

The Syrian private

Faculty of Informatics & Computer Engineering

Interactive Program To Teach Children

Supervised by:

Dr.Fadi Ibrahim

Prepared by:

Omar Hassan alsayed

Khair allah barakeh

Zain AlAbeden Younes

Hasan Abd-almajeed Khalifa

All copy rights reserved for SPU university

Contents

Interactive Program To Teach Children :.....	1
contents :.....	2
Dedication :.....	3
Acknowledgment :.....	4
Key words :.....	5
Chapter 1 . Theoretical study.....	7
1.1 Introduction :.....	7
1.2 description :.....	9
1.3 existing system :.....	11
Chapter 2	Error! Bookmark not defined.
2.1 prototype model of software development:.....	Error! Bookmark not defined.
2.2. technology system requirement.....	17
2.3 system requirement specification :.....	18
2.4 system analysis & design:	Error! Bookmark not defined.
2.5 use case diagram :.....	Error! Bookmark not defined.
2.6 text description :.....	Error! Bookmark not defined.
2.7 factoring commonality among use cases :.....	37
2.8 initial domain:	40
2.9 final domain:	41
2.10 class diagram:.....	42
2.11 e-r diagram:.....	43
Chapter 3	44
3.1 implementation:	44
3.2 few snap shots:	47
Chapter 4 . conclusion :.....	55

Dedication :

Omar Hassan alsayed :

For My parents and my friends

Hasan Abd-almajeed Khalifa :

For my parents and To everyone supported me

Khair allah barakeh :

For my parents and my friends

Zain AlAbeden Younes

For my parents

Acknowledgment :

At the end of this work , we can only recall with gratitude and appreciation to all those who grabbed our hands and gave us guidance to complete the journey work

The purity of whose work and knowledge spaces wide in our hearts and minds

Greeting to all of our college flags :

Dr. Fade Ibrahim

Dr. Ahmad Alnajar

Dr. Ali Skaf

Dr. Raghad Alnijem

Dr. Ghassan Alnemr

Dr. Ali Awada

Dr. Sam Hakeem

Dr. Basem Koseba

Dr. Samer Jaafar

Dr. Wissam Alkhateb

Dr. Hassan ahmad

And all those who helped us from relatives and friends

Keywords

- Online
- Examination
- Internet
- Browser
- Database
- Testing
- Administrator
- Student
- User
- Question
- Answer
- Test
- Exam
- Youtube
- Silverlight
- WPF

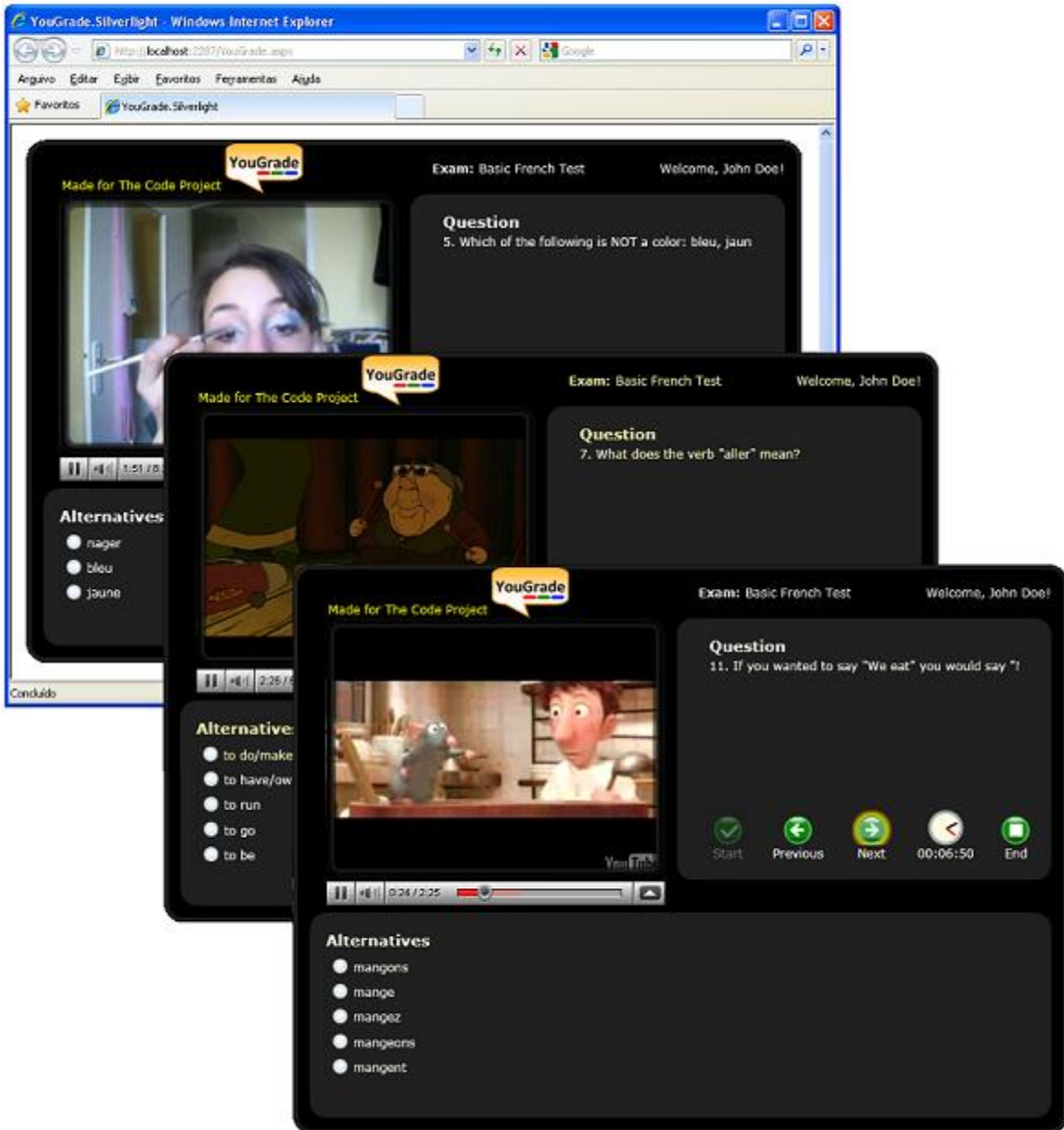


figure 1 interactive program to teach children

Chapter 1

1.1 INTRODUCTION

The purpose of On-Line Examination System is to take online examination in an efficient & effective way & no time wasting for checking the paper.

The main objective of interactive program to teach children is to efficiently evaluate the candidates' examination thoroughly through a fully administrative system that not only saves lot of time but also provides fast results in case of any consideration.

For students they attempt papers according to their convenient time and there is no need of using extra thing like paper, pen etc. The main authority is with the administrator who manages the entire examination system. Because it is an Online System that helps the candidates to attempt the papers according to the selected subjects within time limit that is already mentained by the server throughout. This project can be used in educational institutions as well as in corporate world because the main role of the software to take a partial decision in case of online test. Hence it can be used anywhere any time as it is a web based application (user Location doesn't matter).

There is really no restriction that the examiner has to be present when the candidate takes the test. Therefore, this is a web based software to enhance the online feature of the Candidates as well as the Administrator.

1.2 DESCRIPTION

SCOPE:

Scope of this Online Project is very broad in terms of other manually taking online exams by the other organizations.

Some scopes of our Projects are:-

- **This can be used in educational institutions as well as in corporate world.**
- **Can be used anywhere any time as it is a web based application (user Location doesn't matter).**
- **No restriction that examiner has to be present when the candidate takes the test.**
- **Individual Secret & unique ID & a default password are given to the Candidates for Exam appearance.**
- **No redundancy can be done.**

FEATURES:

- ❖ **Secure**
- ❖ **Time prevention**
- ❖ **Easy to use**
- ❖ **Reliable and accurate**
- ❖ **No need of examiner**
- ❖ **Consistency**
- ❖ **No debugging repeatation**

1.4 EXISTING SYSTEM :

The entire process of assigning test and evaluating their scores after the online test, was done manually till date. Processing the test paper i.e. checking and distributing respective scores used to take time when the software was not installed.

DISADVANTAGES OF CURRENT SYSTEM

- ❖ It is very difficult to analyze the exam manually.
- ❖ To take exam of more candidates more invigilators are required but no need of invigilator in case of on line exam.
- ❖ Results are not precise as calculation and evaluations are done manually.
- ❖ The chances of paper leakage are more in current system as compared to proposed system.
- ❖ Result processing takes more time as it is done manually.

FEATURES OF THE PROPOSED SYSTEM

The online test created for taking online test has following features :

- ❖ In comparison to the present system the proposed system will be less time consuming and is more efficient.
- ❖ Analysis will be very easy in proposed system as it is automated
- ❖ Result will be very precise and accurate and will be declared in very short span of time because calculation and evaluations are done by the simulator itself.
- ❖ The proposed system is very secure as no chances of leakage of question paper as it is dependent on the administrator only.
- ❖ The logs of appeared candidates and their marks are stored and can be backup for future use

Chapter 2 .

2.1 PROTOTYPE MODEL OF SOFTWARE DEVELOPMENT

The basic idea here is that instead of freezing the requirements before a design or coding can proceed, a throwaway prototype is built to understand the requirements. This prototype is developed based on the currently known requirements.

By using this prototype, the client can get an “actual feel” of the system, since the interactions with prototype can enable the client to better understand the requirements of the desired system.

Prototyping is an attractive idea for complicated and large systems for which there is no manual process or existing system to help determining the requirements.

The prototype is usually not complete systems and many of the details are not built in the prototype. The goal is to provide a system with overall functionality.

Advantages of Prototype model:

- Users are actively involved in the development
 - Since in this methodology a working model of the system is provided, the users get a better understanding of the system being developed.
 - Errors can be detected much earlier.
 - Quicker user feedback is available leading to better solutions.
 - Missing functionality can be identified easily
 - Confusing or difficult functions can be identified
- Requirements validation, Quick implementation of, incomplete, but functional, application.

Disadvantages of Prototype model:

- Leads to implementing and then repairing way of building systems.
- Practically, this methodology may increase the complexity of the system as scope of the system may expand beyond original plans.
- Incomplete application may cause application not to be used as the full system was designed
Incomplete or inadequate problem analysis.

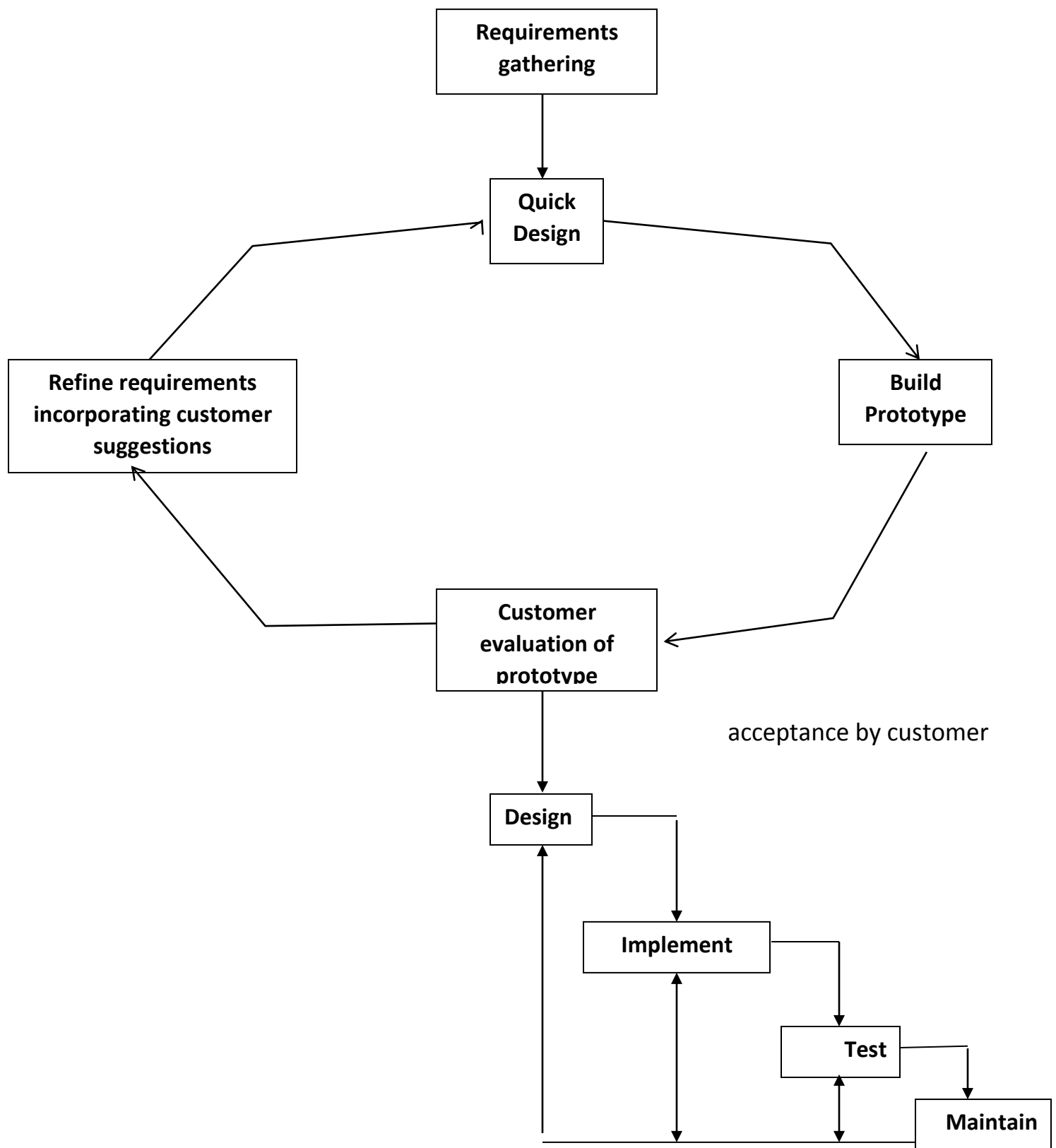


figure 2 prototype

Following is the stepwise approach to design a software prototype:

- **Basic Requirement Identification:**

This step involves understanding the very basics product requirements especially in terms of user interface. The more intricate details of the internal design and external aspects like performance and security can be ignored at this stage.

