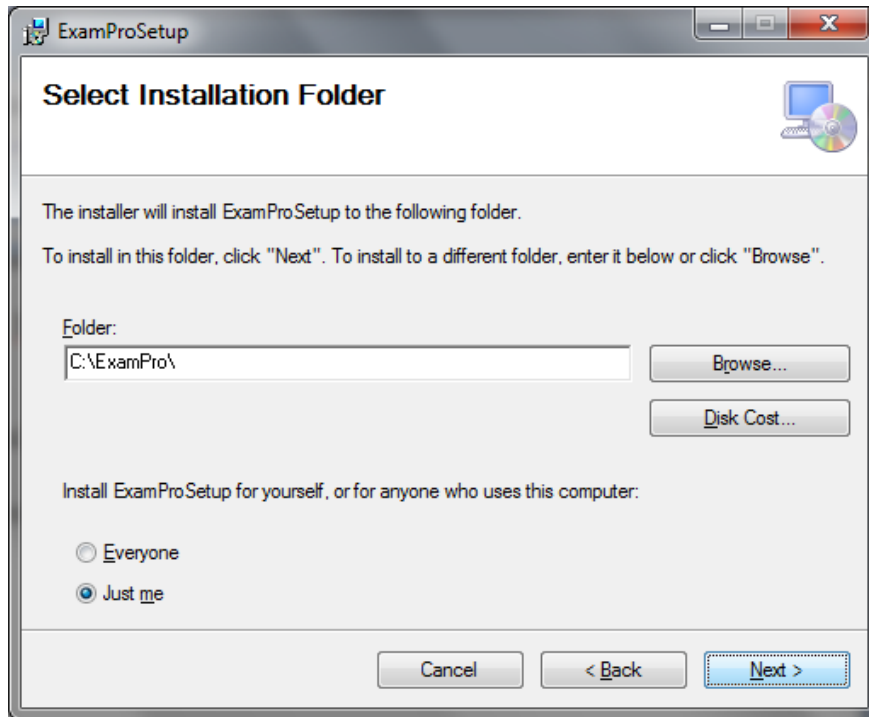


Figure 29. ExamPro Selecting Installation Folder

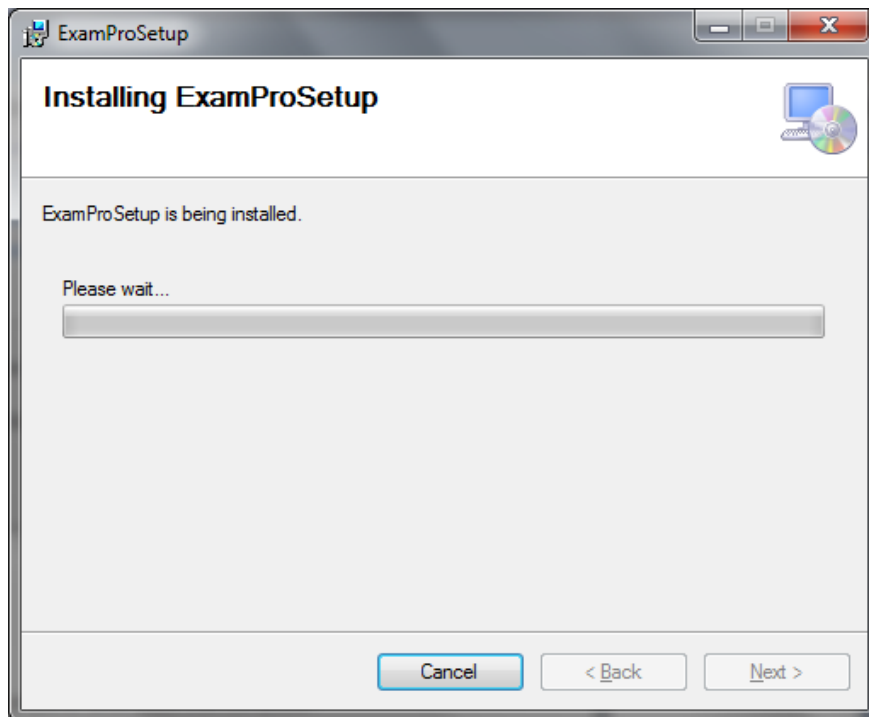
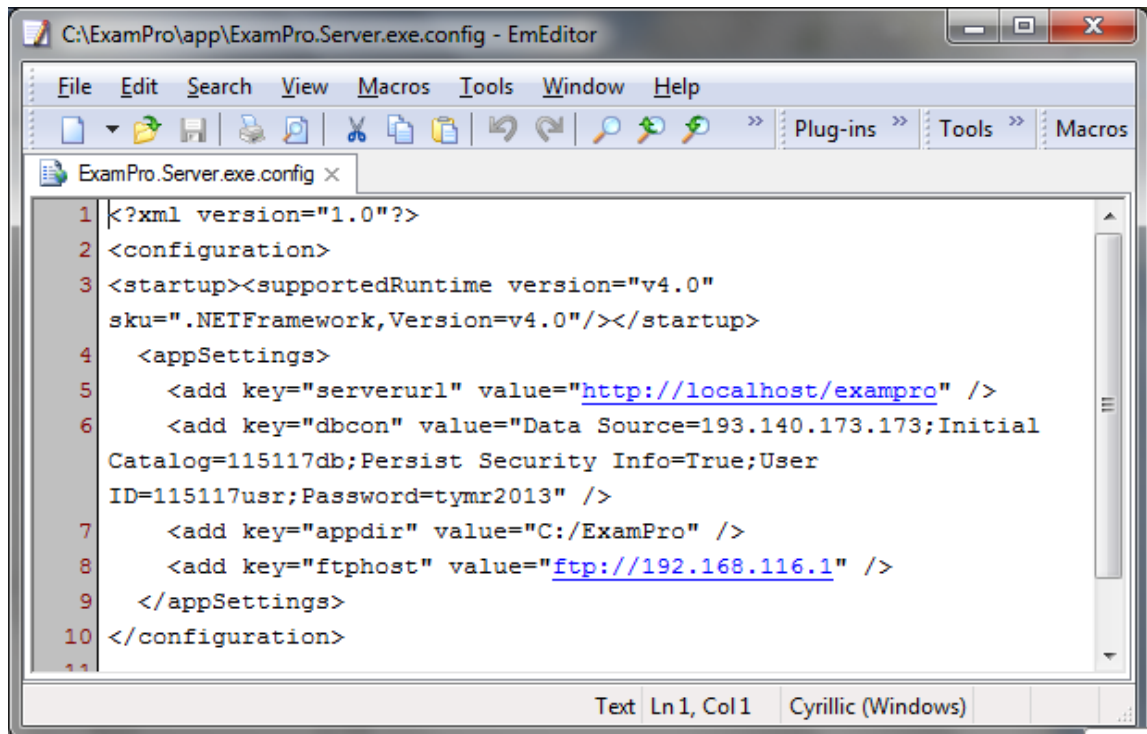


Figure 30. ExamPro Installation Process

After installation finishes it creates shortcuts in desktop and start menu. If you need to change any configuration of the program after installation you can access configuration file from C:\ExamPro\app\ExamPro.Server.exe.config. The extension of this file is .config and the content of the file is shown below in Figure 31.

The image shows a screenshot of a text editor window titled "C:\ExamPro\app\ExamPro.Server.exe.config - EmEditor". The window contains an XML configuration file with the following content:

```
1 <?xml version="1.0"?>
2 <configuration>
3 <startup><supportedRuntime version="v4.0"
  sku=".NETFramework,Version=v4.0"/></startup>
4 <appSettings>
5 <add key="serverurl" value="http://localhost/exampro" />
6 <add key="dbcon" value="Data Source=193.140.173.173;Initial
  Catalog=115117db;Persist Security Info=True;User
  ID=115117usr;Password=tymr2013" />
7 <add key="appdir" value="C:/ExamPro" />
8 <add key="ftphost" value="ftp://192.168.116.1" />
9 </appSettings>
10 </configuration>
11
```

The status bar at the bottom of the editor shows "Text Ln 1, Col 1" and "Cyrillic (Windows)".

Figure 31. ExamPro Configuration File

Configuration file is XML file and structured by key value pairs. The description of keys are shown below:

- **serverurl** – URL address of web application (assignment system). This is for uploading and downloading files to that server.
- **dbcon** – SQL Server's default connection string.

- **appdir** – the folder where program is installed. If you change the installation path during the installation you must change this value too. Otherwise the program will not work properly.
- **ftphost** – the host address of the FTP server.

When you open program the window with menus comes to front.

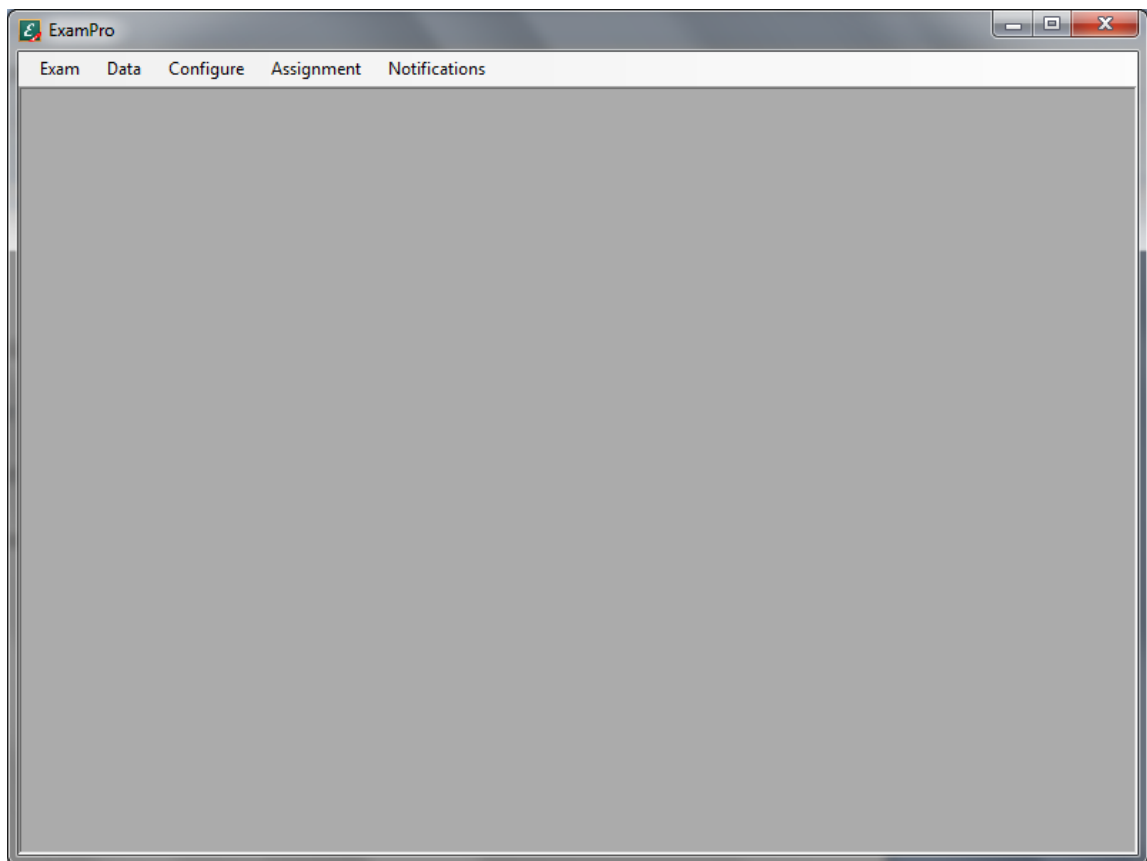


Figure 32. ExamPro Default Window

There are five menus: Exam, Data, Configure, Assignment and Notification. Every menu contains submenus.

