Online Exam

Requirements Specification and Analysis

Version 0.1

20.11.2016

Muhammed Bilal Kutlu

Güneycan Demir

Çağla Çolak

Serdar Akmehmet

Prepared for

SE301 Software Engineering



Table of Contents

[1.Introduction 4](#__RefHeading__450_1335304141)

[1.1.Purpose of the System 4](#__RefHeading__452_1335304141)

[1.2.Scope of the System 4](#__RefHeading__454_1335304141)

[1.3.Objectives and Success Criteria of the Project 4](#__RefHeading__456_1335304141)

[1.4.Definitions, Acronyms, and Abbreviations 4](#__RefHeading__458_1335304141)

[1.5.Overview 4](#__RefHeading__460_1335304141)

[2.Current System 4](#__RefHeading__462_1335304141)

[3.Proposed System 4](#__RefHeading__464_1335304141)

[3.1.Overview 4](#__RefHeading__466_1335304141)

[3.2.Functional Requirements 4](#__RefHeading__468_1335304141)

[3.3.Nonfunctional Requirements 4](#__RefHeading__470_1335304141)

[Usability 4](#__RefHeading__472_1335304141)

[Reliability 5](#__RefHeading__474_1335304141)

[Performance 5](#__RefHeading__476_1335304141)

[Supportability 5](#__RefHeading__478_1335304141)

[Implementation 5](#__RefHeading__480_1335304141)

[Interface 5](#__RefHeading__482_1335304141)

[Packaging 5](#__RefHeading__484_1335304141)

[Legal 5](#__RefHeading__486_1335304141)

[3.4.System Models 5](#__RefHeading__488_1335304141)

[Scenarios 5](#__RefHeading__490_1335304141)

[Use case model 5](#__RefHeading__492_1335304141)

[Object model 5](#__RefHeading__494_1335304141)

[Dynamic model 5](#__RefHeading__496_1335304141)

[User interface—navigational paths and screen mock-ups 5](#__RefHeading__498_1335304141)

[4.Glossary 5](#__RefHeading__500_1335304141)

[5.References 5](#__RefHeading__502_1335304141)

REQUIREMENTS ANALYSIS DOCUMENT

# Introduction

## Purpose of the System

* Online Examination System is a web-based application that provides users to create exam and allow them to take exam on-line.
* There are three levels of user;

1. Instructor
2. Student
3. Administrator

* We will call Online Examination System as ES.
* ES includes a few information about to Instructor like name, address, school, department etc. and a few information about to Student like name, address, degree, department, class.
* ES provides Instructor to create;

1. New Lecture
2. New Class
3. New Exam
4. New Question
5. List Exam Results

* ES provides Student to;

1. Take Exam
2. List Exam Results
3. View the Question’s answer
4. Answer the Question
5. Submit Exam

* ES can calculate the result of an exam of any student automatically.
* The instructor can list the results of the student after submit of the exam in real-time.

.

## Scope of the System

* ES can be used in educational institutions.
* ES can be used anywhere anytime as it is a web-based application (user’s location doesn’t matter.)
* ES provides simplicity to the users no manual work of preparing and storing the result information.
* ES saves time/money of going to far away centers to give/take exam.
* ES will be used to create a lecture, add this lecture a class, and assign students to this class.
* ES will be used to send invitation to the student about the exam and class.

## Objectives and Success Criteria of the Project

* Proper and conducive project plan
* Assigning tasks to the team members by usig GitHub.
* Reviewing and doing a rework when needed.
* Managing project risks efficiently.
* Allocating time for process improvement

## Definitions, Acronyms, and Abbreviations

* ES: Online Examination System

## Overview

The rest of this ES Requirement Analysis Document contains Overall Description, External Interface Requirements, System Features, Non Functional Requirements and Other Requirements.

# Current System

If the new system will replace an existing system, this section describes the functionality and the problems of the current system. Otherwise, this section describes how the tasks supported by the new system are accomplished now.

# Proposed System

Documents the requirements elicitation and the analysis model of the new system

## Overview

Presents a functional overview of the system.