- **Developing the initial Prototype:**

The initial Prototype is developed in this stage, where the very basic requirements are showcased and user interfaces are provided. These features may not exactly work in the same manner internally in the actual software developed and the workarounds are used to give the same look and feel to the customer in the prototype developed.

- **Review of the Prototype:**

The prototype developed is then presented to the customer and the other important stakeholders in the project. The feedback is collected in an organized manner and used for further enhancements in the product under development.

- **Revise and enhance the Prototype:**

The feedback and the review comments are discussed during this stage and some negotiations happen with the customer based on factors like , time and budget constraints and technical feasibility of actual implementation. The changes accepted are again incorporated in the new Prototype developed and the cycle repeats until customer expectations are met.

2.2 TECHNOLOGY SYSTEM REQUIREMENT

TECHNOLOGIES USED

- Silverlight
- C#
- Asp.net
- XML
- WPF

Server:

IIS

Database:

Sql Express

Querying Language:

SQL

2.3 SYSTEM REQUIREMENT SPECIFICATION

FEASIBILITY STUDY :

1- ECONOMIC FEASIBILITY :

Economic analysis is most frequently used for evaluation of the effectiveness of the system. More commonly known as Cost/Benefit analysis, the procedure is to determine the benefit and saving that are expected from a system and compare them with costs, decisions are made to design and implement the system.

This part of feasibility study gives the top management the economic justification for the new system. This is an important input to the management, because very often the top management does not like to get confounded by the various technicalities that bound to be associated with a project of this kind. A simple economic analysis that gives the actual comparison of costs & benefits is much more meaningful in such cases. In the system, the organization is most satisfied by economic feasibility. Because, if the organization implements this system, it need not require any additional hardware resources as well as it saves a lot of time.

2- TECHNICAL FEASIBILITY:

Technical feasibility focuses on the existing manual system of the Online Management process and to what extent it can support the system. According to feasibility analysis procedure the technical feasibility of the system is analyzed and the technical requirements such as software facilities, procedure, inputs etc. are identified. It is also one of the important phases of the system development activities. The system offers greater levels of user friendliness combined with greater processing speed. Thus, the cost of maintenance can be reduced. Since, processing speed is very high and the work is reduced in the maintenance point of view management convinces that the project is operationally feasible.

3- BEHAVIOURAL FEASIBILITY:

People are inherently resistant to change and computer has been known to facilitate any kind of changes. An estimate should be made of how strong the user is likely to move towards the development of computerized system. These are various levels of users such as Administrator, Candidate etc. in order to ensure proper authentication and authorization and security of sensitive data of the organization.

2.4 SYSTEM ANALYSIS & DESIGN :

OBJECT ORIENTED DESIGN USING UML

UML- (Unified Modeling Language)

According to its name it's provide a set of notation (Rectangular, line, ellipse etc) to create a model of a system. The model is very useful in documenting, designing and analysis results

UML is a document for object oriented software development approach

UML diagrams.

UML can be constructed with 9 different types for a system to capture its different view, like a building is modeled along different view. The UML diagram captures following views of a system.[6]

- ❖ **Structural view**
- ❖ **Behavioral View**
- ❖ **Implementation view**
- ❖ **Environmental view**
- ❖ **User view**

User View :

This may be called as functional view. It defines the functionalities made available by the system to the user. The user view is like a black box view of the system where the internal structure & behavior of different system component are not visible. The user view can be considered as central view and all other views are designed to confirm this view.

Structural View :

The structural view defines the kind of classes & object required to understanding the working of system and its implementation. The view shows the relationship among the classes .the structural model also called the static model as structure of a system is not changed with time.

Behavioral View :

It captures how the different object interacts with each other in order to fulfill system behavior. It captures time depends on dynamic behavior of the system.

Implementation View :

Its capture important component of system and three dependences

Environmental View :

This view capture how different component of system implemented on different hardware.

For any given system one should construct all the diagram provided by UML is it necessary, the answer is no, for a simple system the use-case model, class diagram and any one of interaction diagram sufficient.

2.5 USE CASE DIAGRAM :

The Use-Case Model:

The use Cases model consist of set of use-cases that what are the different format by which a user can use the system. It's represent only sequence of interaction between user and system. These interaction consist of one main line sequence

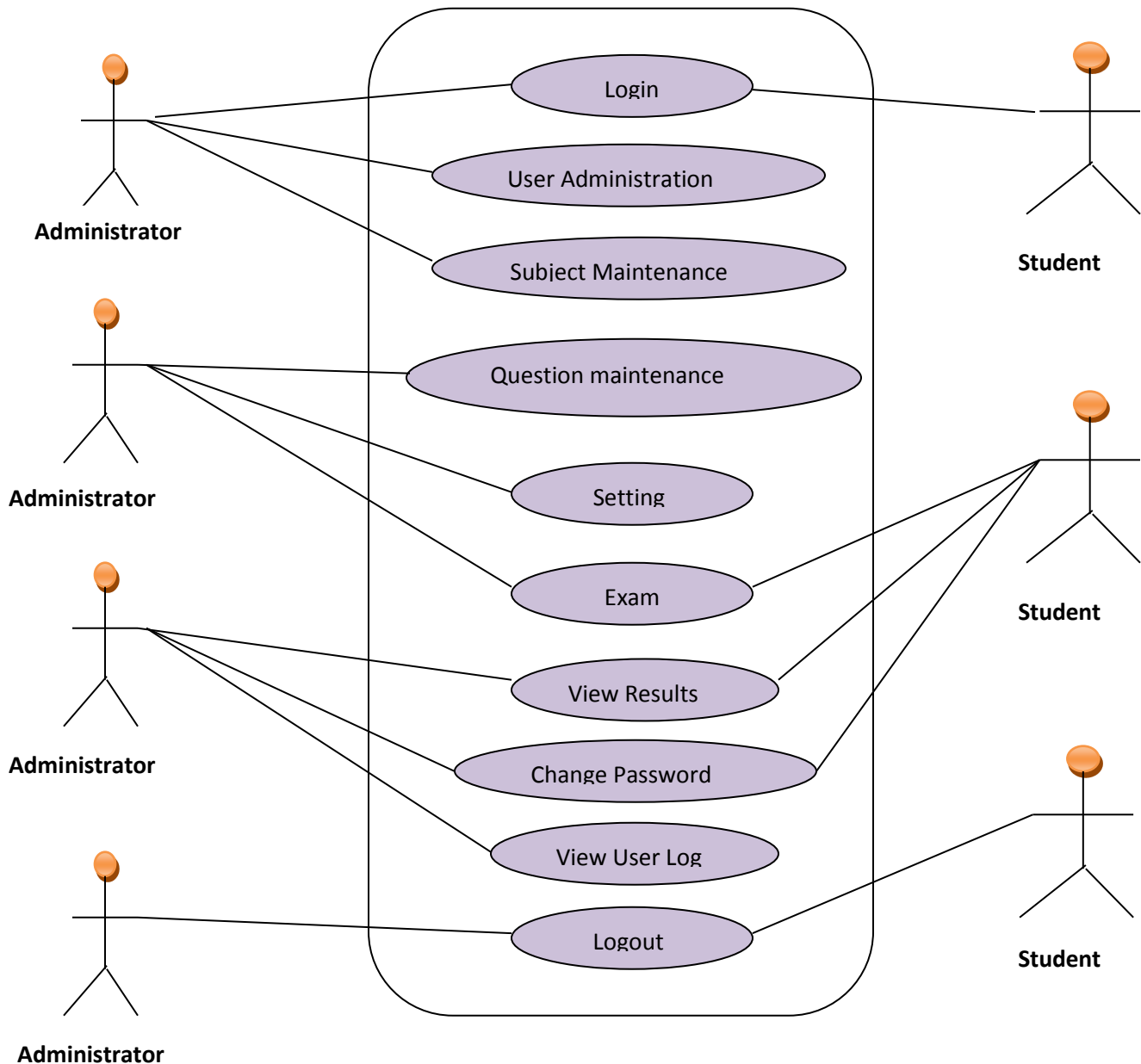


figure 3 use case model

2.6 Text Description

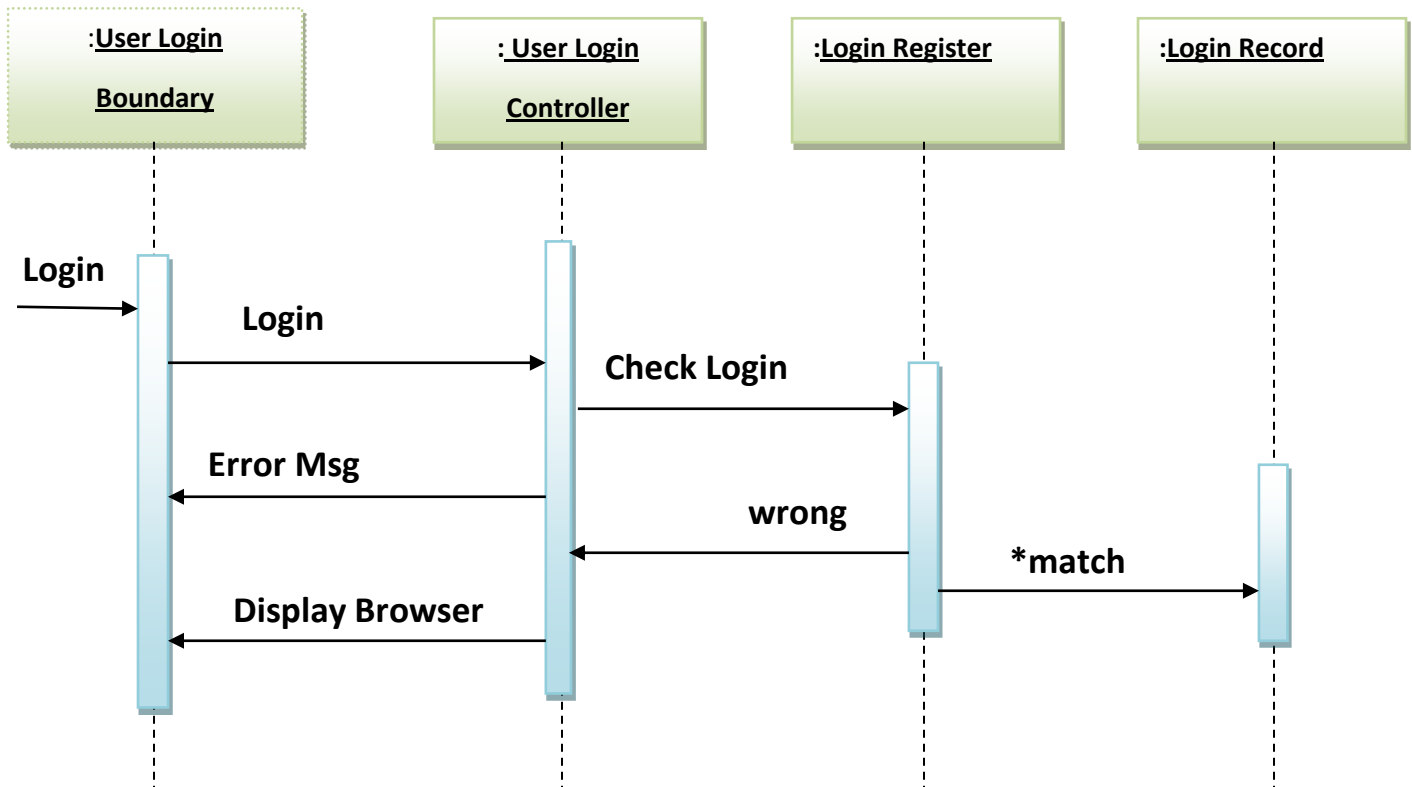


figure 4 sequence diagram for user login

U1: Login:

Using this use case the user can login himself into the system by providing use rid and password

Scenario 1: Mainline Sequence

1. User: select Login option
2. System: display Login window
3. User: enter use rid and password
4. System: display menu window according to user type