The first thing is adding courses to the system. For accessing the following window

Data → Courses

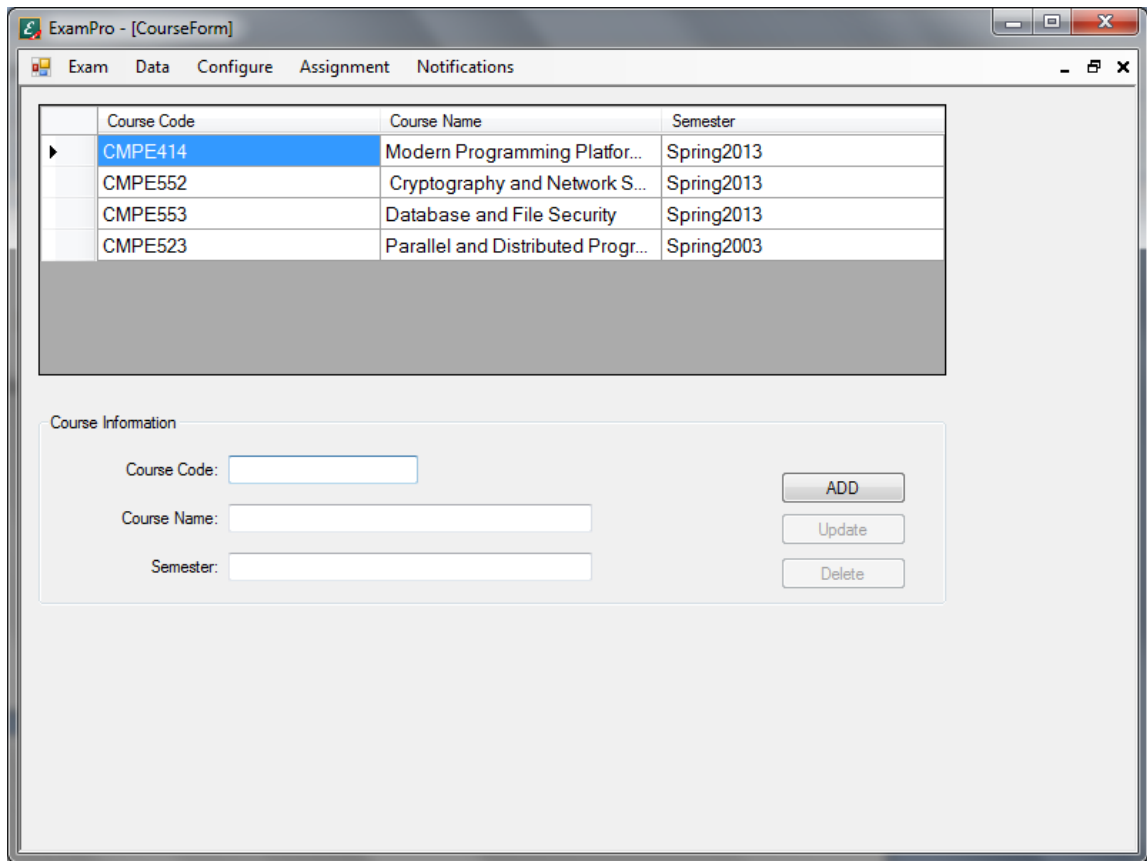


Figure 33. ExamPro Courses Window

For adding new course you must fill the Course Code, Course Name and Semester fields. Existing courses appear in the grid on the top. For updating or deleting course you must double click on the course you want to edit.

For managing students by courses you must open student controlling window from **Data→Student**. It is shown in next figure.

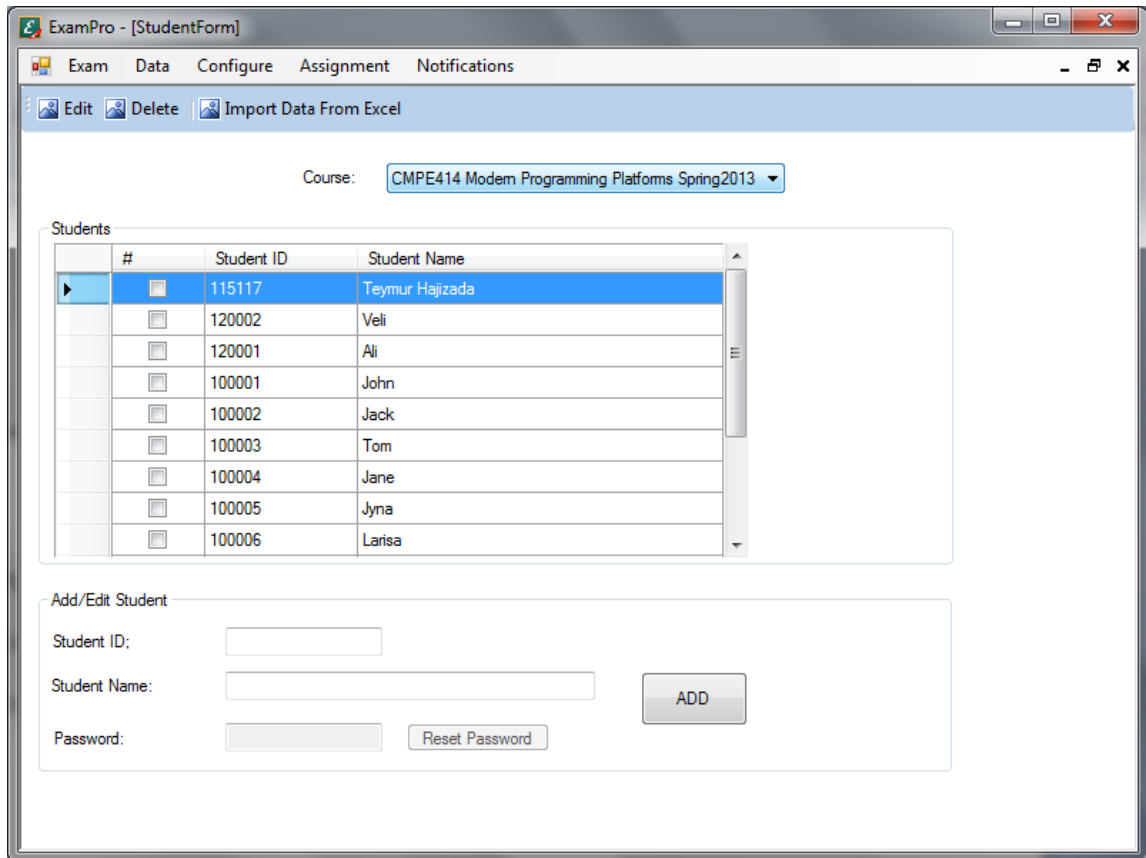


Figure 34. ExamPro Students Window

First the course must be selected. Then students list with student id and student name automatically is shown in grid. It is possible delete more than one student by checking them with checkbox. From Add/Edit Student panel it is possible to add new students, update information of existed student and reset the student password. Also there is opportunity import the list of students from excel file. First excel file must be saved as .CSV format then it can be easily imported to the list.

After defining course and student there are some configurations that must be done before starting exam. One of them is defining laboratory and computers in laboratory. It is accessed from **Configure→Labs**. Lab Information panel is for adding, editing and

deleting the laboratory. After new laboratory is defined, it appears in lab list. “Define IPs for Lab” panel is for adding computers to the laboratory by defining their IP addresses. First the laboratory must be selected from the list and then IP range must be defined in respective fields. It is possible to add only one IP address by leaving the second field empty. All the computers that are in selected laboratory are listed in grid. Grid contains information about lab id, lab name, IP address and computer id. CompID is unique number within the laboratory.

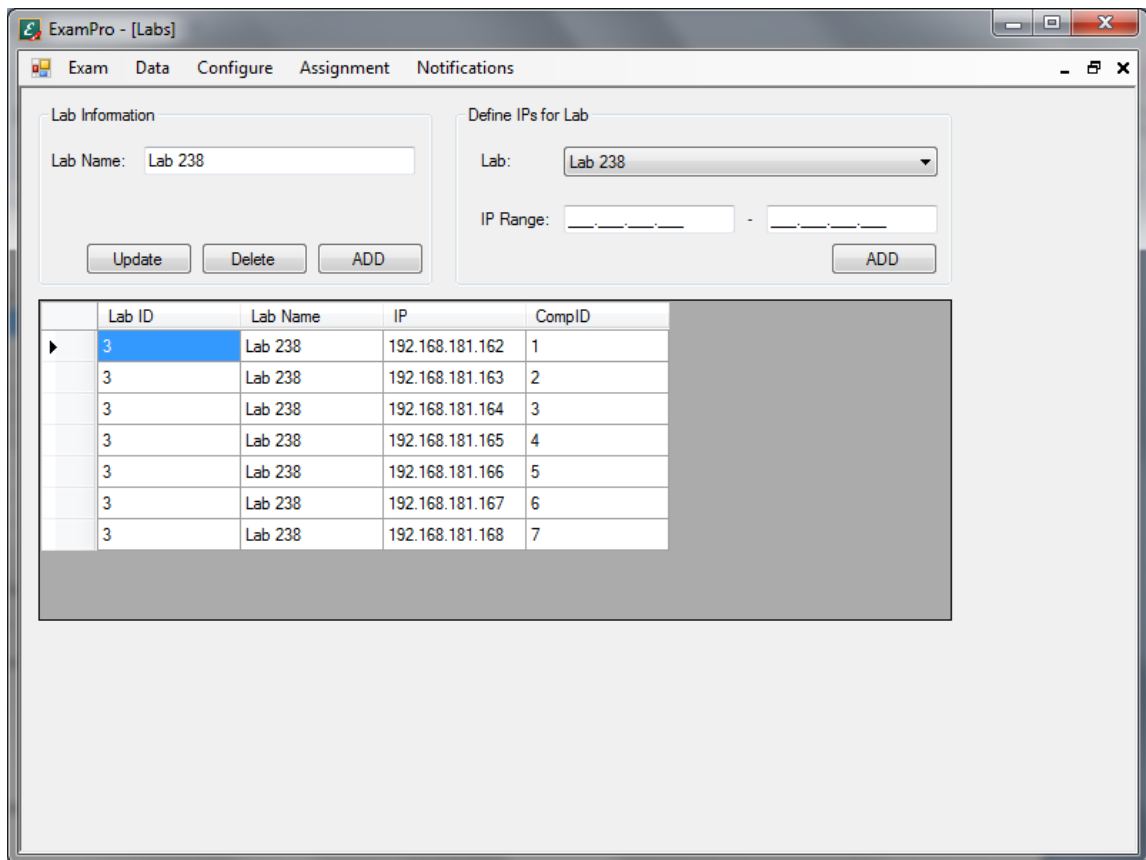


Figure 35. ExamPro Laboratories Window

FTP servers setting are defined in window which is accessible from **Configure→Ftp Settings**.