## Functional Requirements

***Instructor Functional Requirements***

* **Authentication**

Instructor logs on from home page as well. Instructor can log on to the system with SSN number and password.

* **Create New Lecture**

Instructor adds courses he/she gives. Course name and class list is specified here.

* **Create Class**

Instructor creates class list of the course. Thus only the student who takes this course will be able to join this course’s exam.

* **Add Student**

Adds student to class list of course.

* **Delete Student**

Deletes student from class list if necessary like when student drops course or etc.

* **List Student**

Shows current class list of course.

* **Create Exam**

Creating exam with sub functions adding and deleting questions. Questions are updatable after publish but before exam start date and time.

* **Add Question**

Adds a new question and its answer to exam.

* **Delete Question**

Deletes a question from exam.

* **Update Exam**

Updates exam date, questions, or answers before examination date.

* **Update Question**

To update question itself or its answer.

* **Remove Exam**

This function is used for delete whole exam.

* **Remove Lecture**

To remove lecture when necessary.

***Admin Functional Requirements***

* **Authentication;**

Admin part is different from InstructorAuthentication and Student Authentication. Admin should enter the different uml. In this page, admin can login on system with username and password. Password is given by the system database.

* **Edit appointment information**

Admin update information which must be change, like change instructors passwords or instructor authority.

* **List Instructors**

Admin can see instructor’s list .In this list, Admin can add and delete operations.

* **Instructors Request Approve**

Instructors can send request for authorize. Admin can see requests which come from Instructors to take authorize Instructors account.

* **Add New Instructors**

Instructor who want to make exam authorized by Admin to the System.

* **Delete Instructors**

Admin update which is necessary such as a instructor arrive the school.

* **Save**

Admin record all Instructors on the system when he or she entered all information.

* **Log Off**

Admin exit the system when they finished the operations on the system.

***Student Functional Requirements***

* **Authentication**

Student enter his required information into system to authenticate to the system. Student can’t Access any function without authentication.

* **Request Password Remainder**

When student forgot his/her password, he/she can use this option to reset or remain the password.

* **Take Exam**

After authentication, student can list his lessons which he/she takes. He/she can take exam of the lessons which he/she assigned. Student can’t take any exam if the exam time hasn’t come yet.

* **Answer Question**

Student displays the question and answers the question with his/her choice.

* **Submit Exam**

After answered all questions and given answers confirmed, student can submit exam. After submission, student cannot change any information of the exam.

* **List Exam Results**

Student can list his/her exam results of an exam which initiated by the lesson.

* **View Answers**

Student can display the answer of an exam if the exam already submitted.

## Nonfunctional Requirements

## System Models

### Scenarios

**Scenario 1**

|  |
| --- |
| **Scenario name:** Login Admin |
| **Participant actor instances: Admin:Çağla** |
| **Flow of events:**   1. Firstly Çağla writes Examination System/admin on browser. 2. Çağla sees login pages and she enter Username into Username text field also enter password into Password text field on this pages.   a) if Çağla leaves one or more fields empty, system displays a warning message, like "This area can not be empty.".   1. If Username, password and role match to information which are in Online Exam database Çağla can login to the Online Exam System.   a)If the information’s checked from database are not true, system displays a warning message, like "Wrong username or password, please retry.". |

**Scenario 2**

|  |
| --- |
| **Scenario name:** Adding Instructor |
| **Participant actor instances: Admin: Çağla Instructor:Güney** |
| **Flow of events:**   1. Çağla hears that Güney want to make exam in the Online Exam system. 2. Firstly Çağla logs into the Online Exam System. 3. Çağla selects “Add New Instructor” button. 4. Çağla fills the related field with the personal information of the Güney. 5. Then Çağla click “save” button. 6. Online Exam save the Güney’s personal information’s on database . 7. Çağla sees a "Saved successfully" message on screen. Then Çağla logs of. |

|  |
| --- |
| **Scenario name:** Deleting Instructor |
| **Participant actor instances: Admin:**Çağla **Instructor:**Güney |
| **Flow of events:**   1. Çağla hears that Güney leave the school and want to delete his account. 2. Firstly Çağla logs into the ES. 3. Çağla selects “Delete Instructor” button 4. She sees all Instructors who registered in the system. 5. Çağla chooses Güney. 6. Then Çağla click “delete and save” button. 7. OnlineExam finds the Güney on database and delete her all informations on system. 8. The message shown on screen that “The instructor is deleted successfully”. 9. Çağla logs of. |