Scenario 2: At step 4 of the main line sequence

4. Display the message invalid User id and password

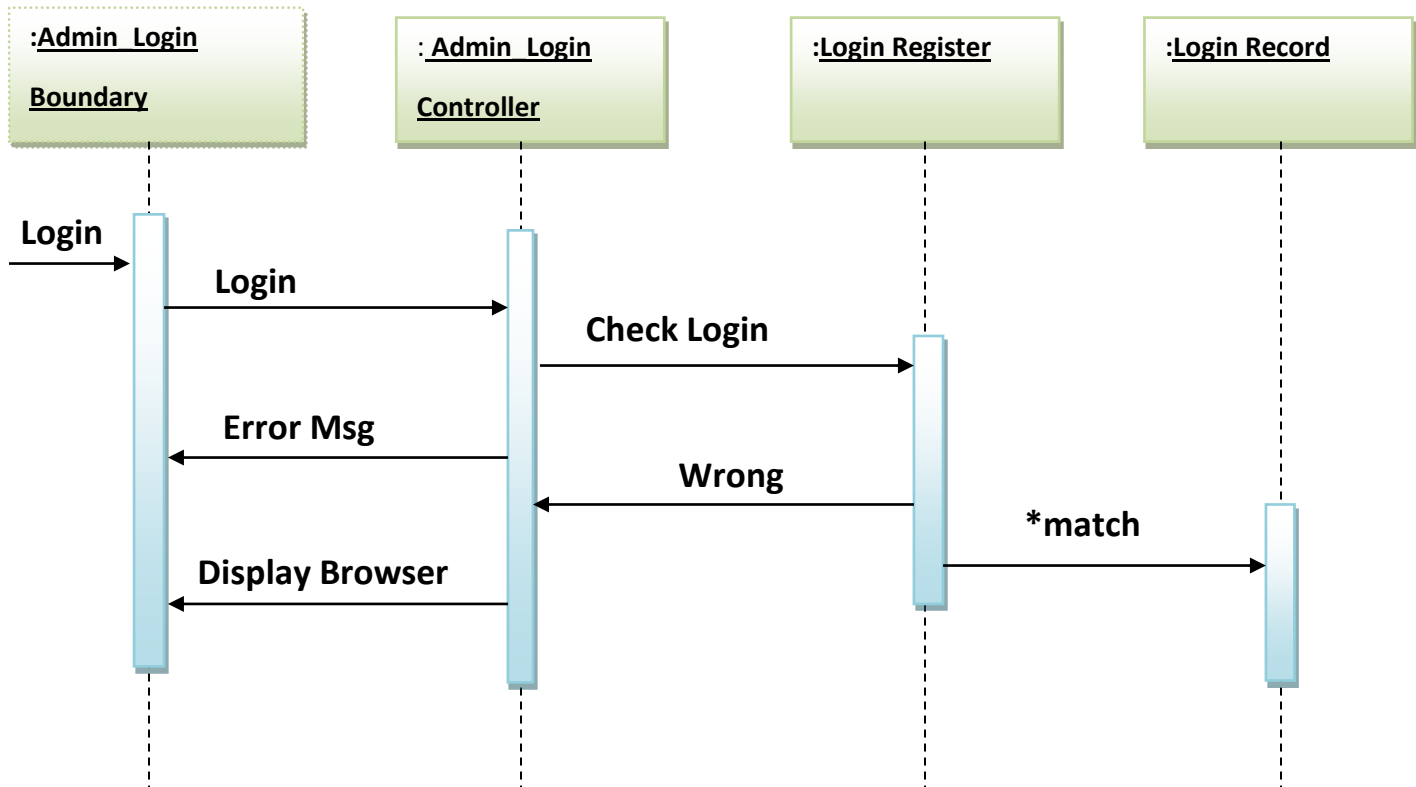


figure 5 administrator login

U2: User Administration

Scenario 1: mainline sequence

1. Administration: select the user administration option
2. System: display User administration window
3. Administration: click Add new button
4. System: Allow to enter use rid and Password
5. Admin: enter use rid and password and select user type as (admin, student)
6. System: display message that user successfully added

Scenario 2: mainline sequence at step 6

6. System display the user id already exists

Scenario 3: mainline sequence at step 3

3. Admin: click remove button

4. System: display message user successfully removes

Scenario 4: mainline sequence at step 3

3. Admin: click on edit option and modify data

4. System: display message user successfully modified

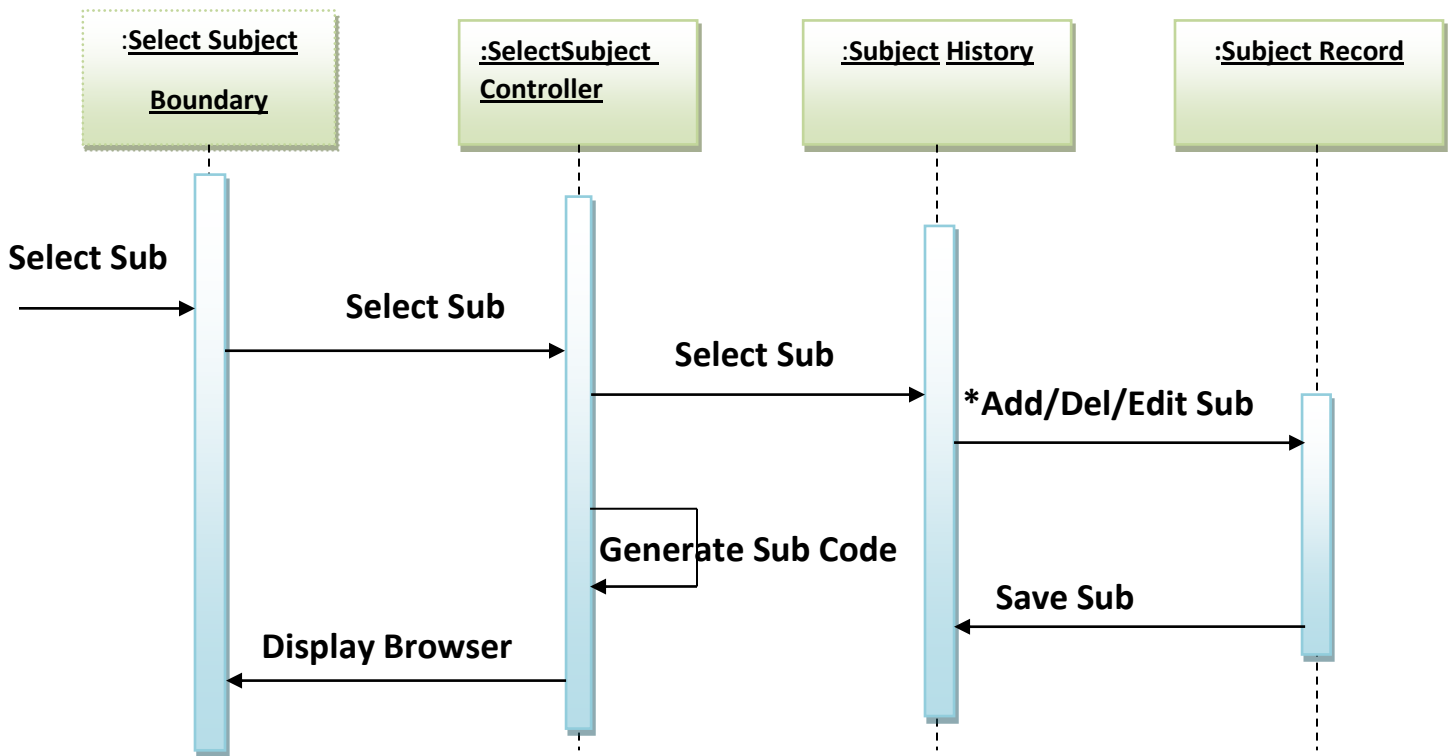


figure 6 sequence diagram for subject maintenance

U3: Subject Maintenance

Scenario 1: main line sequence

1. Admin: select subject maintenance option
2. System: display subject maintenance window
3. Admin: click on add new button
4. System: allow to add a new subject
5. Admin: enter the subject name and click on save
6. System: display message subject successfully added

Scenario2: main line sequence at step 3

3. Admin: click on remove button
4. System: remove the selected subject

Scenario 3: main line sequence at step 3

3. Admin: click on edit option
4. System: allow to edit the subject
5. Admin: perform editing to subject name and click on save
6. System: display a confirmation message

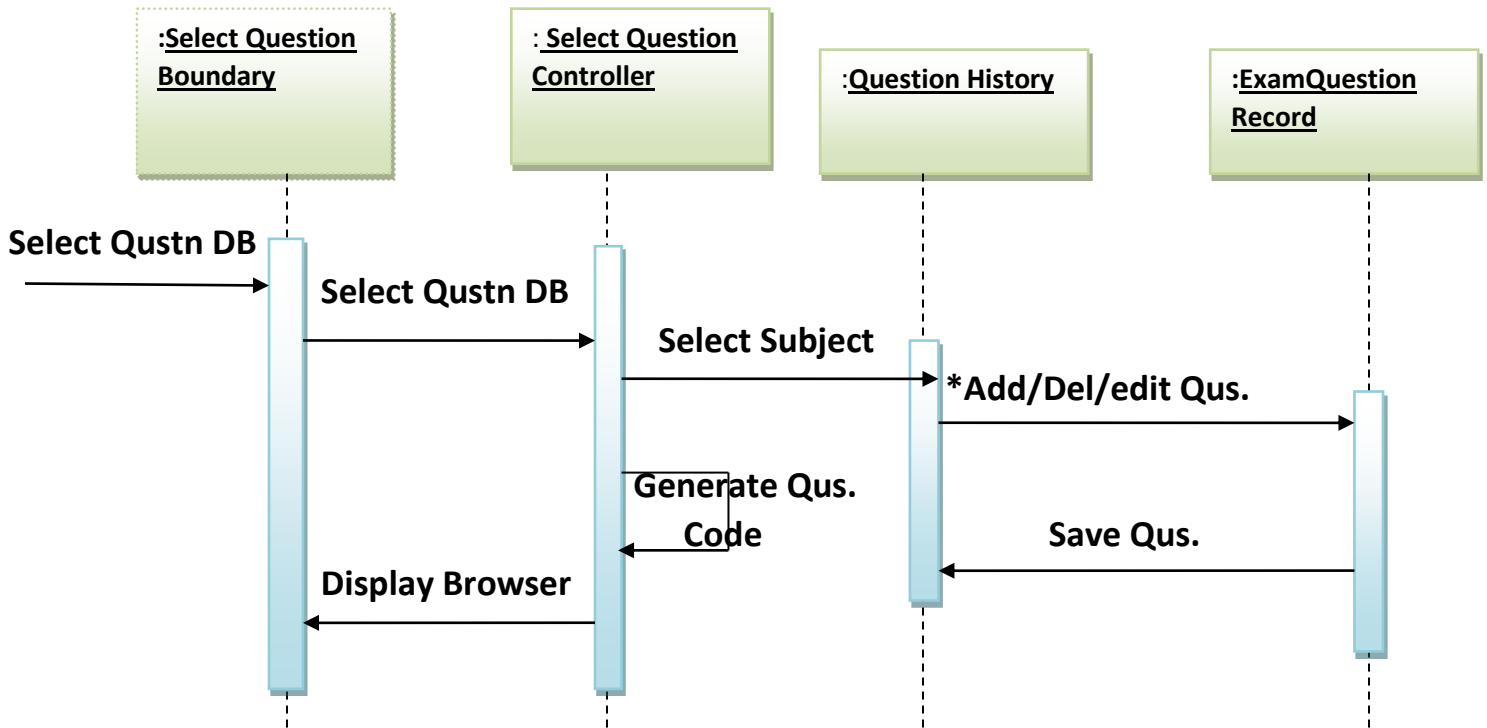


figure 7 sequence diagram for question maintenance

U4: Question Maintenance

Scenario 1: main line sequence

1. Admin: select Question maintenance option
2. System: display question maintenance window
3. Admin: select a subject for adding question for that subject
4. System: display question editing window for the selected subject
5. Admin: click on add new button
6. System: provide text boxes for writing question
7. Admin: provide necessary information and click on save
8. System: display a message that question successfully added

Scenario 2: main line sequence at step 5

5. Admin: click on remove button
4. System: remove the selected question

Scenario 3: main line sequence at step 5

5. Admin: click on edit option
6. System: allow to edit the question
7. Admin: perform editing to question details and click on save
6. System: display a confirmation message

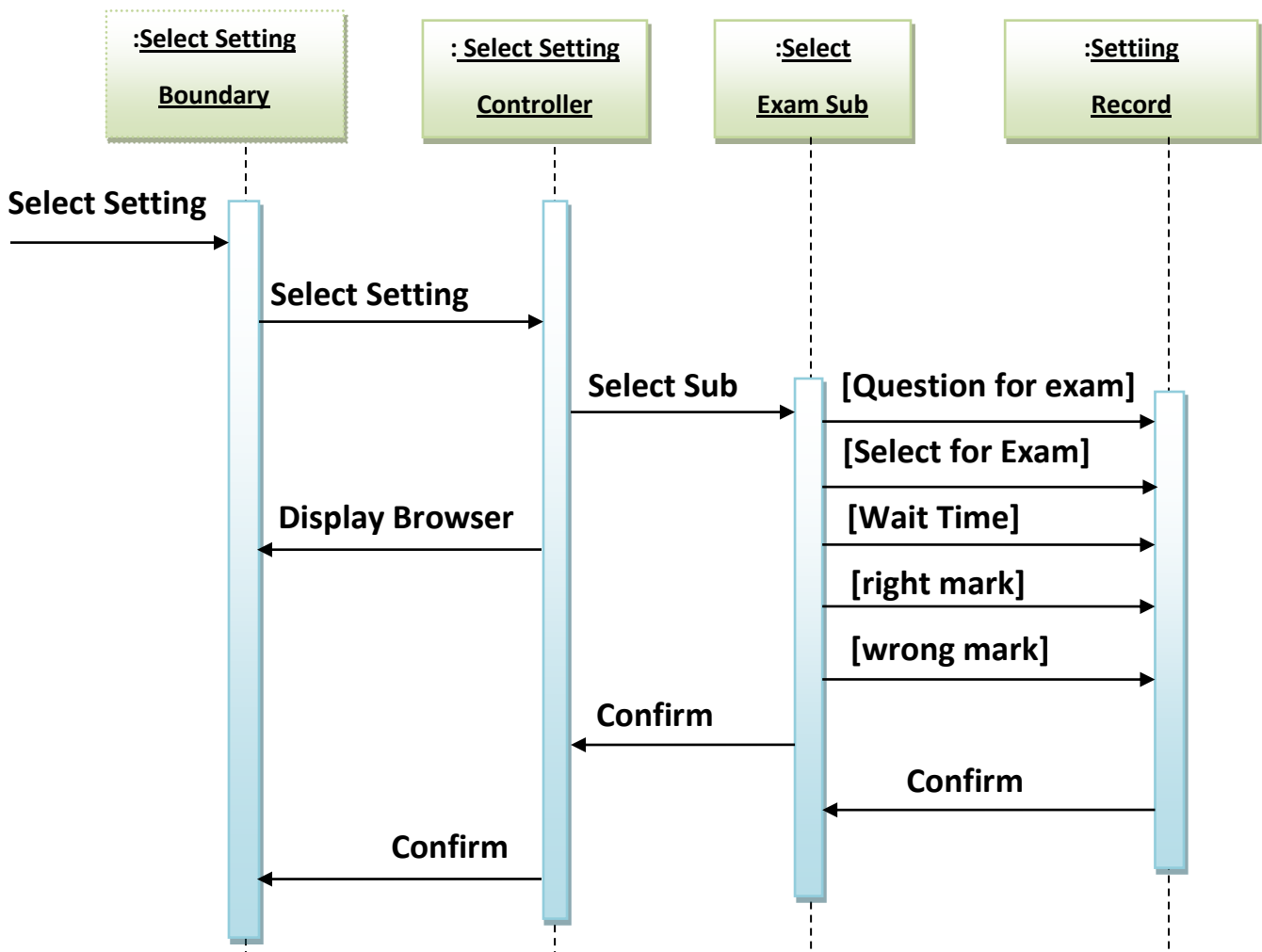


figure 8 sequence diagram for setting

U5: question

Scenario 1: main line sequence

1. Admin: select setting option
2. System: display setting window
3. Admin: select a subject for setting
4. System: display question editing window for the selected subject setting
5. Admin: perform setting to the subject and click on ok button