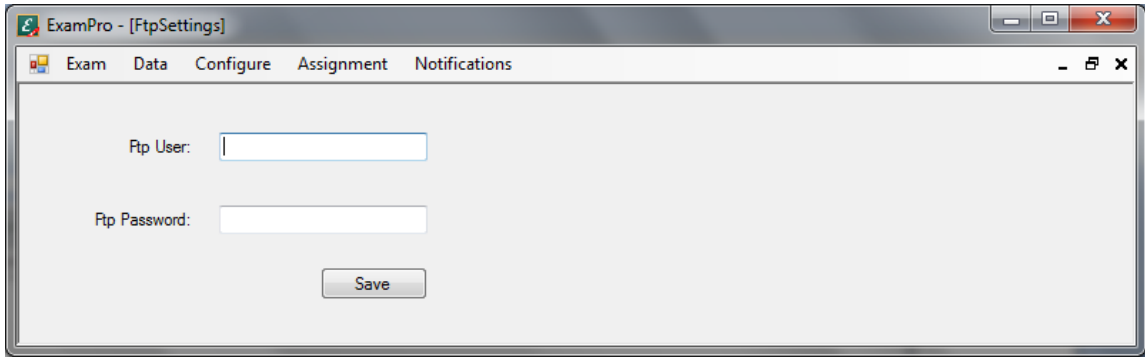


Figure 36. ExamPro FTP Settings Window

Before exam instructor must define new exam. For exam definition **Exam→Define** is available. Instructor can choose new one or select the existing defined exam from the list.

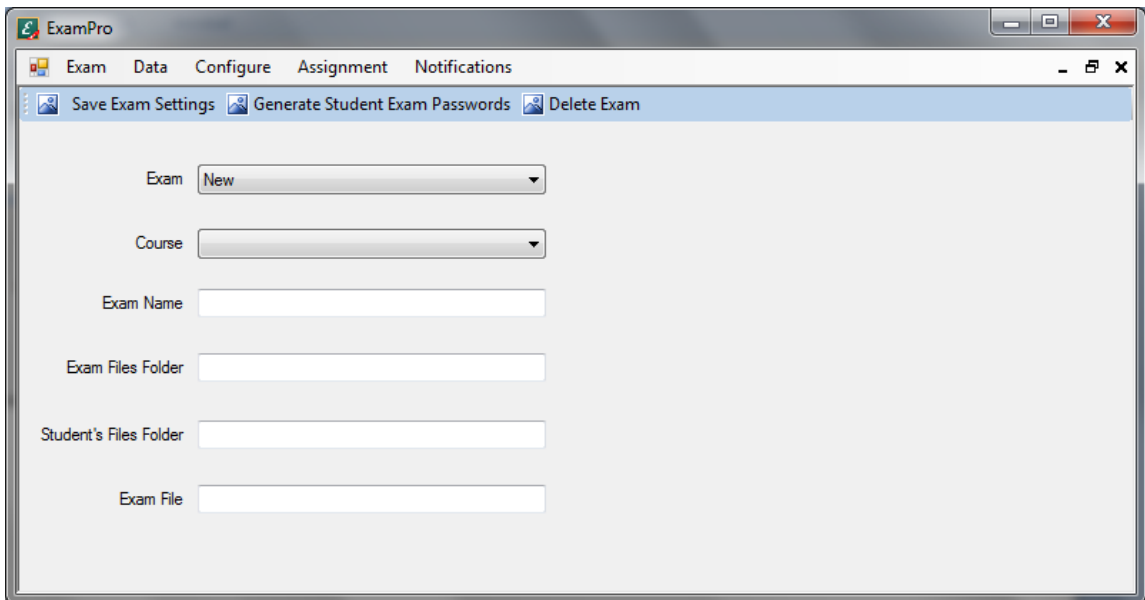


Figure 37. ExamPro Exam Definition Form

Once instructor selects new exam s/he fills the fields shown in figure 37. After saving the exam s/he can generate student passwords for the exam and delete the existing exam.

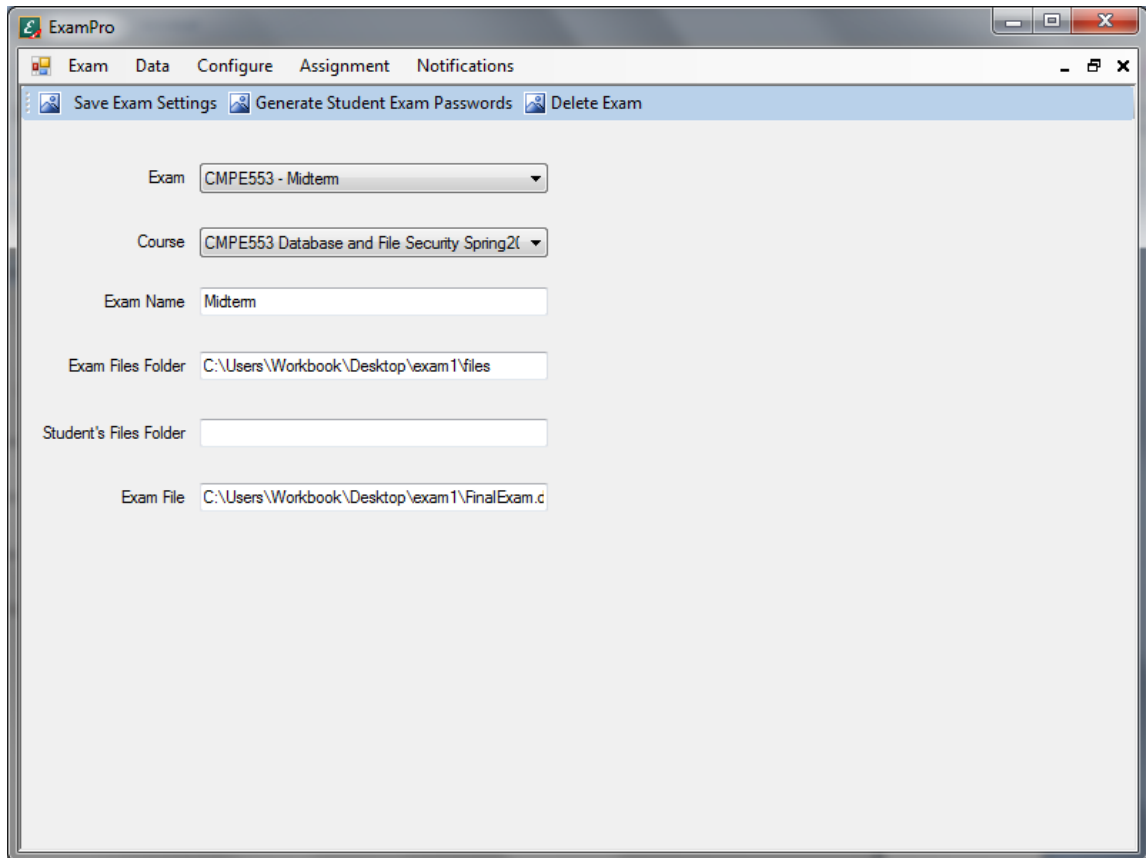


Figure 38. Exam Definition Form for Selected Exam

If instructor clicks the Generate Exam Student Passwords button the form shown in figure 39 comes front. Instructor can print the list of students with their passwords and sit places in exam. Also s/he can send notification by clicking Send to Mobile App button. When instructor clicks this button program automatically sends notification to all students who take selected course which contains the information shown in table.

The screenshot shows a window titled "Exam Students Password". At the top, there are two buttons: "Print" and "Send to Mobile App". Below these buttons is a table with three rows and four columns. The table contains the following data:

111111	Tim	K872AE	Comp: 1
222222	William	PSOA12	Comp: 3
333333	Emre	OAU44T	Comp: 2

Figure 39. Students' Passwords for the Exam

Exam control form is accessible from **Exam**→**Control** menu. In Exam Information Panel instructor selects course, exam, exam room and defines the duration of the exam. Exam list contains the exams that are defined in Exam Definition form shown in figure 38. After controlling the correctness of the information Save Settings button must be clicked. These settings must be saved before students open ExamPro Client application. The list of computers and information about their status is shown in grid after clicking List All Clients button. Then Start Exam button must be clicked. After exam finished Finish Exam button must be clicked.