**Scenario 3**

**Scenario 4**

|  |
| --- |
| **Scenario name:** ListInstructors |
| **Participant actor instances: Admin:**Çağla |
| **Flow of events:**   1. Firstly Çağla logs into the ES. 2. Çağla clicks the "List Instructors" button on administrator main page. 3. She sees all Instructors who registered in the system. 4. Çağla clicks "Back" button. 5. The system directs administrator to his/her main page. 6. Çağla logs of. |

**Scenario 5**

|  |
| --- |
| **Scenario name:** EditInstructorInformations |
| **Participant actor instances: Admin:**Çağla |
| **Flow of events:**   1. Firstly Çağla logs into the ES. 2. Çağla selects a instructor that is going to be editted. 3. Çağla clicks the "Edit Instructor Informations" button. 4. Çağla changes necessary areas with the instructor information of the selected instructor. 5. Çağla clicks the "Save" button. 6. She see a "Saved successfully!" message on a dialogue box. 7. Çağla logs of. |

**Scenario 6**

|  |
| --- |
| **Scenario name:** ApproveRequestofInstructor |
| **Participant actor instances: Admin:**Çağla Instructor:Güney |
| **Flow of events:**   1. Firstly Güney logs into the ES. 2. Güney send request to be instructor to Çağla. 3. Çağla clicks the "Request of Instructors" button. 4. Çağla sees List of Requests of Instructors. 5. Çağla choose Güneys name and click Approve button. 6. Çağla clicks the "Save" button. 7. She see a "Saved successfully!" message on a dialogue box. 8. Çağla logs of. |

**Scenario 7**

|  |
| --- |
| **Use case name:** Logout |
| **Participant actor instances: Admin:**Çağla |
| **Flow of events:**   1. Çağla clicks "Logout" button. 2. System redirects administrator to the login page. |

**Scenario 8**

|  |
| --- |
| **Scenario name:** Instructor Register |
| **Participant actor instances:** **Okan:**Instructor |
| **Flow of events:**   1. Firstly Okan enters [www.ES.com](http://www.ES.com) on browser. 2. ES welcomes Okan with main page. Okan sees register button below the login fields and clicks register. 3. He will be directed to register page after click for choose register as an instructor or student first. 4. After choosing register as an instructor he will be directed to instructor register fields to fill. 5. Okan fills field with information needed. 6. Okan clicks done. 7. Account is created. |

**Scenario 9**

|  |
| --- |
| **Scenario name:** Instructor Login |
| **Participant actor instances:** **Okan:**Instructor |
| **Flow of events:**   1. Firstly Okan enters [www.ES.com](http://www.ES.com) on browser. 2. ES welcomes Okan with main page. Okan fills the login fields and clicks login. 3. If SSN and password matches with database he accesses his account. |

**Scenario 10**

|  |
| --- |
| **Scenario name:** Create New Lecture |
| **Participant actor instances:** **Muge:**Instructor |
| **Flow of events:**   1. This event starts with Muge’s click to the Create a new lecture button. 2. Muge is directed to the create lecture page. 3. Muge fills the lecture informations. 4. Muge clicks done. 5. Lecture is added to the Muge’s database. |

**Scenario 11**

|  |
| --- |
| **Scenario name:** Create Class(Empty) |
| **Participant actor instances:** **Muge:**Instructor |
| **Flow of events:**   1. This event starts with Muge’s click to the Create Class button. 2. Muge is directed to the Create Class page. 3. Muge clicks done without adding any student. 4. An empty class is created for lecture. |