8. System: display a message that setting successfully modified

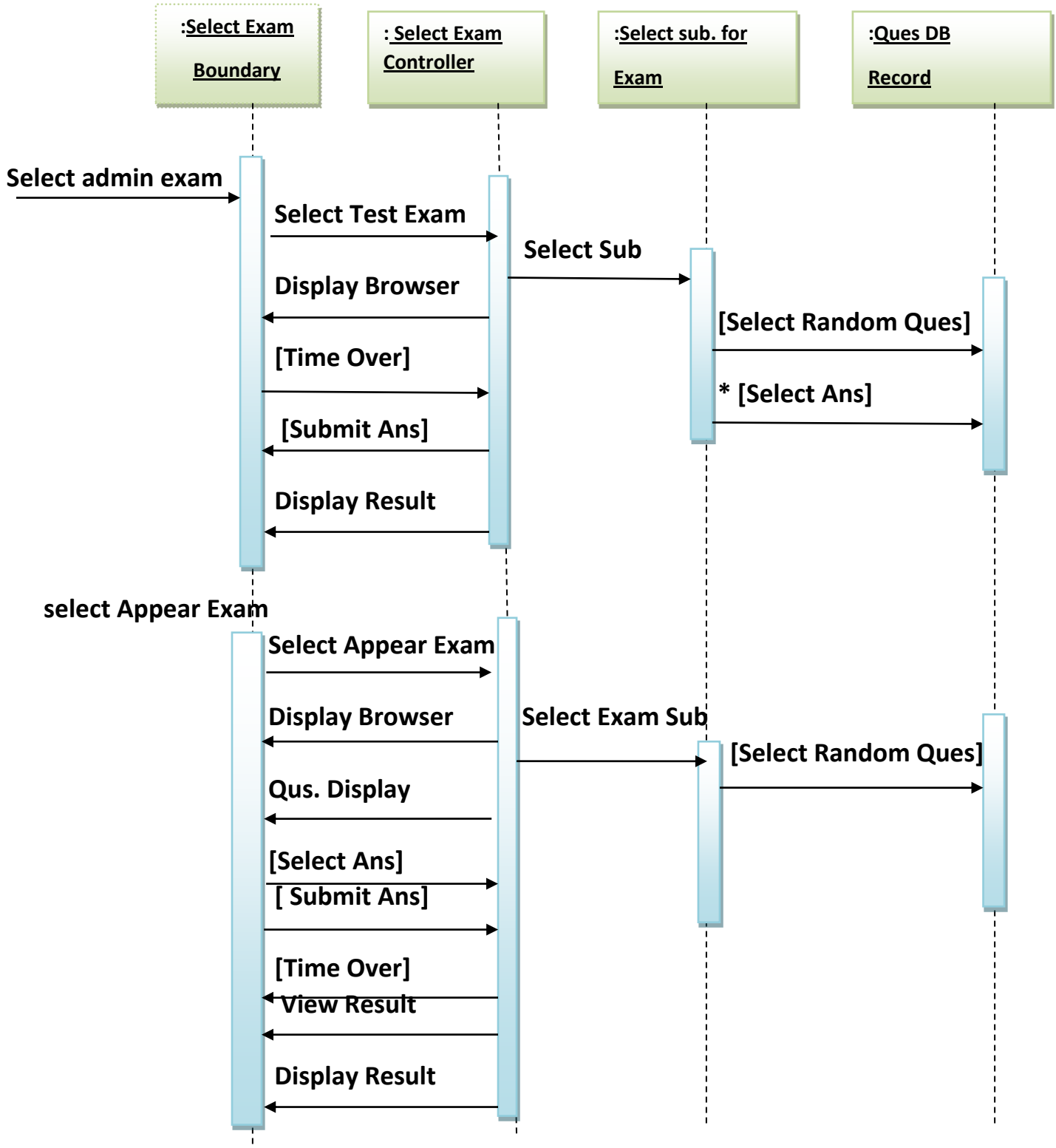


figure 9 sequence diagram for exam

U6: Exam

Scenario 1: main line sequence

1. Admin: select Test exam option
2. System: display subject selection window
3. Admin: select a subject for appearing exam for that subject
4. System: select randomly questions from database and show to user
5. Admin: select answers for all question clicks on submit answer paper
6. System: display results

Scenario2: main line sequence at step 5

5. Admin: not able to giving all answer within specify time
6. System: automatically submit answer paper and show results

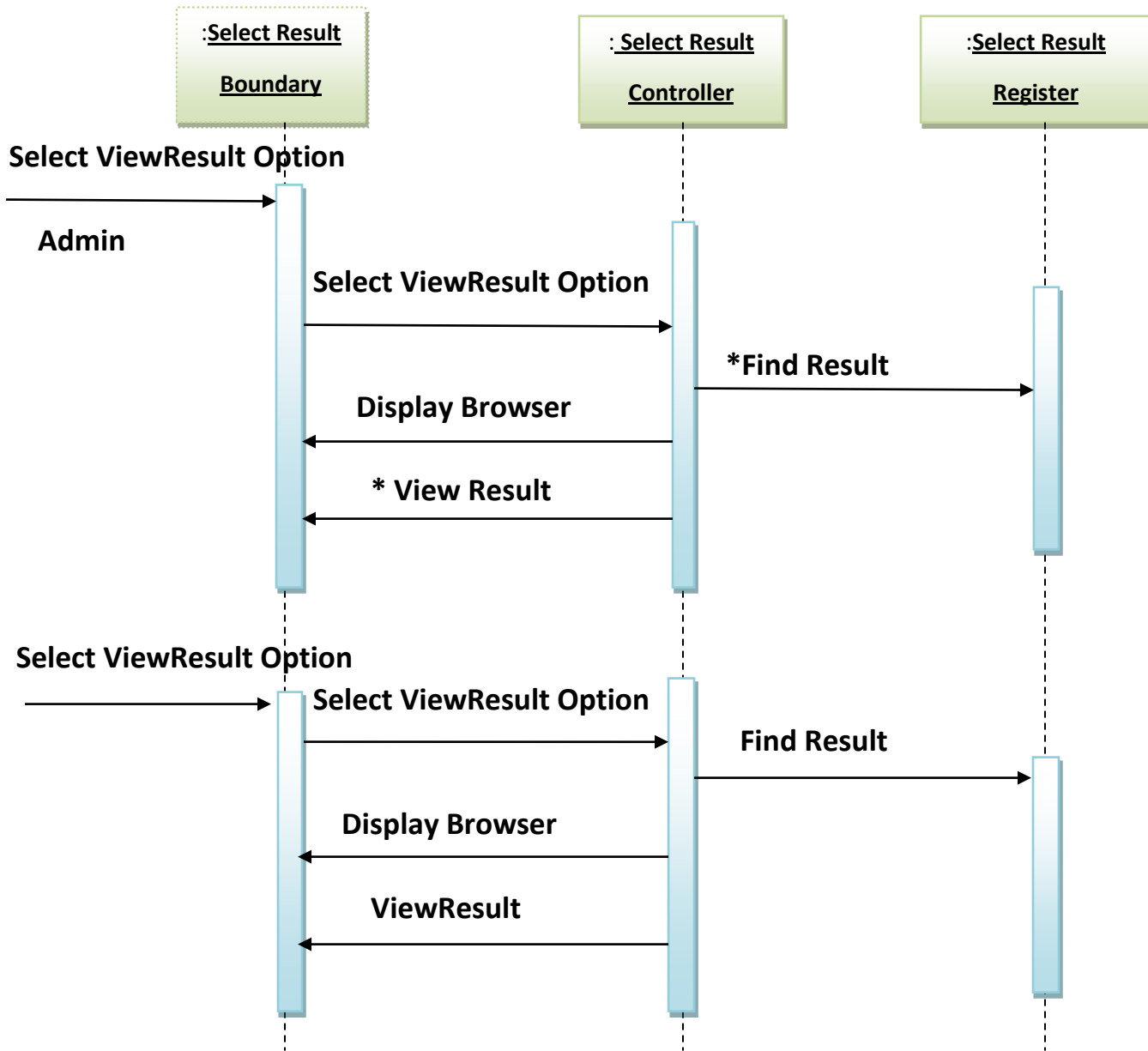


figure 10 Sequence Diagram For View Result

U7: View results

Scenario 1: main line sequence

1. Admin: select view results option
2. System: display results window for all appeared student

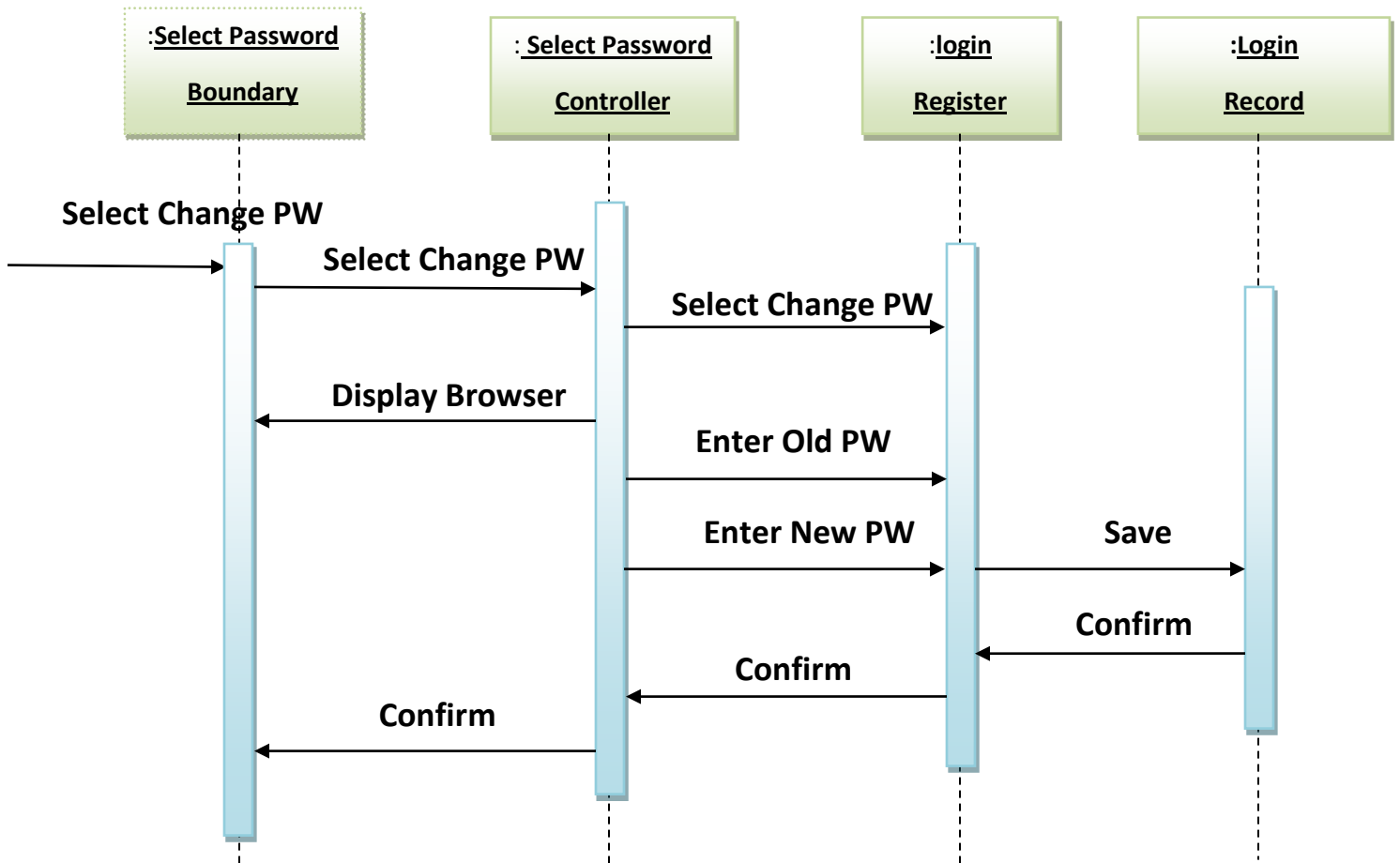


figure 11 sequence Diagram For Change Password

U8: change password

Scenario 1: main line sequence

1. Admin: select change password option
2. System: display password changing window
3. Admin: provide old password and new password and click on ok
4. System: display message password successfully changed