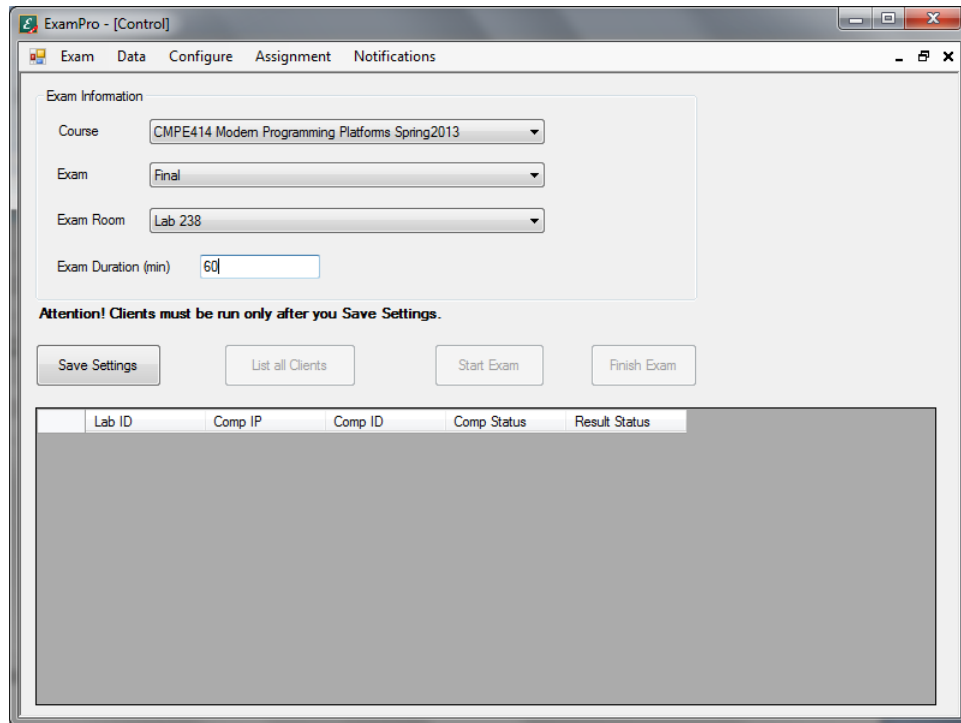


Figure 40. Exam Control Form

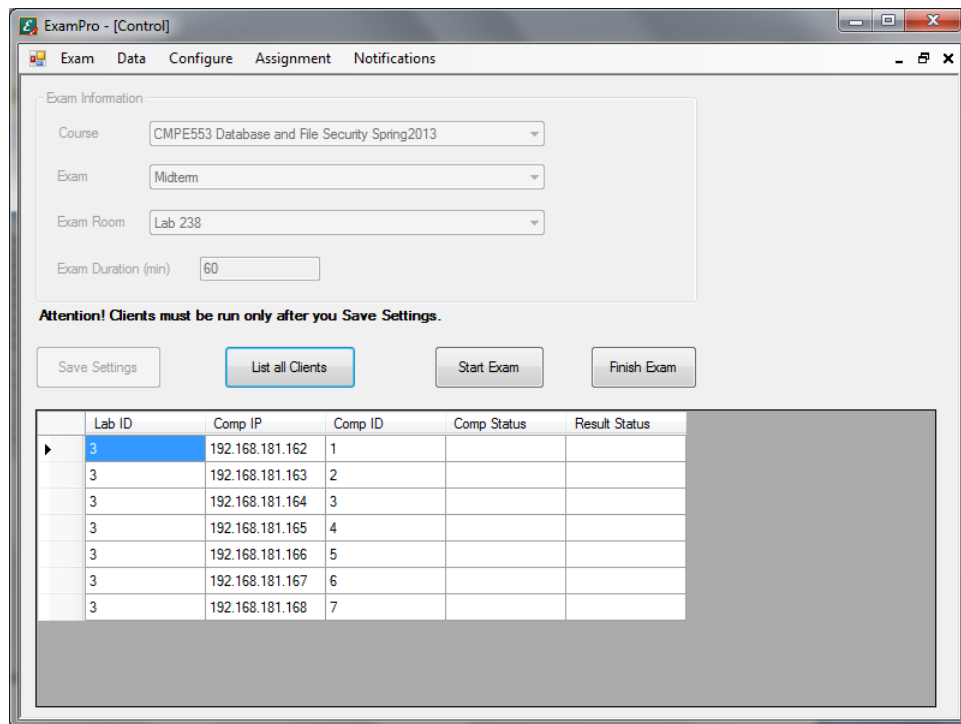


Figure 41. Exam Control Form with List of Clients and Their Statuses

After exam finishes exam can be evaluated. It is accessible from **Exam→Evaluation** menu. After course and exam are selected from the respective lists the list of students is shown in grid. Student Files and Evaluation panel is enabled by double clicking the student from the grid. It is possible to download the files of students one by one and all files at a time by clicking Download all Solutions button. Instructor adds grades for the appropriate questions and then clicks Save button in the bottom. Also it is available to the table of students with their grades by clicking Export Grades button. The table can be refreshed and exported to the CSV file.

The screenshot shows the 'ExamPro - [Evaluation]' application window. At the top, there is a menu bar with 'Exam', 'Data', 'Configure', 'Assignment', and 'Notifications'. Below the menu bar, there are two dropdown menus: 'Course' set to 'CMPE414 Modern Programming Platforms Spring2013' and 'Exam' set to 'Quiz 1'. To the right of these dropdowns are two buttons: 'Export Grades' and 'Download all solutions'.

The main area is divided into two panels. The left panel contains a table with the following data:

Student ID	Student Name
115117	Teymur Hajizada
120002	Veli
120001	Ali
100001	John
100002	Jack
100003	Tom
100004	Jane
100005	Jyna
100006	Larisa
100007	Natasha
100008	Ivan
100009	Emre
100010	Murat
100011	Ayşe

The right panel is titled 'Student Files and Evaluation'. It has a 'Student:' label and two buttons: 'Download Files' and 'Show Student Files'. Below these is a section for adding a grade, with a '#' dropdown set to '1', a 'Grade' input field, and an 'Add' button. At the bottom of this panel is a 'Delete Selected' button.

At the very bottom of the application window is a large 'SAVE' button.

Figure 42. Exam Evaluation Form

ExamPro - [Evaluation]

Exam Data Configure Assignment Notifications

Course: CMPE414 Modern Programming Platforms Spring2013

Exam: Quiz 1

Export Grades Download all solutions

Student ID	Student Name
115117	Teymur Hajizada
120002	Veli
120001	Ali
100001	John
100002	Jack
100003	Tom
100004	Jane
100005	Jyna
100006	Larisa
100007	Natasha
100008	Ivan
100009	Emre
100010	Murat
100011	Ayşe

Student Files and Evaluation

Student: 115117 Teymur Hajizada

Download Files Show Student Files

1 Grade Add

Number	Grade
1	98
2	96
3	100
4	99

Delete Selected

SAVE

Figure 43. Exam Evaluation Form with Grades.

EvaluationExportForm

Refresh Export to CSV

100004	Jane	90	15	56	49
115117	Teymur Hajizada	98	96	100	99
120002	Veli	60	35	75	90

Figure 44. Exam Evaluation - Export Grades Form

Assignment management form can be accessed from **Assignment**→**Assignment Control** menu. Assignment Information panel contains information like assignment name, end date time, course, description and attachment files about the assignment. Assignments list is shown in grid. It can be updated or deleted by double clicking on appropriate assignment in the grid.

The screenshot shows a software window titled "ExamPro - [AssignmentForm]". The window has a menu bar with "Exam", "Data", "Configure", "Assignment", and "Notifications". The main area is divided into two sections: "Assignment Information" and "Assignments".

Assignment Information

Assignment Name:

End Date Time: 29.05.2013 13:28

Course:

File 1: File 4:

File 2: File 5:

File 3:

Description:

Assignments

	Name	End Date	Course
▶	Assignment 1	24.04.2013 12:27	Modern Programming Platforms

Figure 45. Assignment Control Form

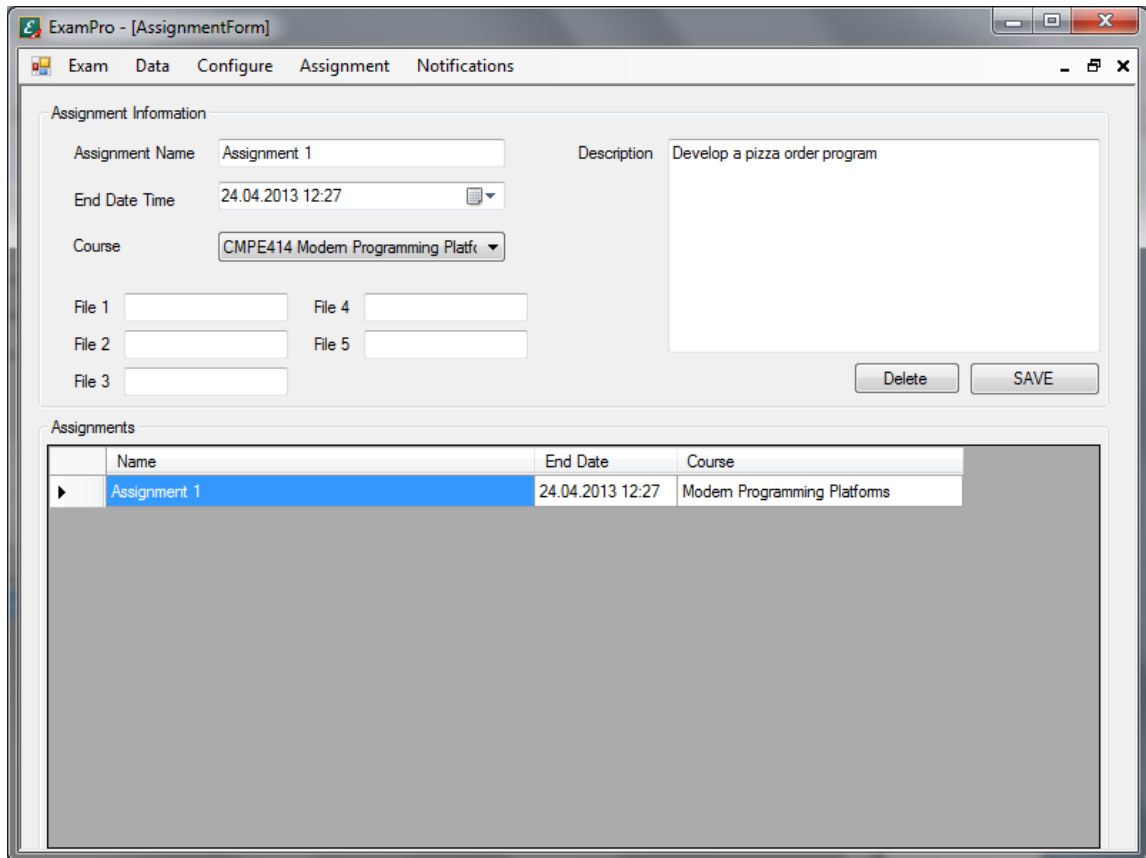


Figure 46. Assignment Control Form – Editing

Assignment evaluation can be accessed from menu **Assignment**→**Assignment Evaluation**. First assignment must be selected from the list then the list of students who submitted solutions to the assignment are shown in the grid. Student Files and Evaluation panel become enabled when the student is selected from the grid by double clicking. The evaluation process is like in exam. The grades for criteria are added and then Save button must be clicked. The screenshot of these forms are shown below:

ExamPro - [AssignmentEvaluationForm]

Exam Data Configure Assignment Notifications

Select Assignment for Evaluation 03.06.2013 13:31 Assignment 2 Modern Programming Platforms

	Student ID	Student Name	Submit Date
▶	115117	Teymur Hajizada	29.05.2013 13:32

Student Files and Evaluation

Show Student Files

1 Grade Add

Grades List:

Number	Grade
--------	-------

DELETE Selected

SAVE

Figure 47. Assignment Evaluation Form

ExamPro - [AssignmentEvaluationForm]

Exam Data Configure Assignment Notifications

Select Assignment for Evaluation 03.06.2013 13:31 Assignment 2 Modern Programming Platforms

	Student ID	Student Name	Submit Date
▶	115117	Teymur Hajizada	29.05.2013 13:32

Student Files and Evaluation

Show Student Files

4 Grade Add

Grades List:

Number	Grade
1	90
2	95
3	100

DELETE Selected

SAVE

Figure 48. Assignment Evaluation Form – Evaluation Process

The last form is notifications which are available from **Notification→Mobile Notifications** menu.