**Scenario 12**

|  |
| --- |
| **Scenario name:** Create Class |
| **Participant actor instances:** **Muge:**Instructor |
| **Flow of events:**   1. This event starts with Muge’s click to the Create Class button. 2. Muge is directed to the Create Class page. 3. Muge clicks add student button and can continue add or delete. 4. Muge clicks done. 5. A class list is constructed and setted to the database. |

**Scenario 13**

|  |
| --- |
| **Scenario name:** Add student |
| **Participant actor instances:** **Muge:**Instructor |
| **Flow of events:**   1. This event starts with Muge’s click to the add Student button after opened Class page. 2. System shows to Muge fields to fill with student info. 3. Muge fills fields with the student information which is taking course. 4. Muge clicks done. 5. A new student added to class list and setted to the database. |

**Scenario 14**

|  |
| --- |
| **Scenario name:** Delete student |
| **Participant actor instances:** **Muge:**Instructor |
| **Flow of events:**   1. This event starts with Muge’s click to the delete Student button after opened Class page. 2. Muge selects a student. 3. Muge clicks delete student button. 4. Student is deleted from class list, database updated. |

**Scenario 15**

|  |
| --- |
| **Scenario name:** List student |
| **Participant actor instances:** **Muge:**Instructor |
| **Flow of events:**   1. This event starts with Muge’s click to the list Student button after opened Class page. 2. System shows to Muge class list. |

**Scenario 16**

|  |
| --- |
| **Scenario name:** Create Exam |
| **Participant actor instances:** **Muge:**Instructor |
| **Flow of events:**   1. This flow starts with Muge’s click to the create exam button. 2. System shows to Muge fields to fill with exam info. 3. Müge fills fields with the exam information which is date, time, etc. 4. Then Muge adds questions. 5. Muge clicks done. |

**Scenario 17**

|  |
| --- |
| **Scenario name:** Add Question |
| **Participant actor instances:** **Muge:**Instructor |
| **Flow of events:**   1. This event starts with Muge’s click to the add question button. 2. System directs Muge to question adding page. 3. Muge first selects a question type. 4. Muge adds question itself. 5. Muge adds answer(s). 6. Muge clicks done. 7. System adds question to the exam and database. |

**Scenario 18**

|  |
| --- |
| **Scenario name:** Delete Question |
| **Participant actor instances:** **Muge:**Instructor |
| **Flow of events:**   1. This event starts with Muge’s click to the delete question button. 2. System directs Muge to exam page. 3. Muge first selects a question. 4. Muge clicks delete button. 5. System deletes question from the exam and database. |

**Scenario 19**

|  |
| --- |
| **Scenario name:** Update Exam |
| **Participant actor instances:** **Muge:**Instructor |
| **Flow of events:**   1. This event starts with Muge’s click to the update exam button. 2. System directs Muge to exam page. 3. Muge updates exam information’s (date, time etc.). 4. Muge click’s done. 5. System updates exam information and database. |

**Scenario 20**

|  |
| --- |
| **Scenario name:** Update Question |
| **Participant actor instances:** **Muge:**Instructor |
| **Flow of events:**   1. This event starts after Muge’s selecting a question. 2. System opens Muge to question page. 3. Muge clicks selects update question. 4. Muge updates question or answer(s). 5. Muge clicks done. 6. System updates question and database. |

**Scenario 21**

|  |
| --- |
| **Scenario name:** Remove Exam |
| **Participant actor instances:** **Muge:**Instructor |
| **Flow of events:**   1. This event starts after Muge clicks remove exam button. 2. System deletes exam from lecture and database. |

**Scenario 22**

|  |
| --- |
| **Scenario name:** Remove Lecture |
| **Participant actor instances:** **Muge:**Instructor |
| **Flow of events:**   1. Flow starts when Muge clicks to the remove lecture button. 2. System removes lecture from Muge’s account and database. |

**Scenario 23**

|  |
| --- |
| **Scenario name:** List students results |
| **Participant actor instances:** **Muge:**Instructor |
| **Flow of events:**   1. Flow starts with Muge’s click to show results button. 2. System opens the student grade list for the lecture to Muge. |