Scenario 2: main line sequence at step 4

4. System: display message old password not correct allow user to re enter

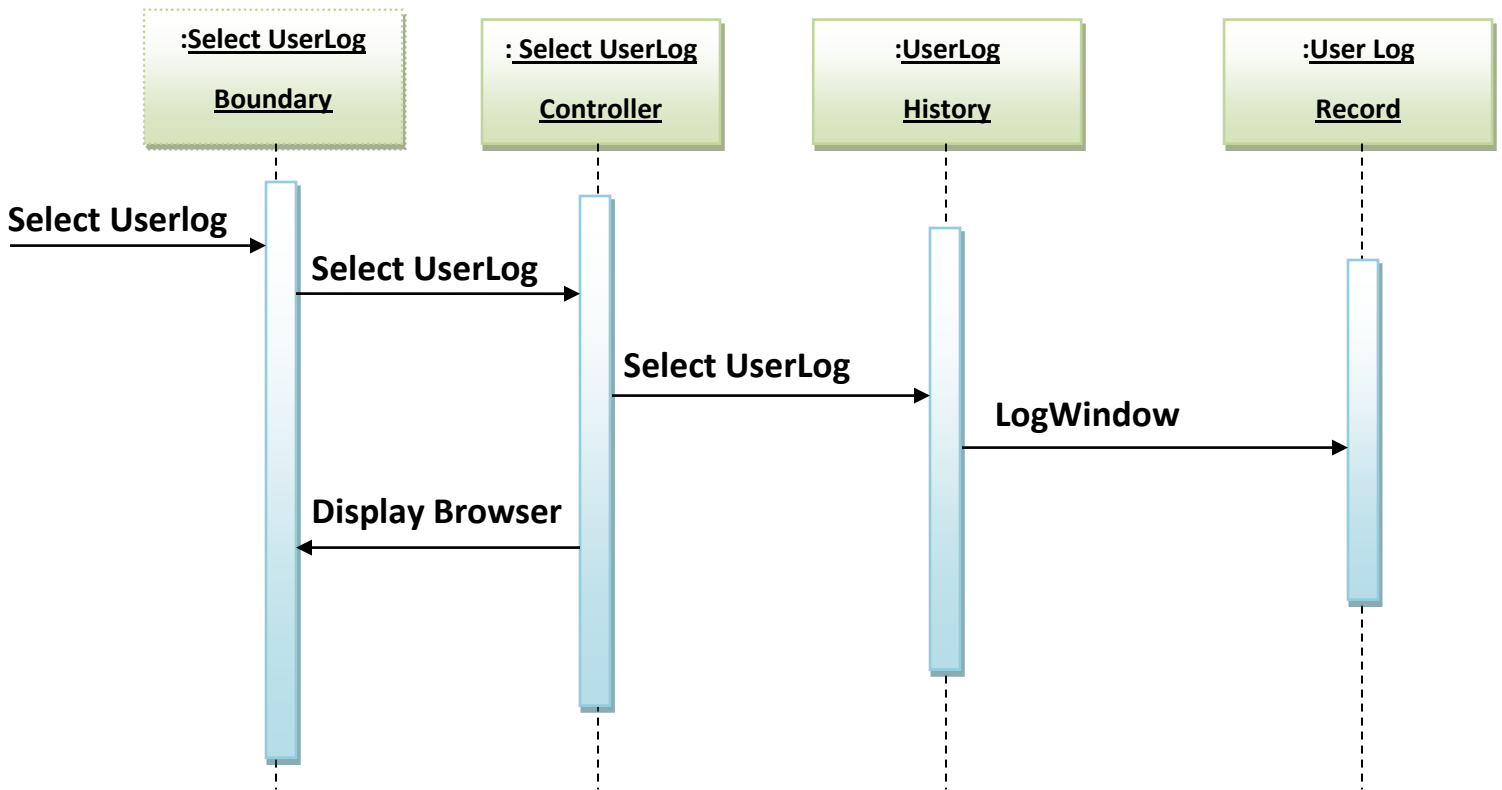


figure 12 Sequence Diagram For UserLog

U9: View user Log

Scenario 1: main line sequence

1. Admin: select view user Log option
2. System: display user Log window for all appeared student

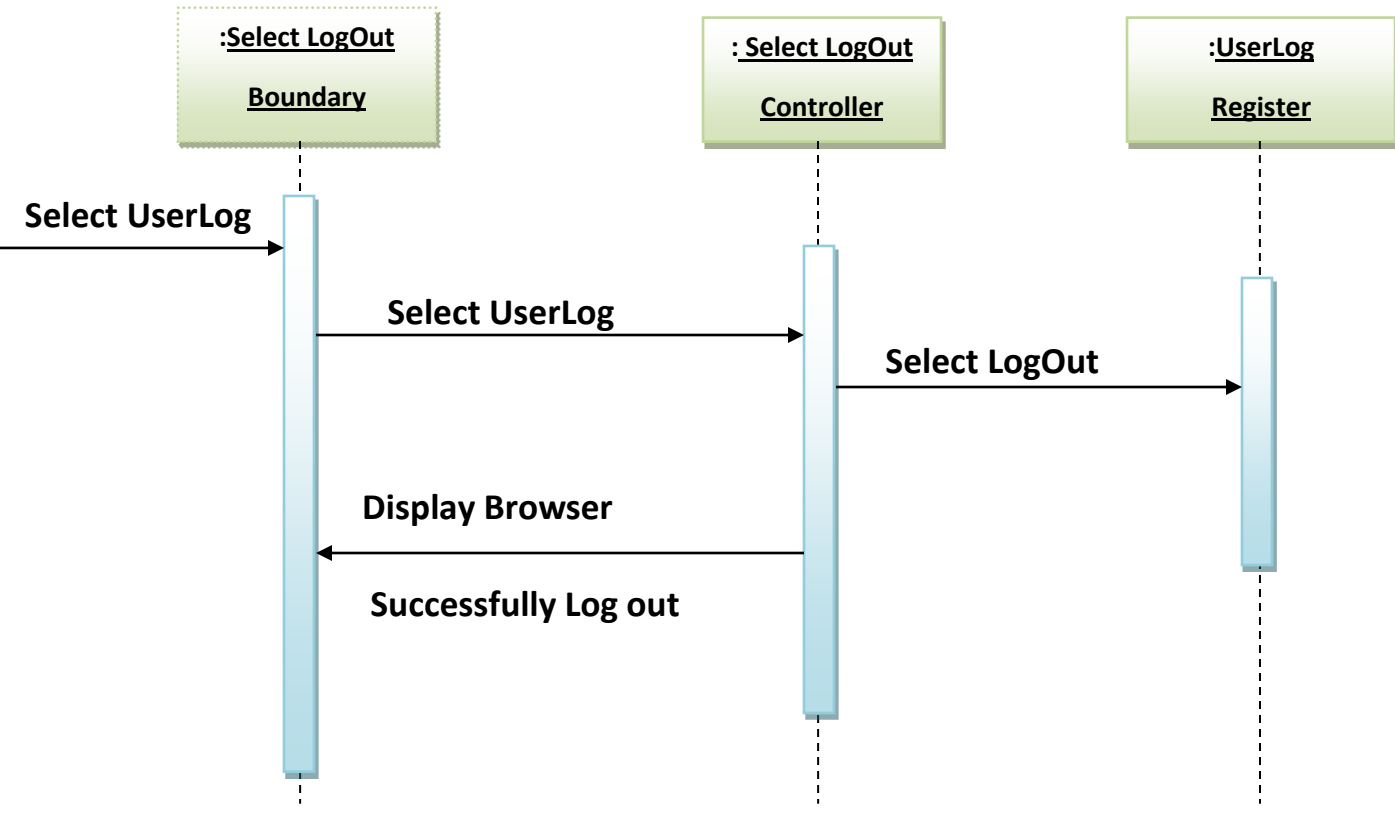


figure 13 Sequence Diagram For LogOut

U10: Logout

Scenario 1: main line sequence

1. Admin: select Logout option
2. System: display message "success fully log out"

2.7 Factoring commonality among use cases :

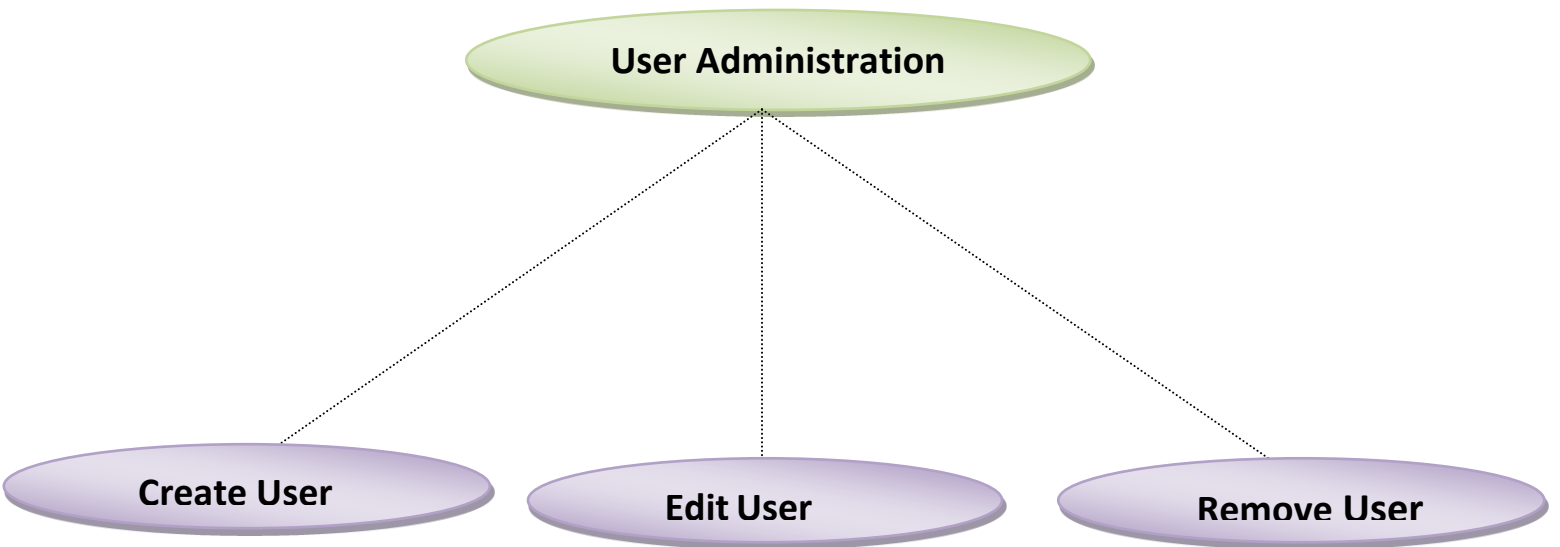


figure 14 Admin can create user , edit user or remove user

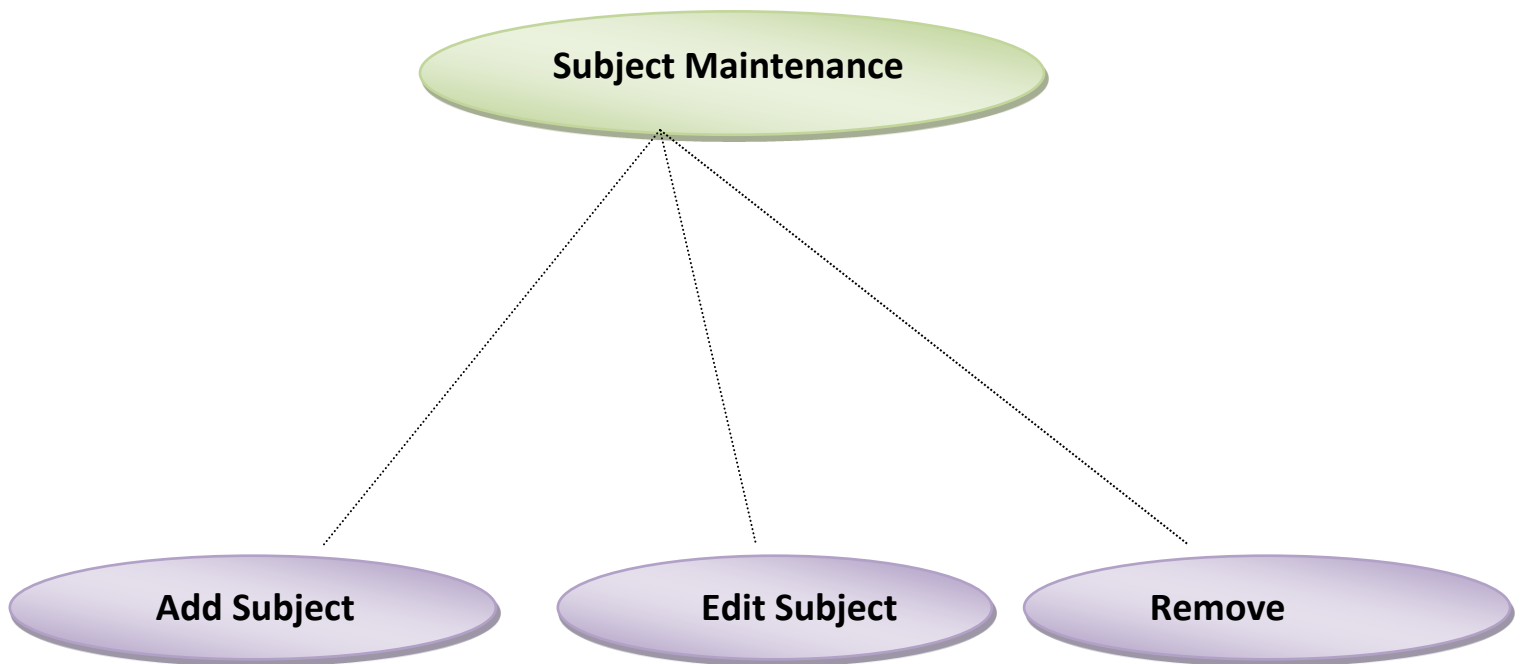


figure 15 subject maintenance can add subject, edit subject or remove subject

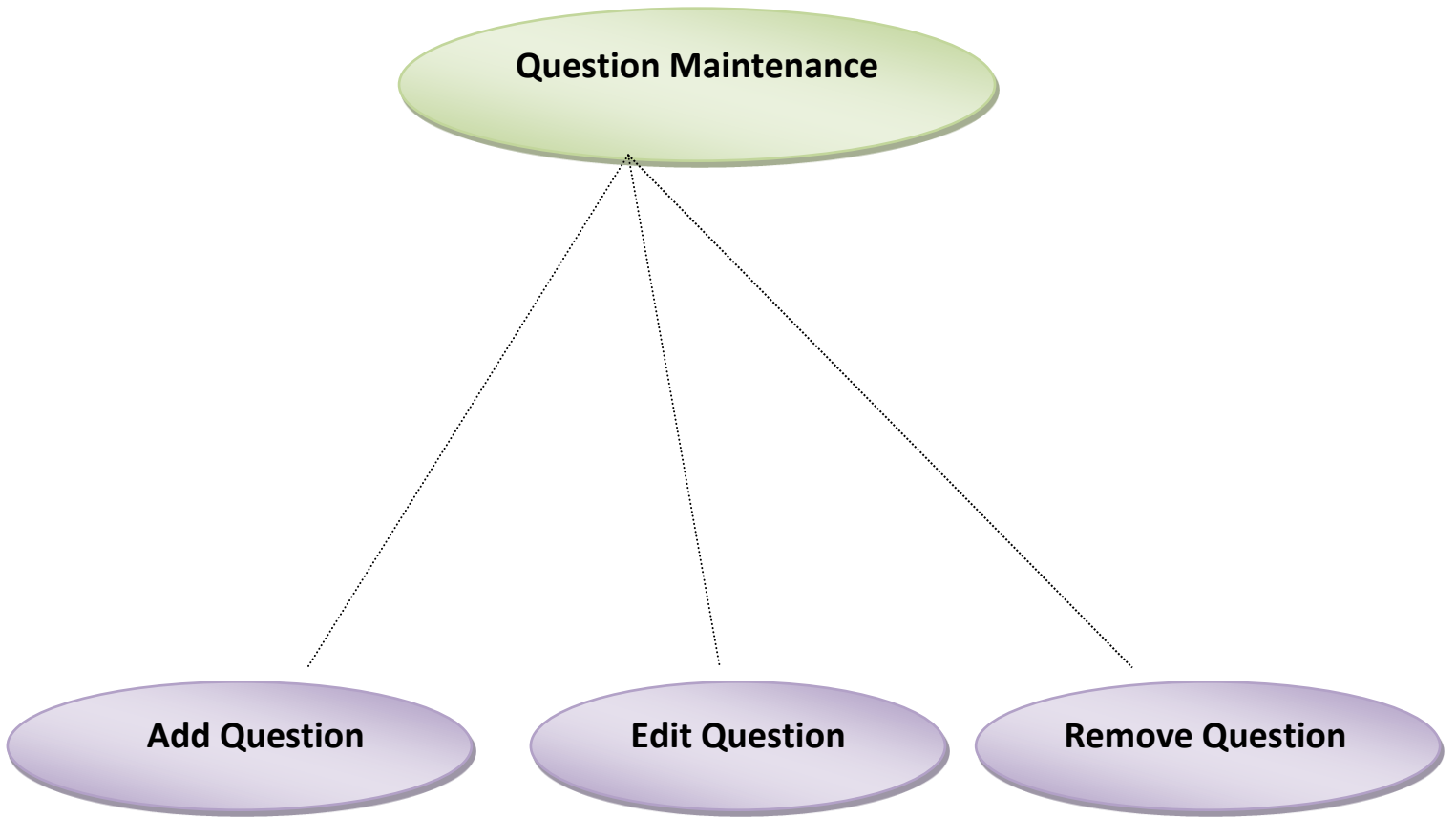


figure 16 Question maintenance can add question, edit question or remove question