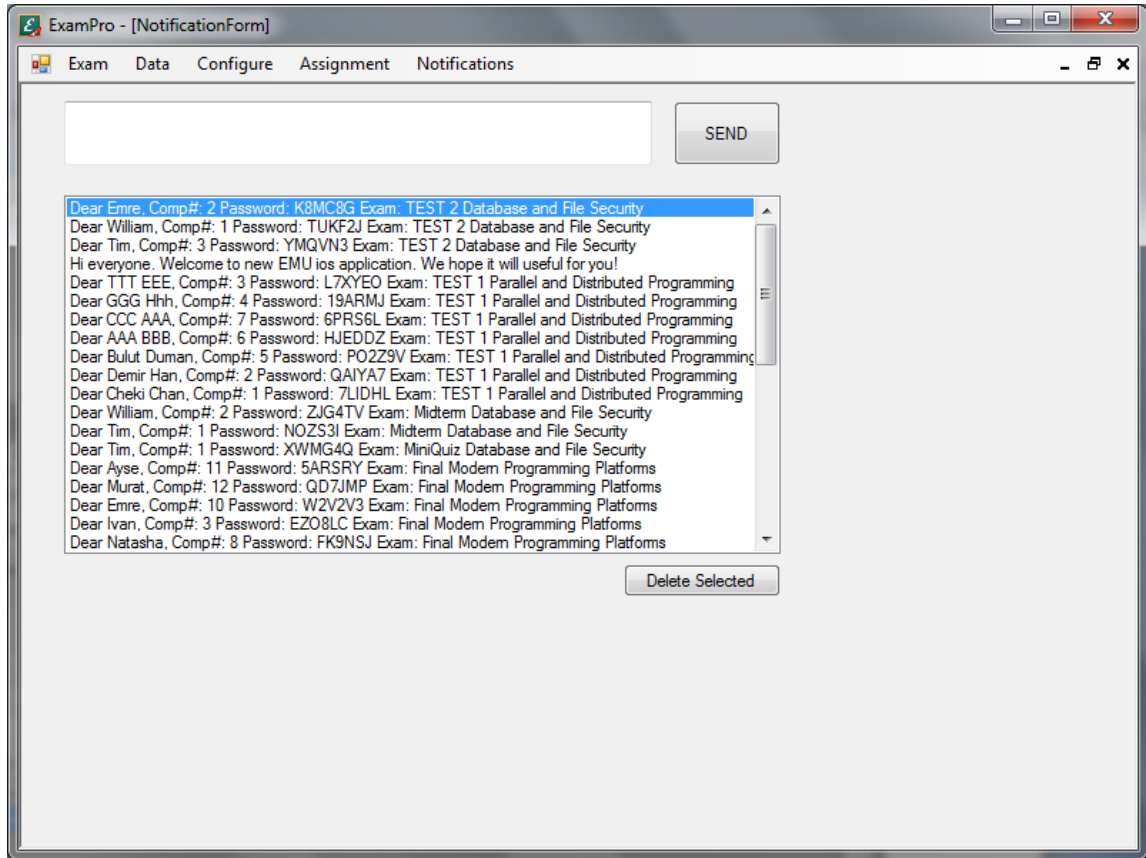
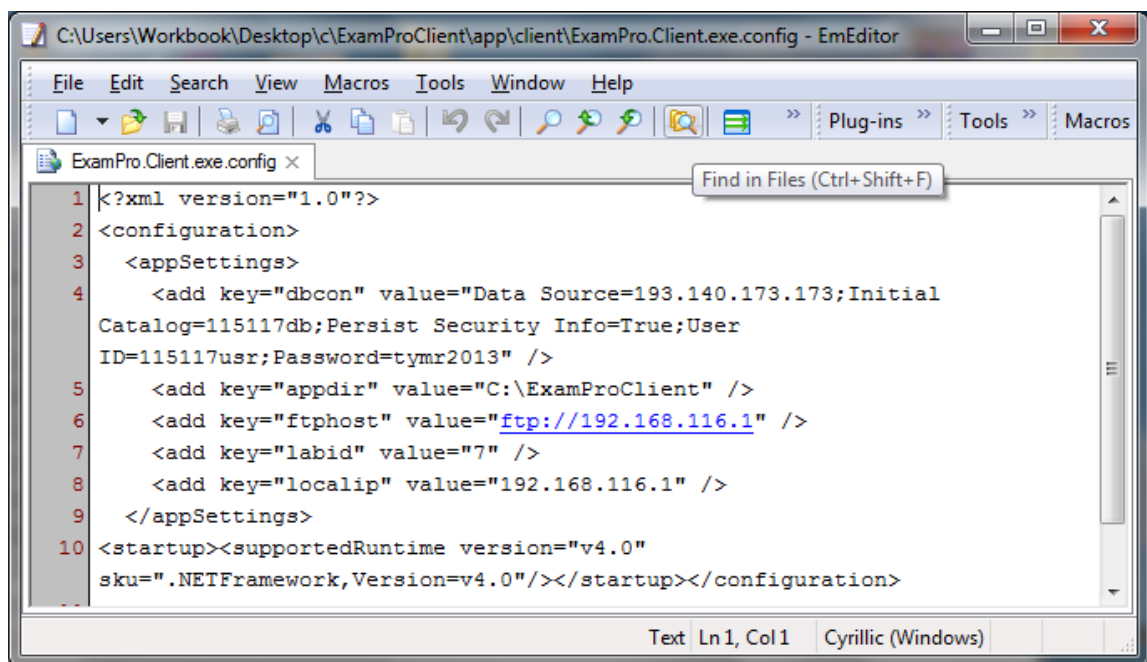


Figure 49. Mobile Notifications Form

Appendix B: User Documentation for Exam Client Application

Exam program for students can be installed with ExamProClientSetup.msi setup file. The installation is simple like a program for instructors. During installation don't forget to check install for all users radio box. It is recommended not change the installation path, but if it is changed it can be configured from configuration file. The configuration file is described below in Figure 50.



```
1 <?xml version="1.0"?>
2 <configuration>
3   <appSettings>
4     <add key="dbcon" value="Data Source=193.140.173.173;Initial
Catalog=115117db;Persist Security Info=True;User
ID=115117usr;Password=tymr2013" />
5     <add key="appdir" value="C:\ExamProClient" />
6     <add key="ftphost" value="ftp://192.168.116.1" />
7     <add key="labid" value="7" />
8     <add key="localip" value="192.168.116.1" />
9   </appSettings>
10 <startup><supportedRuntime version="v4.0"
sku=".NETFramework,Version=v4.0"/></startup></configuration>
```

Figure 50. Exam Client Program Configuration File

Configuration file contains XML string which is designed in key value pairs. The description of keys is explained below:

- **dbcon** – SQL Server default connection string
- **appdir** – the installation path of program.
- **ftphost** – FTP Server's host address.

- **labid** – the id of the laboratory. It can be found in instructor’s program Laboratory form. When you select library, the library id is shown in grid. It is very necessary to define correct labid value, otherwise program will not work properly.
- **localip** - IP address of the machine. Sometimes computers can connect to several networks, so they can have more than one IP address. In this case localip value must be defined. If computer connects only to one network this value can be leaved empty. System automatically finds the IP address if there is only one.

First thing is running ExamPro Launcher program for starting exam. This application disables all network connections by adding rules to the firewall, disables USB storage usage, sets password to administrator, creates new Windows user named Exam and logs off the system. When this program is run the following window comes front:



Figure 51. ExamPro Launcher Program

After system logs off the Windows users are shown to the student. S/he can choose only Exam user. Student cannot open system as administrator because s/he does not know administrator's password. When the OS loads student must open ExamPro Client application:

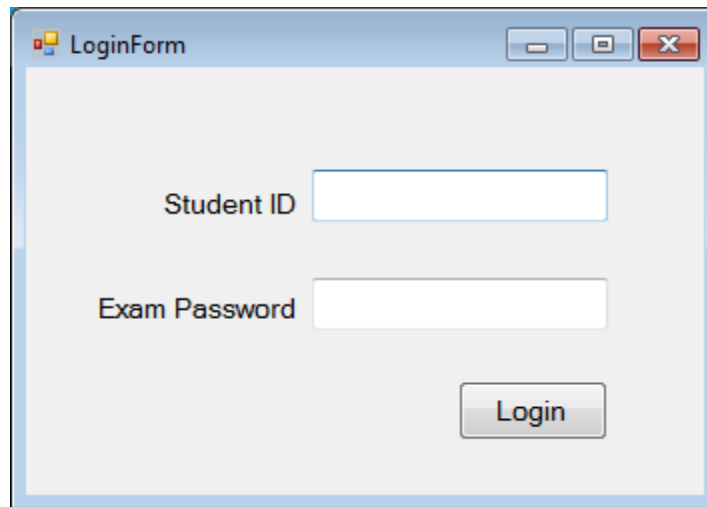


Figure 52. ExamPro Client Login Window

Student enters student id and exam password that is given by instructor or got from iOS application.

When the main form of application is opened student id, time, download and upload sections are shown. Student can access the exam files direct from download panel or by clicking button from file explorer.

Student can add unlimited files and folders by clicking ADD button. Also s/he can remove files from the list with Delete button.