**Scenario 24**

|  |
| --- |
| **Scenario name:** InstructorLogoff |
| **Participant actor instances:** **Muge:**Instructor |
| **Flow of events:**   1. Flow starts with Muge’s click to log off button. 2. System directs Muge to the main page. 3. Muge logged off. |

**Scenario 25**

|  |
| --- |
| **Scenario name:** Student Register |
| **Participant actor instances: Robert:**Student |
| **Equipment:** Any computer with a supported browser. |
| **Flow of events:**   1. Firstly Robert writes [www.ES.com](http://www.ES.com) on browser. 2. ES welcomes the Robert with main page and, He realized that he was not registered on the system. So, He clicks the Register button which is located below login section. 3. Roberts chooses the student section to register as a student. 4. After choosing student section he is redirect to the [www.ES.com/StudentRegister.aspx](http://www.ES.com/StudentRegister.aspx) page to register. 5. Robert fills the information which is required to finish the register process. 6. Robert entered all information, and clicked to Submit button to finish process. 7. Now, Robert can access to the system with his information which is provided in register section. |
| **Entry Condition:** Robert should have clicked to the register button. |
| **Exit Condition:** Register information has set to database successfully. |

**Scenario 26**

|  |
| --- |
| **Scenario name:** Student Login |
| **Participant actor instances: Maria:**Student |
| **Equipment:** Any computer with a supported browser. |
| **Flow of events:**   1. Firstly Maria writes [www.ES.com](http://www.ES.com) on browser. 2. ES welcomes the Maria with main page and, She fills the login fields with his SSN and Password to login. 3. After entered SSN and Password she clicks to login button. 4. After clicking login section she is redirect to the [www.ES.com/StudentDashboard.aspx](http://www.ES.com/StudentDashboard.aspx) page. 5. Now Maria can access functions of the system. |
| **Entry Condition:** Maria should be registered to the system. |
| **Exit Condition:** SSN number which entered by Maria and SSN number which submitted on database is matching and has been successfully accessed to the student dashboard. |

**Scenario 27**

|  |
| --- |
| **Scenario name:** Take Exam |
| **Participant actor instances: Filip:** Student |
| **Equipment:** Any computer with a supported browser. |
| **Flow of events:**   1. Firstly Filip writes [www.ES.com](http://www.ES.com) on browser. 2. ES welcomes the Filip with main page and, He fills the login fields with his SSN and Password to login, Filip authenticated. 3. Filip list his upcoming exams by clicking upcoming exams button. 4. After clicking upcoming exams section he is redirect to the [www.ES.com/upcomingexams.aspx](http://www.ES.com/upcomingexams.aspx) page. 5. Filip clicks the CSE111 Midterm 1 exam which the date of the exam is came. 6. Filip can now answer the question of CSE111 Midterm. |
| **Entry Condition:** Filip is invited to the exam and the date of the exam should came. |
| **Exit Condition:** None. |

**Scenario 28**

|  |
| --- |
| **Scenario name:** View Answers |
| **Participant actor instances: Daniel:** Student |
| **Equipment:** Any computer with a supported browser. |
| **Flow of events:**   1. Firstly Daniel writes [www.ES.com](http://www.ES.com) on browser. 2. ES welcomes the Daniel with main page and, He fills the login fields with his SSN and Password to login, Daniel authenticated. 3. Daniel lists his upcoming exams by clicking upcoming exams button. 4. After clicking upcoming exams section he is redirected to the [www.ES.com/upcomingexams.aspx](http://www.ES.com/upcomingexams.aspx) page. 5. Daniel clicks the MATH101 Midterm 2 exam which the date of the exam is came. 6. Daniel answers the questions of MATH101 Midterm 2 exam 7. After he answered all the questions, ES shows a “view answers” button which is allow Daniel to view the answer of the MATH101 Midterm 2 exam. 8. Daniel clicks view answers button to display the answers. 9. ES shows the answers in a table. 10. Daniel logs off. |
| **Entry Condition:** Daniel is invited to the exam and the date of the exam should come. |
| **Exit Condition:** Daniel should have answered the entire question and submit the exam. |