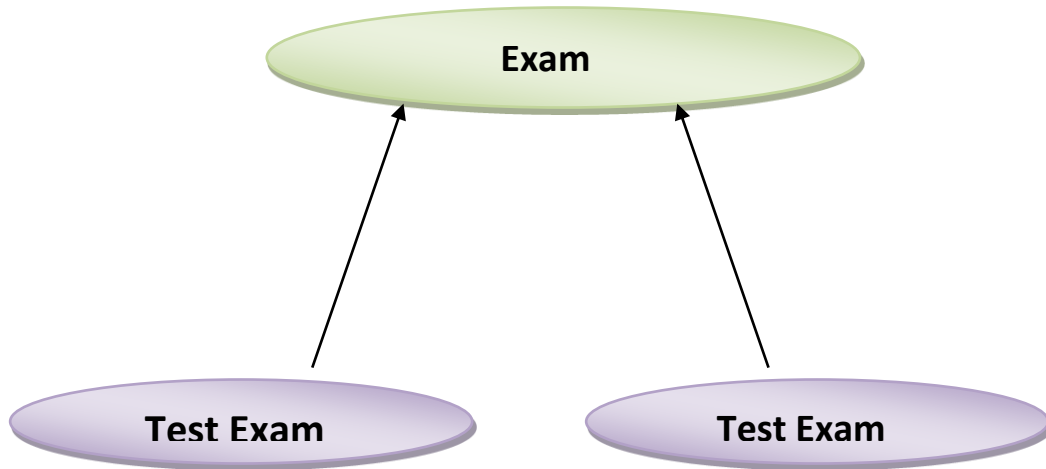


figure 17 Test Exam and Test Exam are inheritance from Exam

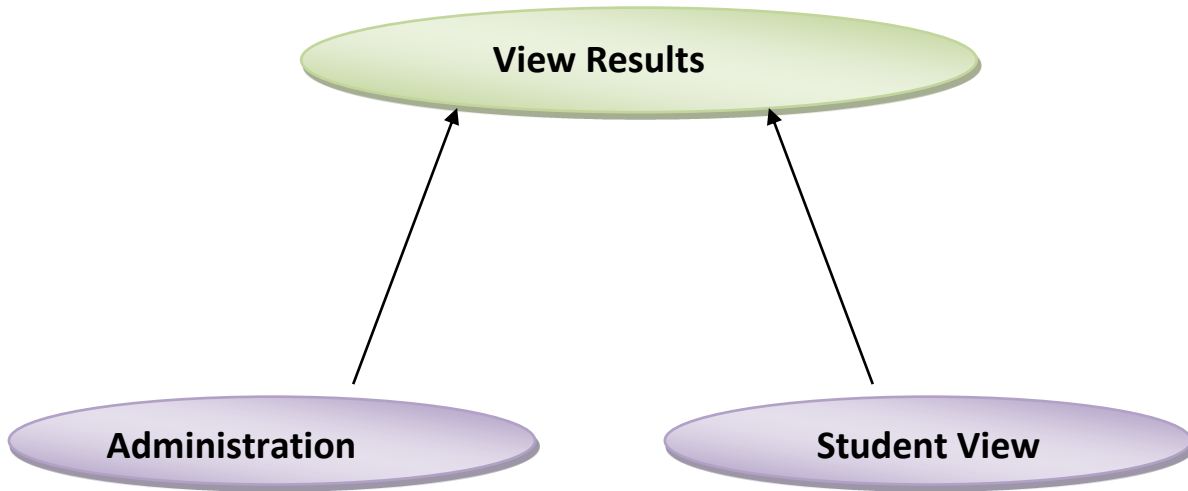


figure 18 Administration View and Student View are inheritance from View Result

2.8 INITIAL DOMAIN :

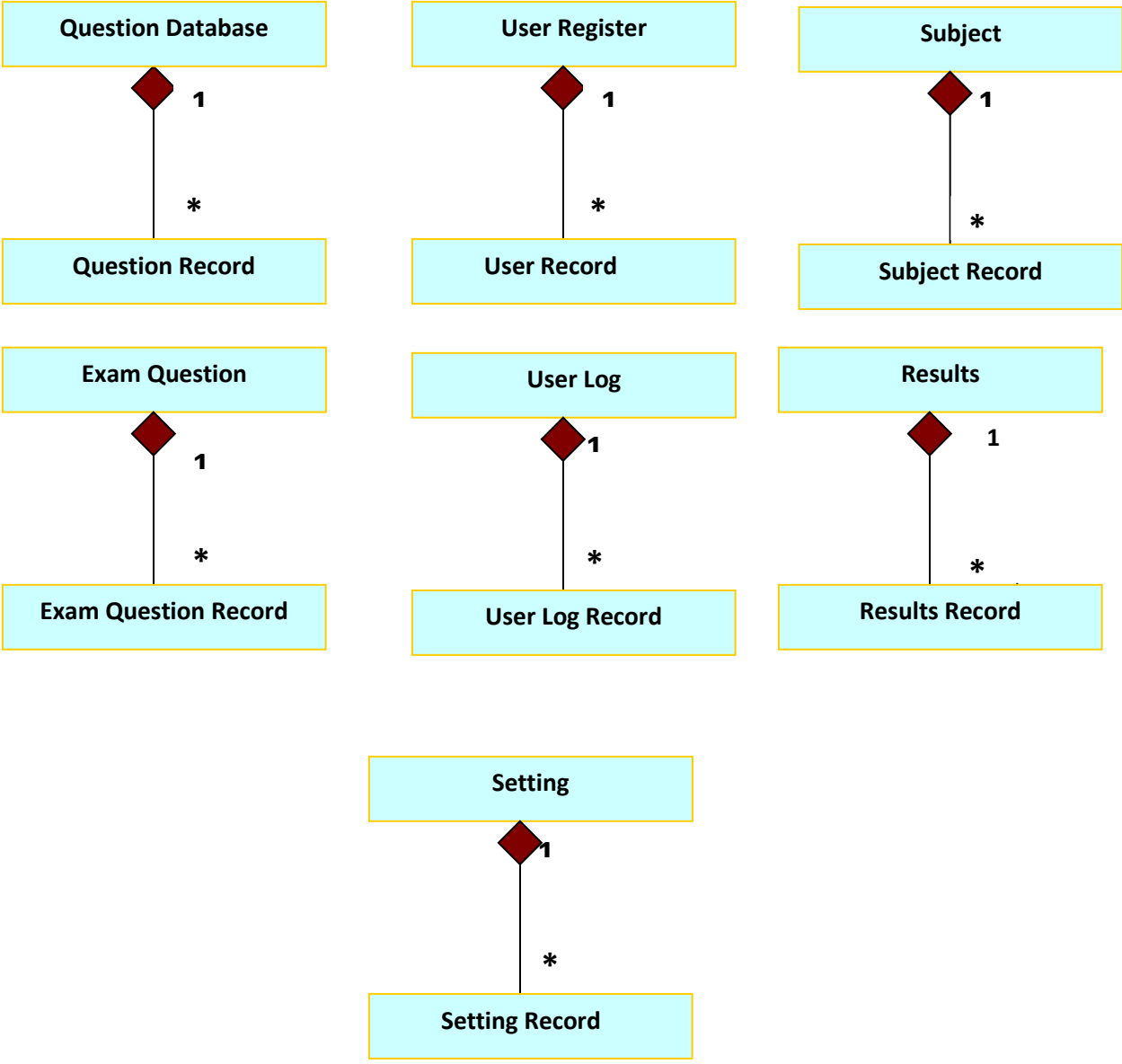


figure 19 The Initial Domain Model

2.9 FINAL DOMAIN :

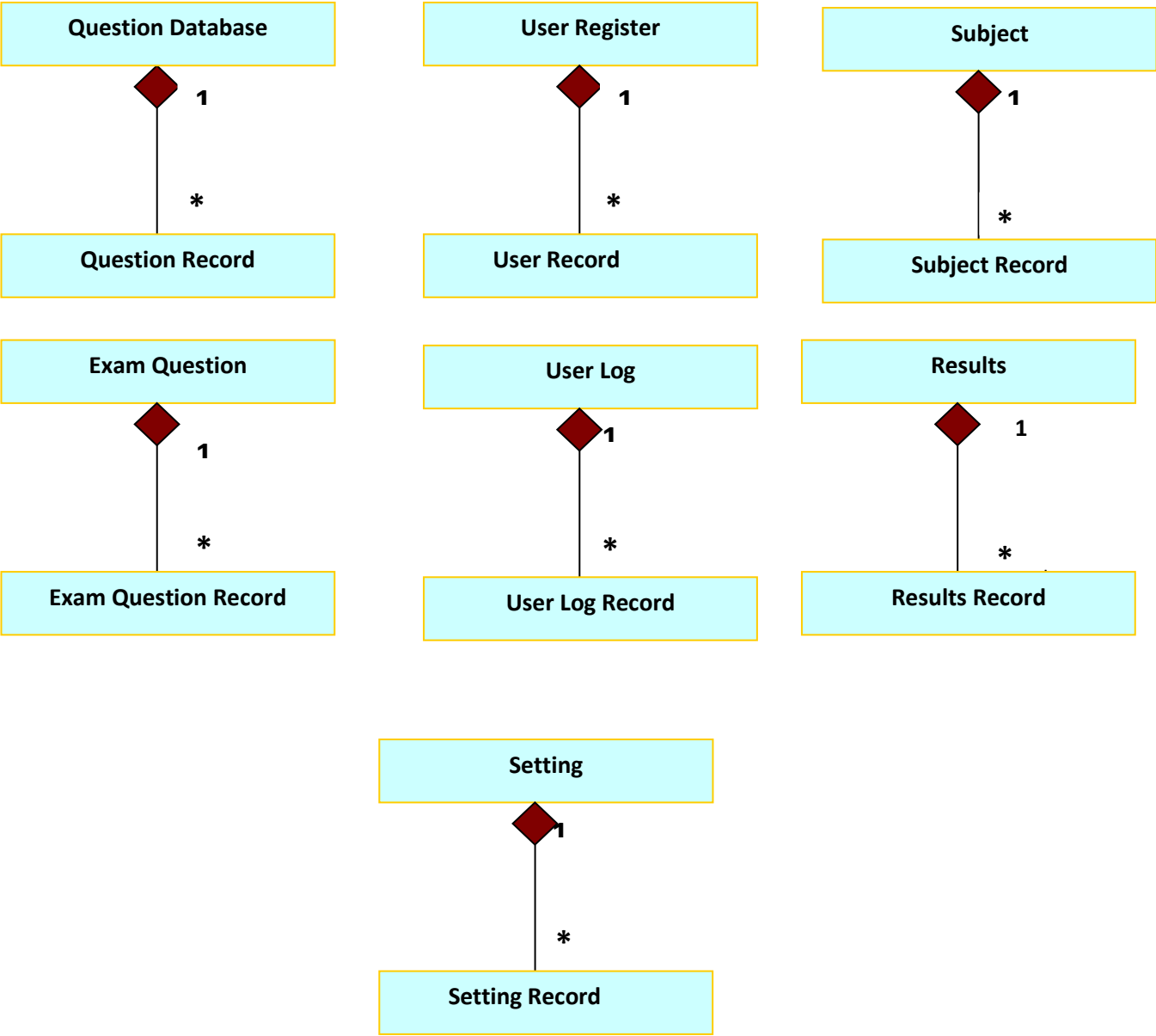


figure 20 The Final Domain model

2.10 CLASS DIAGRAM :

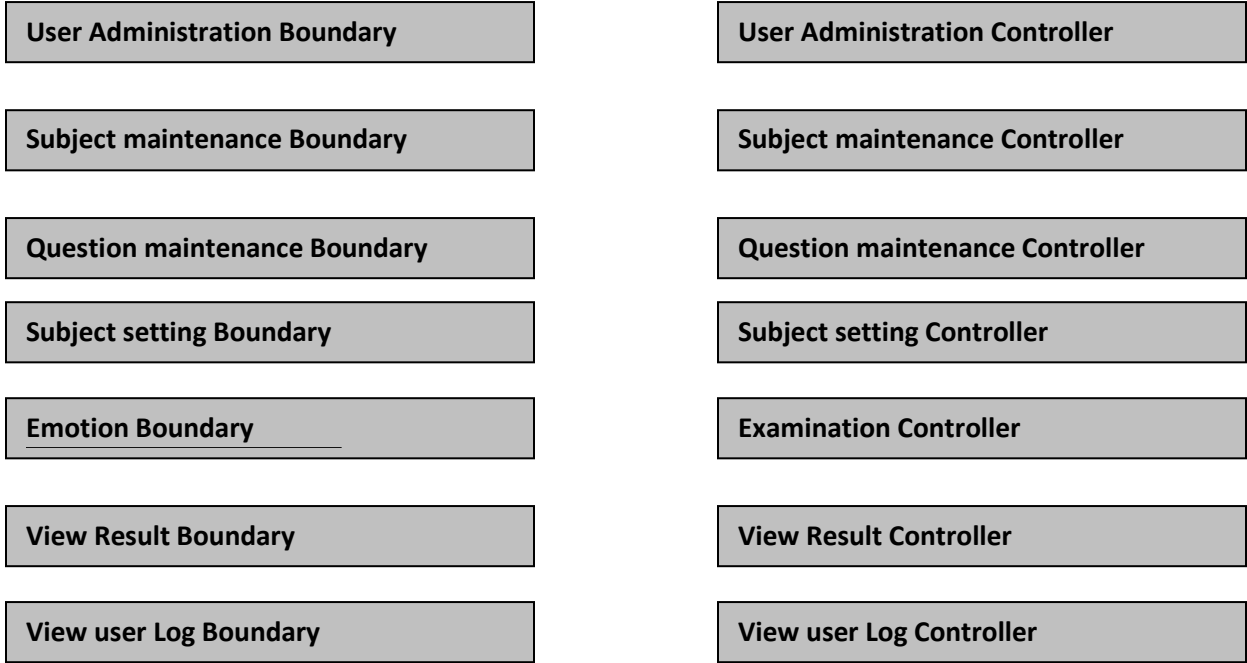
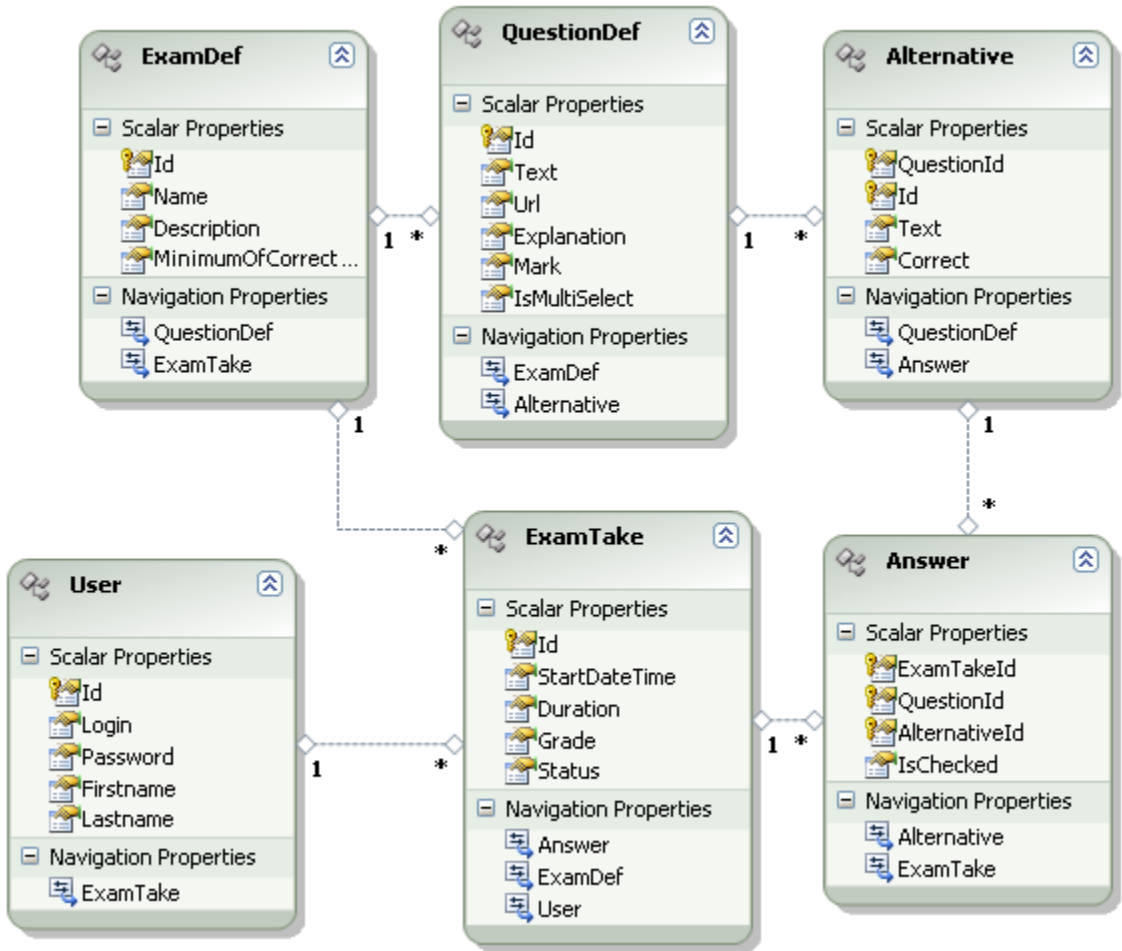


figure 21 class Diagram

2.11 E-R DIAGRAM :

<i>Entity</i>	<i>Description</i>
User	Represents the user/student that is applying for the test.
ExamDef	Represents the exam definition, containing name, definition, minimum of correct answers, and duration.
QuestionDef	Represents the exam questions (along with the YouTube video IDs).
Alternative	Represents the question alternatives.
ExamTake	Represents the data for each user/student exam appliance.
Answer	Represents the answers the user/student has chosen for the alternatives for each question.

figure 22



Chapter 3 .

3.1 implementation :

Project solution

<i>Project</i>	<i>Description</i>
YouGrade.Silverlight	This is the Silverlight project itself.
YouGrade.Silverlight.Controls	This project holds some of the custom controls I use in the project.
YouGrade.Silverlight.Web	This project is the start up project that contains the application's entry page. Also, this web application holds the Silverlight-enabled WCF service that provides the Silverlight project with data services.
Yougrade.Silverlight.Core	This Silverlight Class Library project just holds transfer objects (TOs) which are transferred between other projects.

Si
lv
er
li
g
ht
Pr

oject

No doubt this is the main focus of this article. For the interface of this application, I could choose between many technology options, such as WPF, Windows Forms, ASP.NET, ASP.NET MVC, Silverlight, and so on. I finally decided for Silverlight for some reasons:

- From the user's point of view, it's more convenient to access Web-based applications than to install and run Windows applications.
- Many users don't have administrative privileges in their company machines, and installing the application might require additional technical support.[9]
- I'm sure ASP.NET or ASP.NET MVC would satisfy the needs of this application. But I also wanted a technology that provided users with a Rich Application Interface experience, without forcing the developer to know a lot of JavaScript and CSS.[7]
- And finally, I really, really wanted a strong motivation to start developing something with Silverlight...

The MVVM Pattern:

We used the Model-View-ViewModel (MVVM) pattern. MVVM became the *lingua franca* for WPF developers. The same could be said about Silverlight developers, since the Silverlight framework is a subset of the WPF foundation.