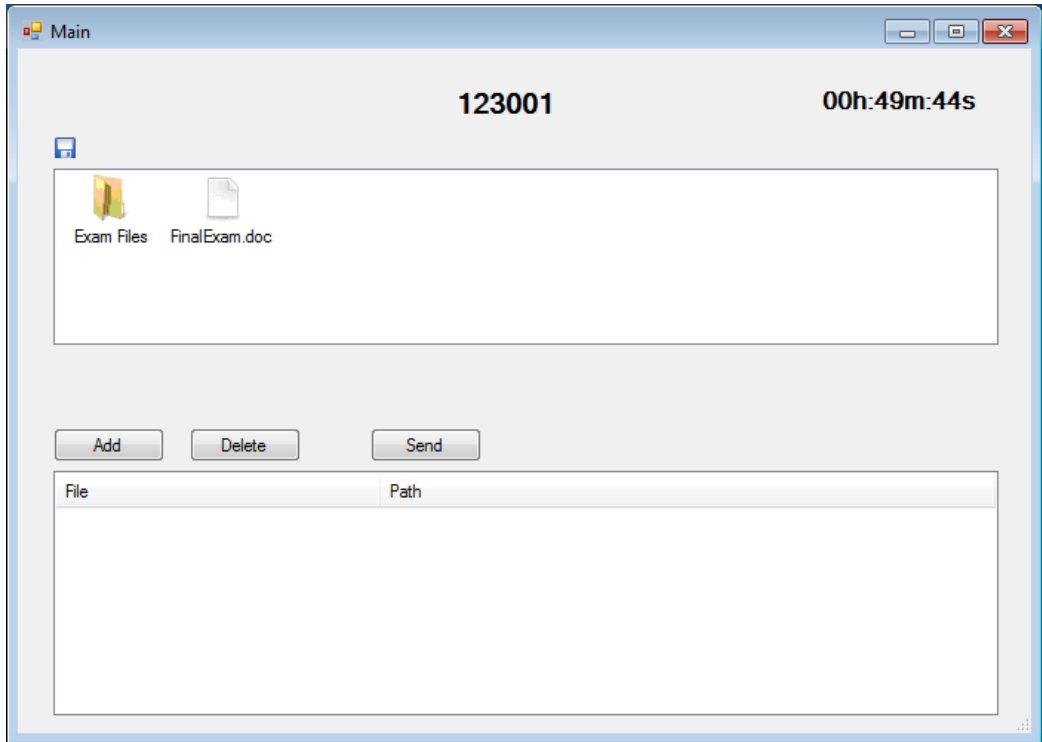


Figure 53. ExamPro Client Main Form

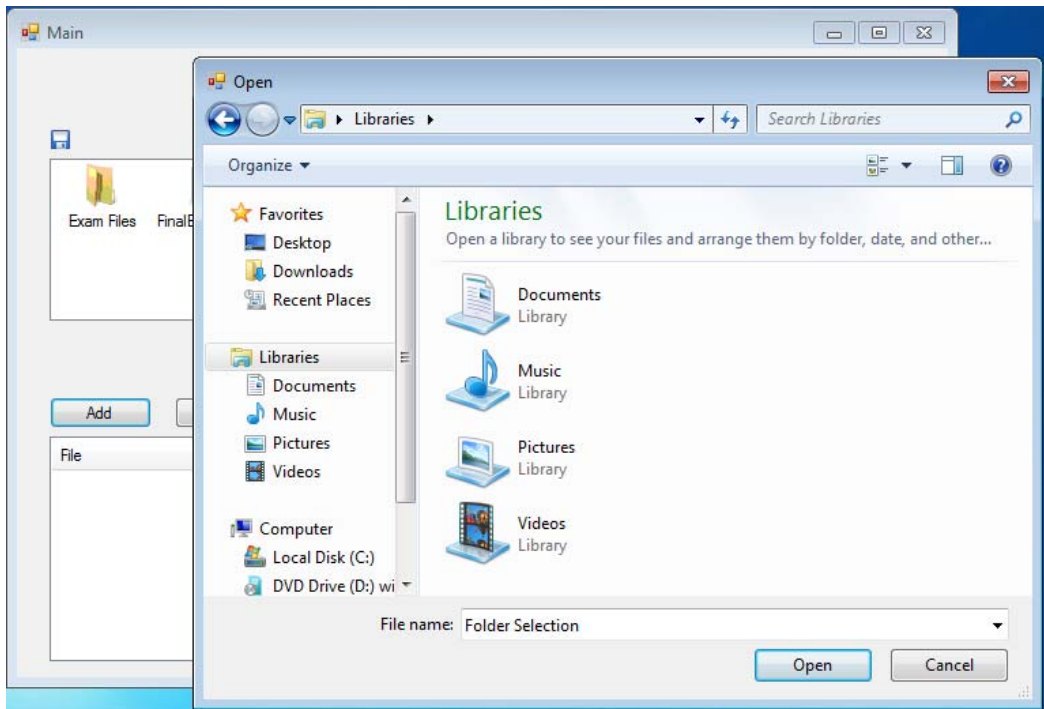


Figure 54. ExamPro Client – Add Files

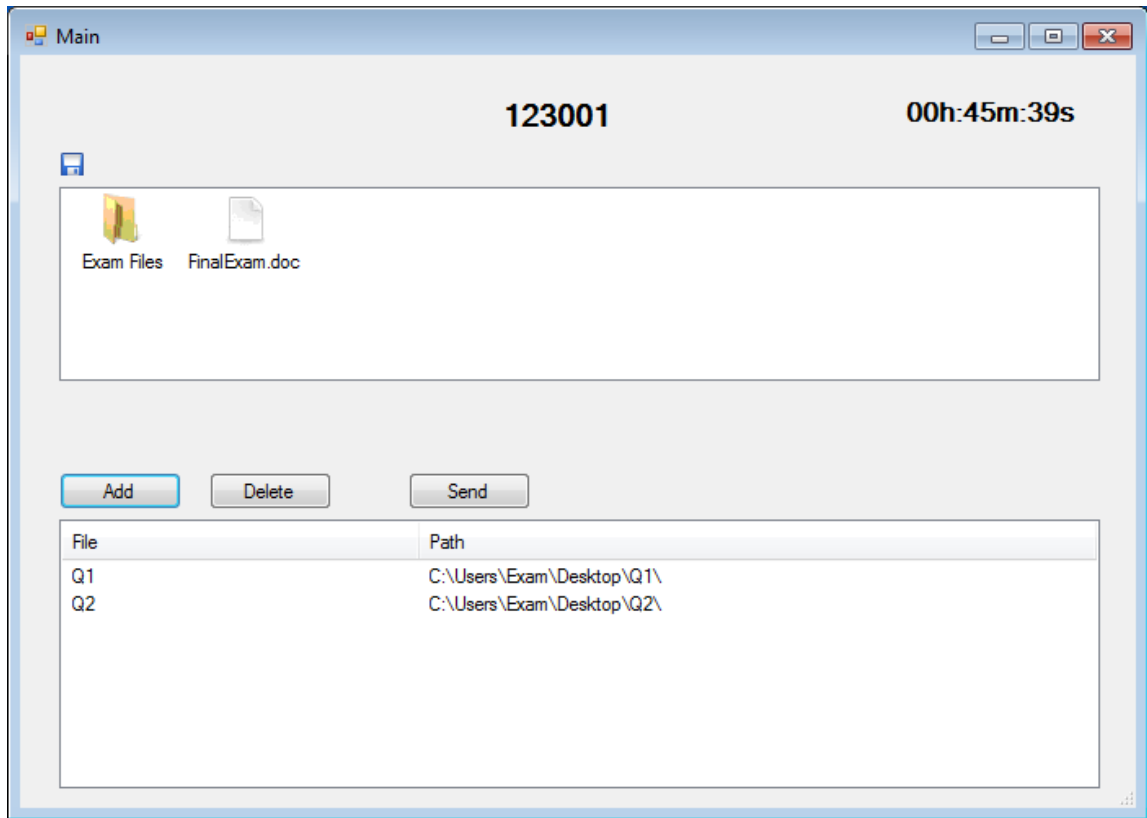


Figure 55. ExamPro Client Application with Files

After adding all solutions student can send them by clicking Send button. When s/he clicks Send button s/he gets dialog window which asks if student is sure to submit. Because student cannot submit solutions second time. System prevents such action. If s/he submitted successfully the form shown in figure 56 comes to front:

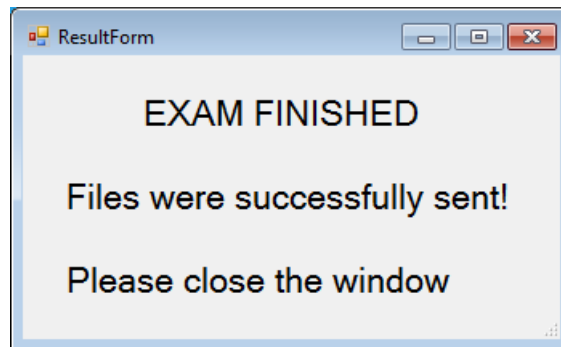


Figure 56. ExamPro Client Result Form

Appendix C: User Documentation for Assignment Submission System

Assignment submission application is available from web. Now it is available from the URL address <http://students.emu.edu.tr/115117>. Student can login to the system with student id and the password given by instructor.

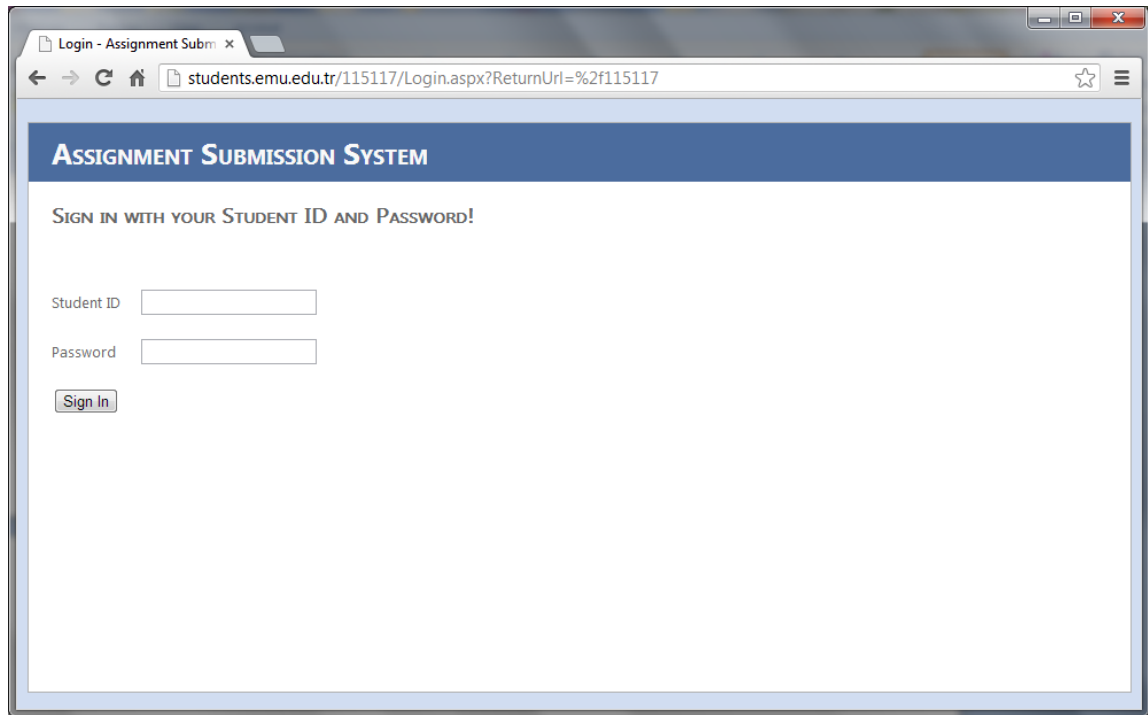


Figure 57. Assignment Submission System – Login Page

When student signs in successfully to the system the list of assignments is shown to the student. This table contains information like name of assignment, end date and course name. For details of the assignment and for submission student can click Show link behind the assignment row that s/he wants to see.

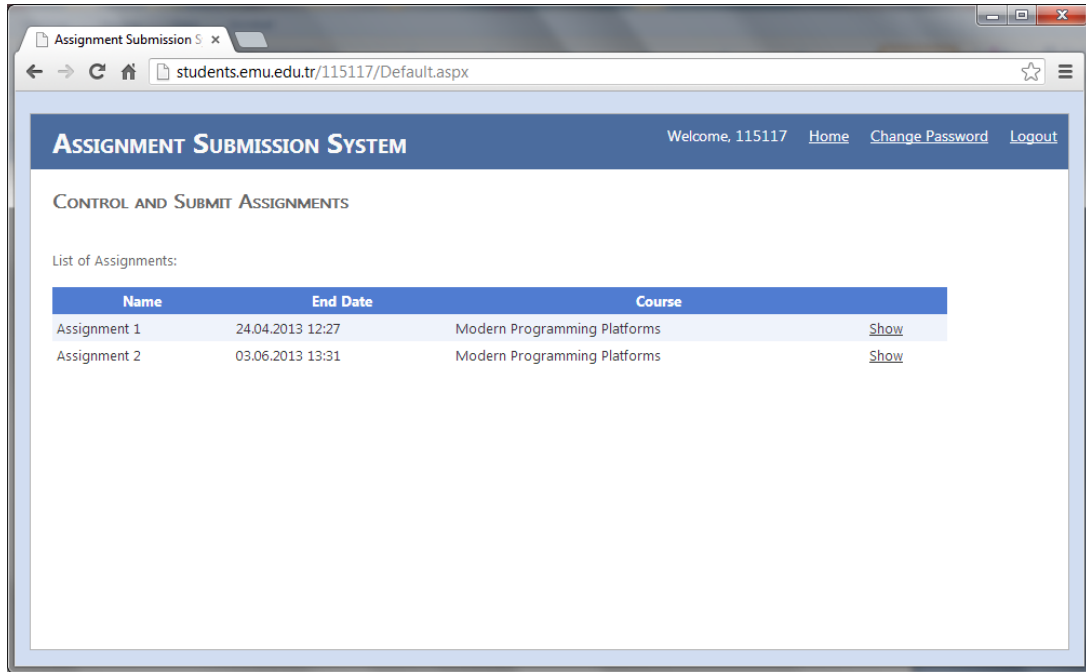


Figure 58. Assignment Submission System – Assignments List Page



Figure 59. Assignment Submission System – Control and Submit Page

After reading the description of the assignment and solving the assignment student can submit the assignment. First from upload file section s/he can choose solution files up to 5 files. If there are more than 5 files he can archive it and submit in zip format. After uploading, student must click Submit button. Then student gets message about the submission. If everything is alright s/he gets message “Assignment was submitted successfully”.