**Scenario 29**

|  |
| --- |
| **Scenario name:** List Exam Results |
| **Participant actor instances: Lisa:** Student |
| **Equipment:** Any computer with a supported browser. |
| 1. **Flow of events:** 2. Firstly Lisa writes [www.ES.com](http://www.ES.com) on browser. 3. ES welcomes the Lisa with main page and, she fills the login fields with his SSN and Password to login, Lisa authenticated. 4. Lisa list her exams which is already attended by clicking the list attended exam section. 5. After clicking list attended exam section she is redirected to the [www.ES.com/attendedexams.aspx](http://www.ES.com/attendedexams.aspx) page. 6. ES shows the result of the exam in an order. ES shows the result of the exam in a row. 7. Lisa checks his results. 8. Lisa logs off. |
| 1. **Entry Condition:** Lisa should have already attended an exam. |
| 1. **Exit Condition:** Lisa clicked the list attended exams section. |

**Scenario 30**

|  |
| --- |
| **Scenario name:** Answer Question |
| **Participant actor instances: Bilal:** Student |
| **Equipment:** Any computer with a supported browser. |
| 1. **Flow of events:** 2. Firstly Bilal writes [www.ES.com](http://www.ES.com) on browser. 3. ES welcomes the Bilal with main page and, He fills the login fields with his SSN and Password to login, Bilal authenticated. 4. Bilal lists his upcoming exams by clicking upcoming exams button. 5. After clicking upcoming exams section he is redirected to the [www.ES.com/upcomingexams.aspx](http://www.ES.com/upcomingexams.aspx) page. 6. Bilal clicks the ENG102 Final exam which the date of the exam is came. 7. ES shows a brief information about exam before the exam is started. 8. Bilal checks the time duration, number of questions, exam type and rules. 9. Bilal clicks start the exam. 10. ES displays the questions in an order and answers shown as well. 11. Bilal choose an answer and clicks next question button. 12. After answered all question, Bilal submits the exam and logs off. |
| 1. **Entry Condition:** Bilal is invited to the exam and the date of the exam should come. |
| 1. **Exit Condition:** Bilal should have chosen an answer. |

**Scenario 31**

|  |
| --- |
| **Scenario name:** Fixing Answer of a Question |
| **Participant actor instances: Murat:** Student |
| **Equipment:** Any computer with a supported browser. |
| **Flow of events:**   1. Firstly Murat writes [www.ES.com](http://www.ES.com) on browser. 2. ES welcomes the Murat with main page and, He fills the login fields with his SSN and Password to login, Murat authenticated. 3. Murat lists his upcoming exams by clicking upcoming exams button. 4. After clicking upcoming exams section he is redirected to the [www.ES.com/upcomingexams.aspx](http://www.ES.com/upcomingexams.aspx) page. 5. Murat clicks the MATH102 Final exam which the date of the exam is came. 6. ES shows a brief information about exam before the exam is started. 7. Murat checks the time duration, number of questions, exam type and rules. 8. Murat clicks start the exam. 9. ES displays the questions in an order and answers shown as well. 10. Murat choose an answer and clicks next question button. 11. Murat now wishes the change the answer of the previous question. 12. Murat clicks previous question button to display previous question. 13. Murat can now change the answer of the question. 14. After answered all question and confirmed all the information, Murat logs off. |
| 1. **Entry Condition:** Bilal is invited to the exam and the date of the exam should come. |
| 1. **Exit Condition:** Bilal should have clicked the previous question button. |

**Scenario 32**

|  |
| --- |
| **Scenario name:** Submit Exam |
| **Participant actor instances: Filip:** Student |
| **Equipment:** Any computer with a supported browser. |
| **Flow of events:**   1. Firstly Filip writes [www.ES.com](http://www.ES.com) on browser. 2. ES welcomes the Filip with main page and, He fills the login fields with his SSN and Password to login, Filip authenticated. 3. Filip list his upcoming exams by clicking upcoming exams button. 4. After clicking upcoming exams section he is redirect to the [www.ES.com/upcomingexams.aspx](http://www.ES.com/upcomingexams.aspx) page. 5. Filip clicks the TUR101 Final exam which the date of the exam is came. 6. Filip answers all question of TUR101 Final. 7. After all question answered, Filip clicks submit button to finish process. 8. Filip logs off. |
| **Entry Condition:** Filip is invited to the exam and the date of the exam should come. |
| **Exit Condition:** None. |