For those of you who don't know MVVM yet, here's a little explanation: almost every developer knows what the "M" (Model) and "V" (View) mean. Model contains data and View presents the data. In most systems, the Model part maps database tables, and the View part shows those data to the user, via Windows Forms, plain HTML, WebForms, and in the case of WPF and Silverlight, this role is played by XAML files. So far, nothing new. But why do you need the ViewModel part? In fact, you are not forced to use View Models (VM) in Silverlight in any way, just like you are not forced to use the MVC pattern when you build an ASP.NET application. But there are people that think MVC is a nice pattern for ASP.NET, just like many people feel natural to use the MVVM pattern with XAML. In fact, it's so natural that MVVM was born while WPF was still under construction.

That being said, what's the difference between using or not using the ViewModel? Without the ViewModel, you would have something like the usual Windows Forms paradigm, where you put code on the *MyForm.cs* file to retrieve data from your business or data layer and then populate the fields in the form. In Silverlight, you could use the same approach and put code on your *MyWindow.xaml.cs* file to get data, and populate the XAML elements in the view, and also control the events like button clicks and listbox selection changes. The ViewModel approach, on the other hand, splits the presentation layer into View and ViewModel, so that the View doesn't retrieve data from the data source anymore. Instead, each View (that is, the XAML file) has its element properties bound to specific properties in the ViewModel class. Once these bindings are set up, the View elements and ViewModel properties stay synchronized, that is, each modification in one part automatically updates the other

As we can see from **Figure 23** below, the View doesn't access the data directly; instead, it relies on the bindings provided by its ViewModel counterpart. For example, a grid element named "gridProducts" in a View might have a binding to an ObservableCollection property named "Products" in the ViewModel. Any changes to the "Products" collection property would reflect in the gridProducts element on the View side. On the other hand, a Button element named "btnInsert" might have a binding to a Command property named "InsertCommand" in the ViewModel, so that any time the user clicks the button, the Insert() method on the ViewModel side is automatically invoked

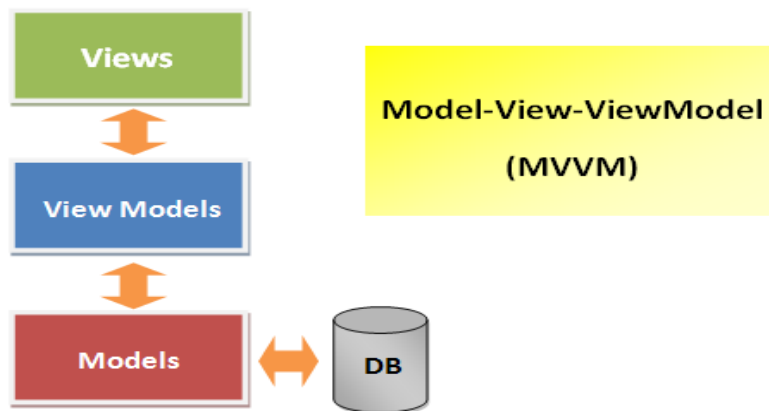


figure 23 Model-View-ViewModel (mvvm)

3.2 Few snap shots :

The Login View

As expected, the login view sits in front of everything else in our application. It simply authenticates the user and then blocks or allows access to the rest of our application.

The only user you'll have is "Omar Omar", which has the login omar and the password omar. You may also create more users, if you wish, but you should do this by inserting users directly in the User table, because currently there is no appropriate user interface for this.