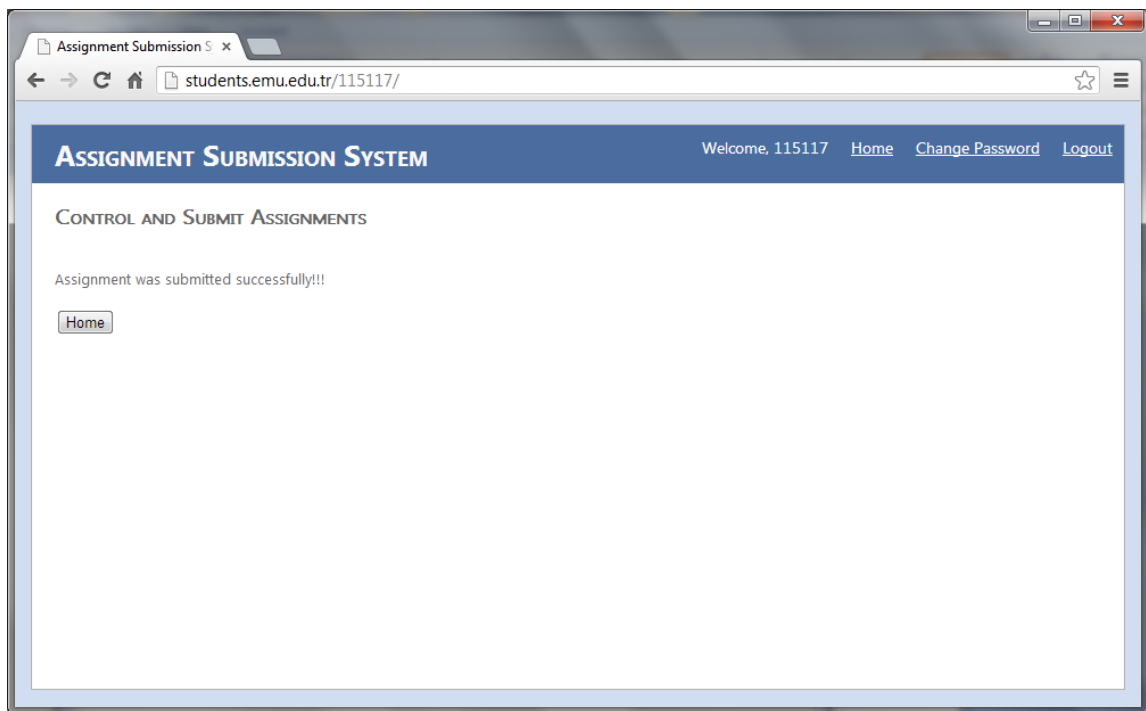


Figure 60. Assignment Submission Form – Submission Result Page

System prevents student submit solutions to the assignment more than one time. If student tries to submit second time s/he gets message shown in Figure 61:

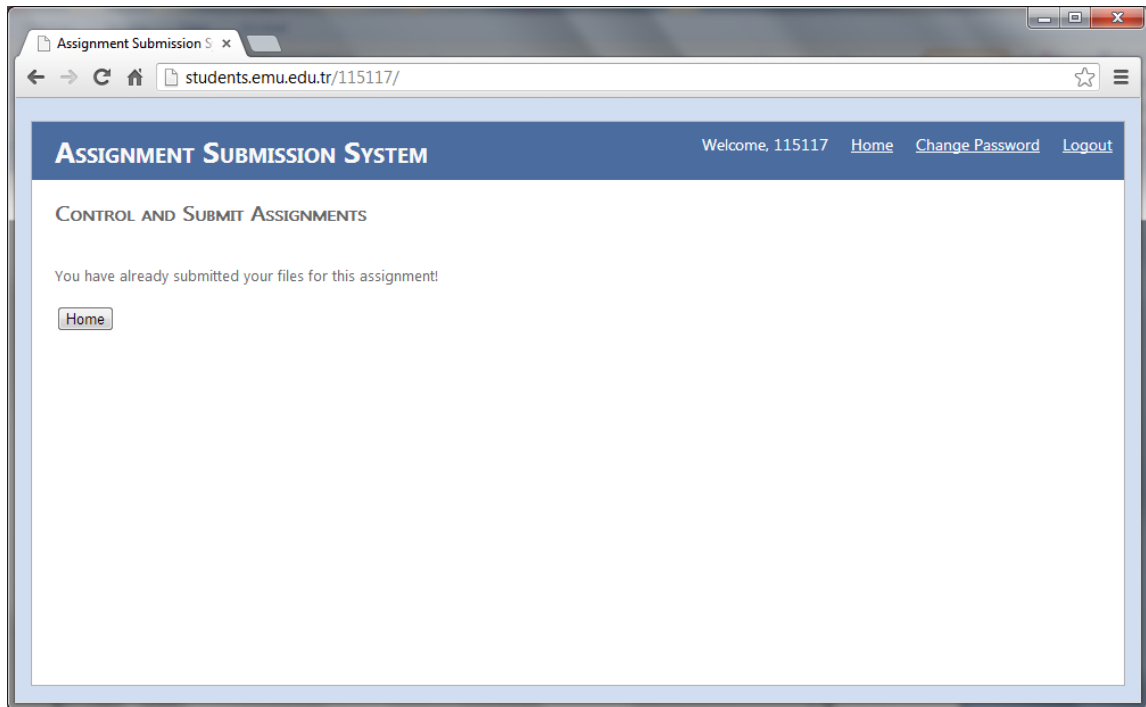


Figure 61. Assignment Submission System – System Prevent

Also student can change the password for accessing the system by clicking Change Password link. The password can be changed by defining old password and new one twice. After student finishes his/her session, s/he can click Logout link.

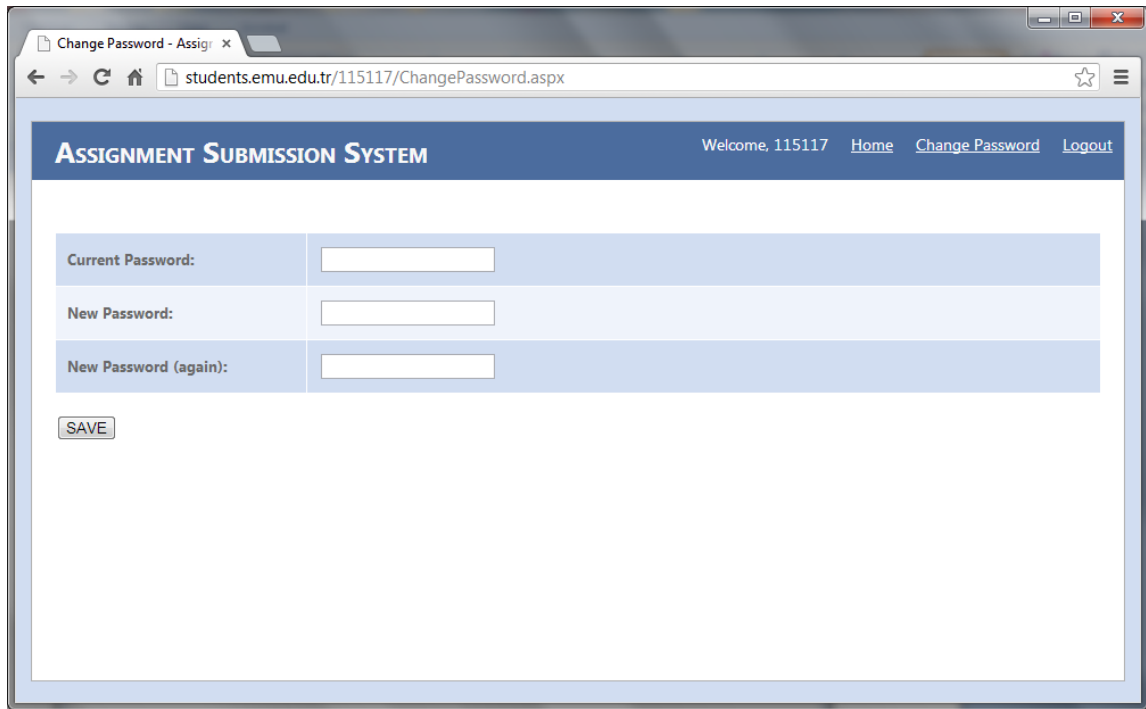


Figure 62. Assignment Submission System – Password Change Page

Appendix D: User Documentation for iOS Application

iOS application is designed for learning the grades of assignments and exam and getting notifications from instructors. It is available on App Store and can be downloaded free. When application is opened it needs authentication. Student can login with student id and password. After successfully signing in course list is shown. Student selects course, then assignment or exam for getting grades. Also s/he gets notifications from instructors. The screenshots of the application is shown in figures 63-66:

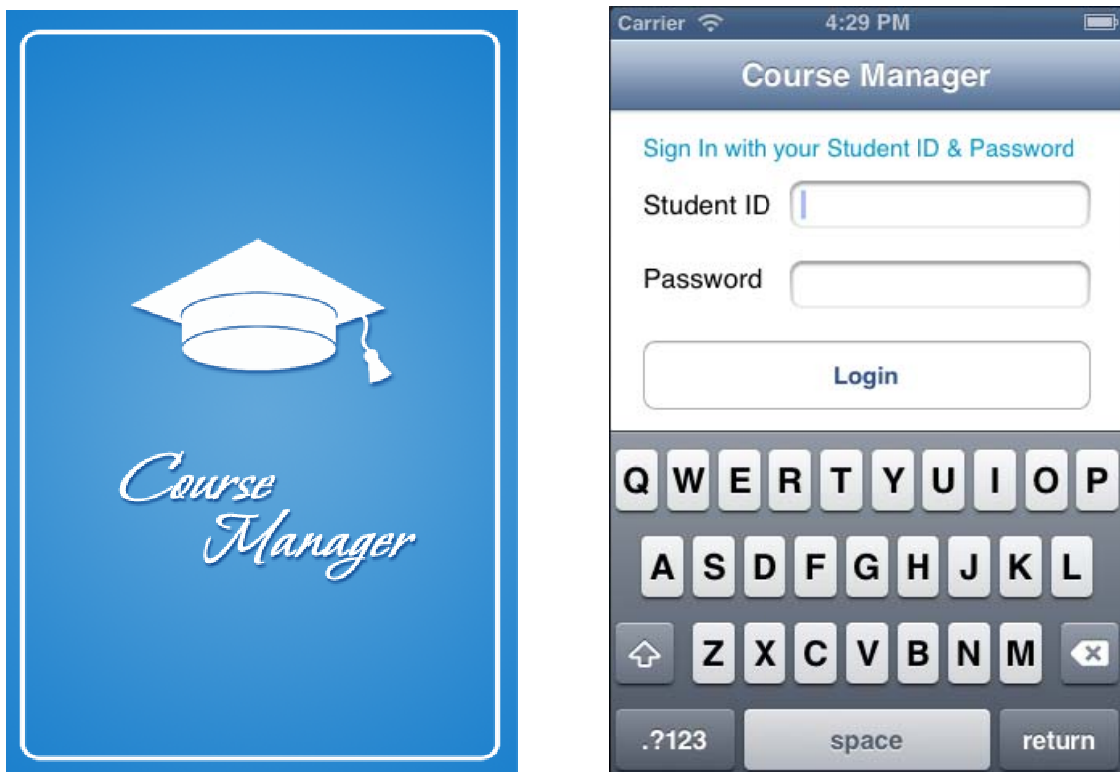


Figure 63. iOS Application Greeting Page and Authentication



Figure 64. iOS Application Course Page

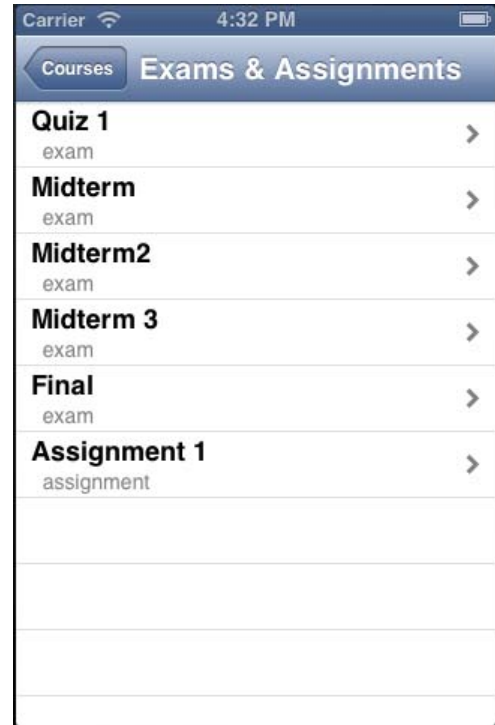


Figure 65. iOS app – Exam and Assignment Page

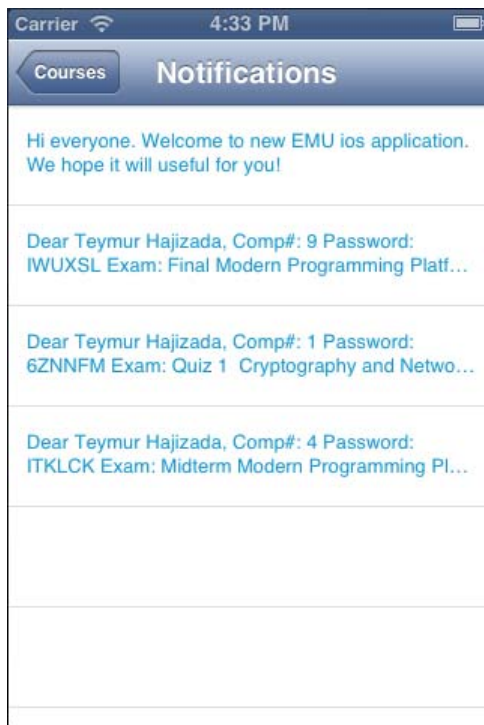


Figure 66. iOS Application Notification Page

Appendix E: Business Methods Region Source Code

```
private static PropertyInfo<int> IDProperty =
    RegisterProperty(new PropertyInfo<int>("ID"));
[System.ComponentModel.DataAnnotations(true, false)]
public int ID
{
    get { return GetProperty(IDProperty); }
    set { SetProperty(IDProperty, value); }
}

private static PropertyInfo<string> CourseCodeProperty =
    RegisterProperty(new PropertyInfo<string>("CourseCode"));
[System.ComponentModel.DataAnnotations(false, false)]
public string CourseCode
{
    get { return GetProperty(CourseCodeProperty); }
    set { SetProperty(CourseCodeProperty, value); }
}

private static PropertyInfo<string> CourseNameProperty =
    RegisterProperty(new PropertyInfo<string>("CourseName"));
[System.ComponentModel.DataAnnotations(true, false)]
public string CourseName
{
    get { return GetProperty(CourseNameProperty); }
    set { SetProperty(CourseNameProperty, value); }
}

private static PropertyInfo<string> SemesterProperty =
    RegisterProperty(new PropertyInfo<string>("Semester"));
public string Semester
{
    get { return GetProperty(SemesterProperty); }
    set { SetProperty(SemesterProperty, value); }
}
```


Appendix F: Validation Rules Region Source Code

```
#region Validation Rules

protected override void AddBusinessRules()
{
    ValidationRules.AddRule(Csla.Validation.CommonRules.StringMaxLength,
        new Csla.Validation.CommonRules.MaxLengthRuleArgs
            (CourseCodeProperty, 10));

    ValidationRules.AddRule(Csla.Validation.CommonRules.StringMaxLength,
        new Csla.Validation.CommonRules.MaxLengthRuleArgs
            (CourseNameProperty, 100));

    ValidationRules.AddRule(Csla.Validation.CommonRules.StringMaxLength,
        new Csla.Validation.CommonRules.MaxLengthRuleArgs
            (SemesterProperty, 15));
}

#endregion
```

Appendix G: Factory Methods Region Source Code

```
#region Factory Methods

    public static Course NewCourse()
    {
        return DataPortal.Create<Course>();
    }

    public static Course GetCourse(int id)
    {
        return DataPortal.Fetch<Course>(new SingleCriteria<Course, int>(id));
    }

    public static Course GetCourse(SafeDataReader dr)
    {
        return DataPortal.Fetch<Course>(new SafeDataReader(dr));
    }

    public static void DeleteCourse(int id)
    {
        DataPortal.Delete(new SingleCriteria<Course, int>(id));
    }

    private Course()
    { /* Require use of factory methods */ }

#endregion
```

Appendix H: Data Access Region Source Code

#region Data Access

```
[RunLocal]
protected override void DataPortal_Create()
{
    // TODO: Load default values
    // omit this override if you have no defaults to set
    base.DataPortal_Create();
}

private void DataPortal_Fetch(SingleCriteria<Course, int> criteria)
{
    // TODO: Load values
    using (var cn = ConnectionManager<SqlConnection>.GetManager
            (CurrentConnection.GetConnectionString(), false))
    {
        using (var cm = cn.Connection.CreateCommand())
        {
            cm.CommandType = CommandType.Text;
            cm.CommandText = "SELECT * from Course where CourseID=@id";
            cm.Parameters.AddWithValue("@id", criteria.Value);
            using (var dr = new SafeDataReader(cm.ExecuteReader()))
            {
                if (dr.Read())
                {
                    DataPortal_Fetch(dr);
                }
            }
        }
    }
}

private void DataPortal_Fetch(SafeDataReader dr)
{
    LoadProperty(IDProperty, dr.GetValue("CourseID"));
}
```

```

        LoadProperty(CourseCodeProperty, dr.GetValue("CourseCode"));
        LoadProperty(CourseNameProperty, dr.GetValue("CourseName"));
        LoadProperty(SemesterProperty, dr.GetValue("Semester"));
    }

    [Transactional(TransactionalTypes.TransactionScope)]
    protected override void DataPortal_Insert()
    {
        using (var cn = ConnectionManager<SqlConnection>.GetManager
                (CurrentConnection.GetConnectionString(), false))
        {
            using (var cm = cn.Connection.CreateCommand())
            {
                cm.CommandType = CommandType.Text;
                string SQL = " INSERT INTO Course ";
                SQL += " (";
                SQL += "CourseCode ";
                SQL += ",CourseName ";
                SQL += ",Semester ";
                SQL += " ) VALUES (";
                SQL += "@coursecode ";
                SQL += ",@coursename ";
                SQL += ",@semester ";
                SQL += ") ";
                cm.CommandText = SQL;
                cm.Parameters.AddWithValue("@coursecode",
                    ReadProperty<string>(CourseCodeProperty));
                cm.Parameters.AddWithValue("@coursename",
                    ReadProperty<string>(CourseNameProperty));
                cm.Parameters.AddWithValue("@semester",
                    ReadProperty<string>(SemesterProperty));
                cm.ExecuteNonQuery();
            }
        }
    }
}

```

```

[Transactional(TransactionalTypes.TransactionScope)]
protected override void DataPortal_Update()
{
    // TODO: update values
    using (var cn = ConnectionManager<SqlConnection>.GetManager
            (CurrentConnection.GetConnectionString(), false))
    {
        using (var cm = cn.Connection.CreateCommand())
        {
            cm.CommandType = CommandType.Text;
            string SQL = " UPDATE Course SET ";
            SQL += "CourseCode=@coursecode ";
            SQL += ",CourseName=@coursename ";
            SQL += ",Semester=@semester ";
            SQL += " WHERE CourseID=@courseid";
            cm.CommandText = SQL;
            cm.Parameters.AddWithValue("@coursecode",
                ReadProperty<string>(CourseCodeProperty));
            cm.Parameters.AddWithValue("@coursename",
                ReadProperty<string>(CourseNameProperty));
            cm.Parameters.AddWithValue("@semester",
                ReadProperty<string>(SemesterProperty));
            cm.Parameters.AddWithValue("@courseid",
                ReadProperty<int>(IDProperty));
            cm.ExecuteNonQuery();
        }
    }
}

[Transactional(TransactionalTypes.TransactionScope)]
protected override void DataPortal_DeleteSelf()
{
    DataPortal_Delete(new SingleCriteria<Course, int>(ID));
}

```

```

[Transactional(TransactionalTypes.TransactionScope)]
private void DataPortal_Delete(SingleCriteria<Course, int> criteria)
{
    // TODO: delete values
    using (var cn = ConnectionManager<SqlConnection>.GetManager
            (CurrentConnection.GetConnectionString(), false))
    {
        using (var cm = cn.Connection.CreateCommand())
        {
            cm.CommandType = CommandType.Text;
            string SQL = " DELETE FROM Course ";
            SQL += " WHERE CourseID=@courseid";
            cm.CommandText = SQL;
            cm.Parameters.AddWithValue("@courseid", criteria.Value);
            cm.ExecuteNonQuery();
        }
    }
}

#endregion

```