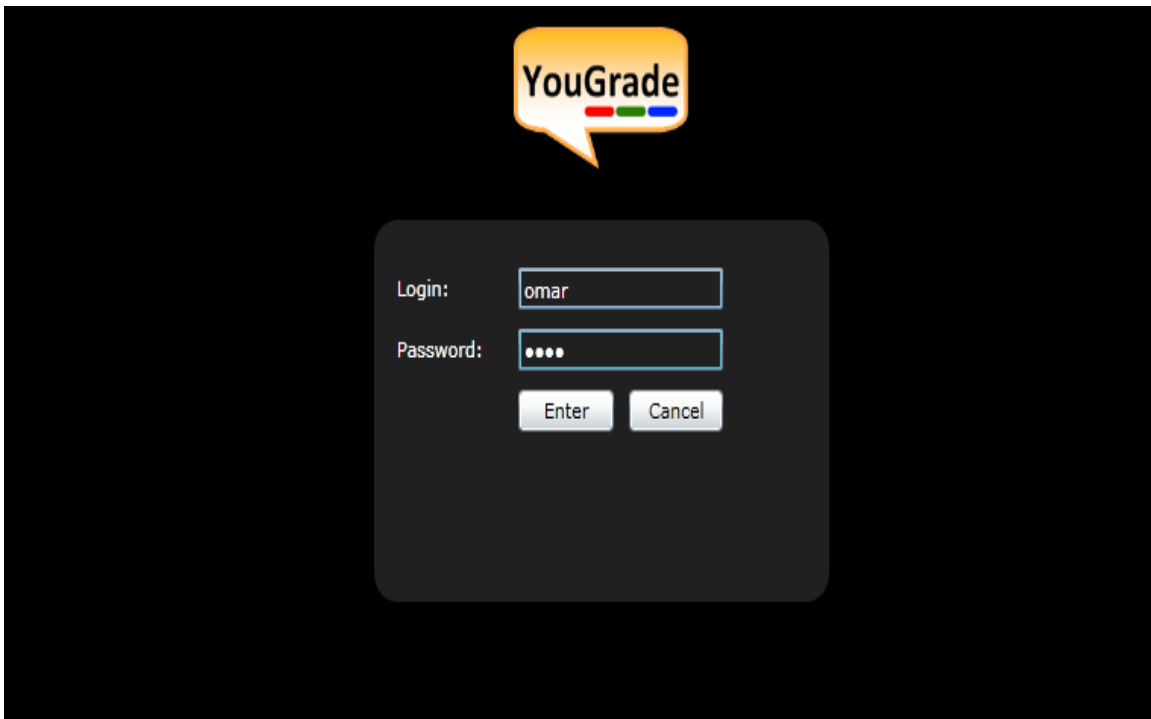


figure 24 The Login View

As you may find out, I didn't pay much attention to this view, but just because I rather focused mostly on the Exam View, which is the real core of the application. If this was a real application, I could use, for example, ASP.NET Membership, which provides powerful authentication features.

On the other side, the LoginViewModel implements the Login property. Notice that this is 2-way binding, so any changes made to the text in the txtLogin element will reflect immediately in the Login property in the LoginViewModel. And any changes in the Login property will affect the text in the txtLogin element.

The Exam View

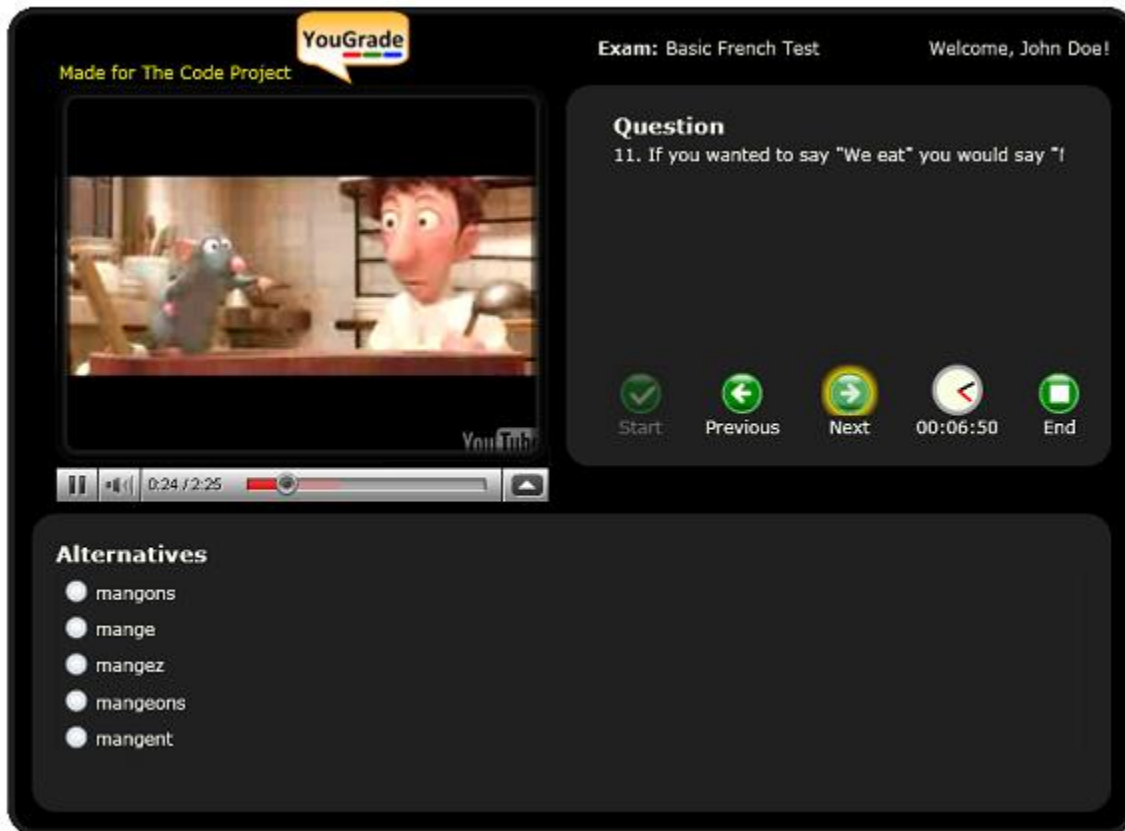


figure 25 The Exam View

Silverlight lacked a control where you could host HTML content in your application, until the release of Silverlight 4 Beta (remember that this article deals with Silverlight 3 only). And even in Silverlight 4, you could only use WebBrowser or HtmlBrush while working on Silverlight Out Of Browser applications. But fortunately for us, DivElements company provides the HtmlHost control for free, and it works with in-browser applications, even with Silverlight 3.

There's one more important thing about the the Exam View: usually, the YouTube player shows advertisement links and links to other YouTube videos. In order to ensure the correct use of YouGrade, the user is not supposed to click such links and navigate away from the Silverlight application

Custom Button Control

As a little XAML exercise, I decided to create a custom button with a nice look, which I called the "Green Button". It's a circular button which looks somewhat plastic, with a light effect on the top and a shadow effect on the bottom. At the center, you can choose which image you are using for the button. That's why the buttons on the ExamView are the same control. The only difference is the central image, as seen in Figure below



figure 26 Custom Button Control

While reading some books about Silverlight, some authors gave me advice to create a new Silverlight Class Library project and put the controls there.

Clock/Timer Control

Our exam candidates will have a limited amount of time to finish the test. This is why we have a timer control.

The timer control is in fact a set of elements inside a Grid control. As you can see in Figure 13, the control is made up by a circular white background with two hands, for the minutes and for the seconds.



figure 27 Clock/Timer Control

Silverlight-Enabled WCF Service

One of the most important points in the YouGrade application is the data access. We must be able to perform a number of data access operations, such as authenticating the user, getting exam definitions, and saving exam results to the database.

As many of you must know, Silverlight applications have no access to local resources, and limited access to network resources. The easiest way to do it is via WCF services (the other ways are HTTP communication and sockets communication). Fortunately, Visual Studio 2008 has a template named Silverlight-Enabled WCF Service which, as the name already says, facilitates the consumption of a WCF service by Silverlight applications, by automatically adding references and modifying the web.config file, for example.

Validating a User

The user is validated by taking a login and password and comparing them to the users in the database

Getting Exam Definition

After validating the user, we must get the exam definition from the database. Notice that we are using Entity Framework for data access. Notice also that the `Include("QuestionDef.Alternative")` method tells the Entity Framework to return not only the exam definition, but also questions and alternatives associated to the exam definition.

Saving Results to the Database

Saving the results to the database is a bit more complicated. Notice in the example below, that I used ExamTakeTO as a parameter, instead of the Exam entity. ExamTakeTO is a DTO (Data Transfer Object), that is, a POCO (Plain Old CLR object) that holds the data for the "exam take", that is, the data containing the user information along with the answers he or she chose for the questions.

Showing Result Report

After the user has finished the test, it's time to show the report.

The user will pass or fail the test, depending on whether the user has scored more or less points than the minimum required by the test (that is, the Minimum Of Correct Answers property of the Exam Definition entity).

As we can see in Figure below, user failed the test because he scored only 4 points, while the minimum required was 10:

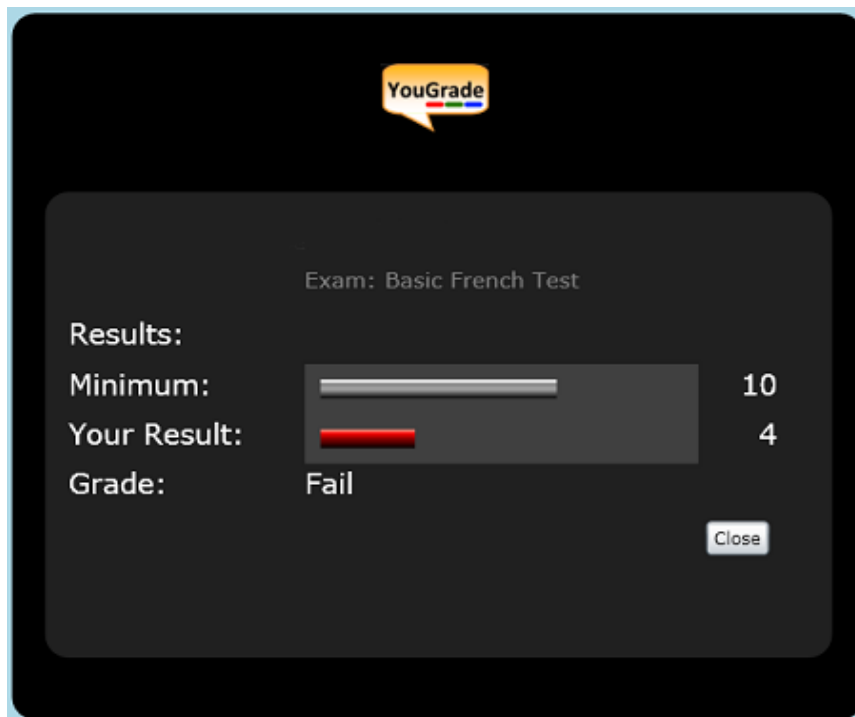


figure 28 The Result

Then user studied a bit more and passed the test in another occasion, scoring 13 points

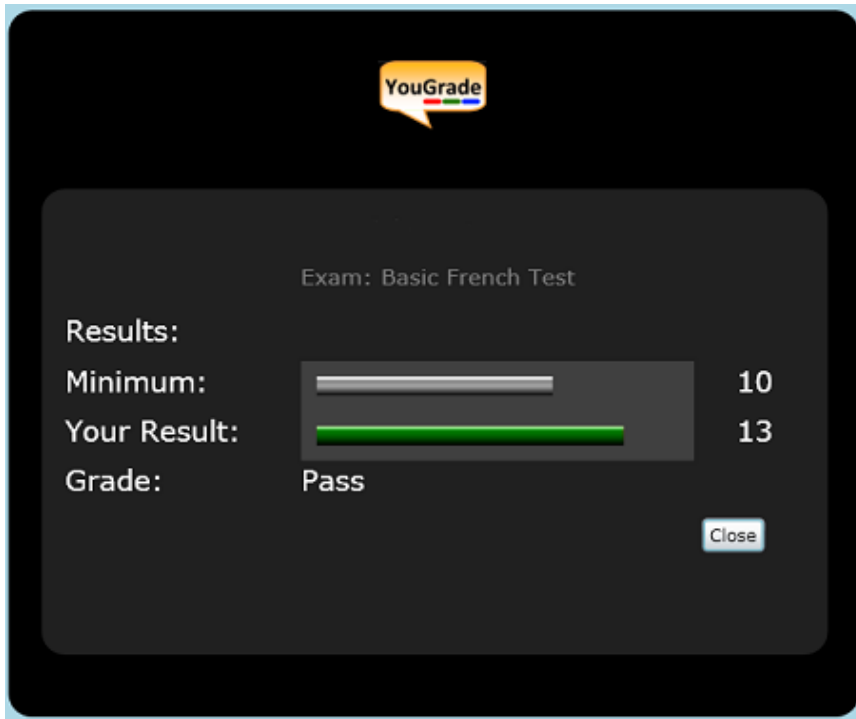


figure 29 The Result

Chapter 4 :

Conclusion

There are still some issues with the application, and some desirable improvements which I hope I'll be correcting soon:

- It would be great to have an exam definition editor. I'm planning to make one with WPF and publish it in a new article.
- In the database model, the user is not linked to the exam definition in any way. It would be nice to have a schedule so that you could assign different exams to different users, control exam availability, validity periods, etc.
- Sometimes the application doesn't properly stop the YouTube audio from playing while switching from one question to another.
- I had some issues while passing the ExamDef entity from the Silverlight SaveExamTake method, and that's why I used transfer objects to pass data. Although it's not a critical problem, it would be nice and cleaner if I used only Entity Framework entities in those communications.

references :

- [1] Software Requirements Specification for project iTest
- [2] IEEE Recommended Practice for Software Requirements Specifications, Software Engineering Standards Committee of the IEEE Computer Society.
- [3] Software Requirements Specification for PPDP Contact Management System (CMS)
- [4] Software Requirements Specification for Problem Based Learning Module
- [5] Software Design Specification (SDS) Acropolis Course Management System
- [6] Software Requirement Specifications, Online Examination System
- [7] http://whatis.techtarget.com/definition/0,,sid9_gci1103696,00.html
- [8] http://www.ehow.com/facts_5156877_preface-book.html
- [9] <http://www.sil.org/lingualinks/literacy/referencematerials/glossaryofliteracyterms/WhatIsAPreface.htm>
- [10] <http://www.deftinfosystems.com/index.php/application/e-education-system/online-examination-system.html>