### Use case model

**Use Case 1**

|  |
| --- |
| **Use Case Name:** Create New Lecture |
| **Participant actor instances:** Instructor |
| **Flow of events:**   1. This event starts with Instructor’s click to the Create a new lecture button. 2. Instructor is directed to the create lecture page. 3. Instructor fills the lecture information’s. 4. Instructor clicks done. 5. Lecture is added to the database. |
| **Entry Condition:** User should be logged on as instructor. |
| **Exit Condition:** Lecture is created. Or cancelled if user left page without click done |
| **Alternative flow:**  **4a.** Instructor lefts page without click done.   1. System cancelled function automatically. |
| **Quality Requirement:**  **1.**  System response to the actor arrives in a few seconds. |

**Use Case 2**

|  |
| --- |
| **Use Case Name:** Create Class |
| **Participant actor instances:** Instructor |
| **Flow of events:**   1. This event starts with Instructor’s click to the Create Class button. 2. Instructor is directed to the Create Class page. 3. Instructors clicks add student button and can continue add or delete. 4. Instructor clicks done 5. A class list is constructed and settled to the database. |
| **Entry Condition:** User should be logged on as instructor. And also should already create a lecture. |
| **Exit Condition:** A class is created. Or alternatively empty one is created. |
| **Alternative flow:**  **3a.** Instructor clicks done.  **1.** System created an empty class list. |
| **Quality Requirement:**  **1.** System response to the actor arrives in a few seconds. |

**Use Case 3**

|  |
| --- |
| **Use Case Name:** Add student |
| **Participant actor instances:** Instructor |
| **Flow of events:**   1. This event starts with Instructor’s click to the add Student button after opened Class page. 2. System shows to Instructor fields to fill with student info. 3. Instructor fills fields with the student information which is taking course. 4. Instructor clicks done. 5. A new student added to class list and settled to the database. |
| **Entry Condition:** User should be logged on as instructor. And also should already create a lecture and class. |
| **Exit Condition:** A student is added to class list. |
| **Quality Requirement:**  **1.** System response to the actor arrives in a few seconds. |

**Use Case 4**

|  |
| --- |
| **Use Case Name:** Delete student |
| **Participant actor instances:** Instructor |
| **Flow of events:**   1. This event starts with Instructor’s click to the delete Student button after opened Class page. 2. Instructor selects a student. 3. Instructor clicks delete student button. 4. Student is deleted from class list, database updated. |
| **Entry Condition:** User should be logged on as instructor. And also should already created a lecture and class. |
| **Exit Condition:** Selected student(s) is deleted from class list. |
| **Quality Requirement:**  **1.** System response to the actor arrives in a few seconds. |

**Use Case 5**

|  |
| --- |
| **Use Case Name:** List student |
| **Participant actor instances:** Instructor |
| **Flow of events:**   1. This event starts with Instructor’s click to the list Student button after opened Class page. 2. System shows to Instructor class list. |
| **Entry Condition:** User should be logged on as instructor. And also should already create a lecture and class. |
| **Quality Requirement:**   1. System response to the actor arrives in a few seconds. 2. List is showed in alphabetic order. |
| **Exit Condition:** A student is added to class list. |

**Use Case 6**

|  |
| --- |
| **Use Case Name:** Create Exam |
| **Participant actor instances:** Instructor |
| **Flow of events:**   1. This flow starts with Instructor’s click to the create exam button. 2. System shows to Instructor fields to fill with exam info. 3. Instructor fills fields with the exam information which is date, time, etc. 4. Then Instructor adds questions. 5. Instructor clicks done. |
| **Entry Condition:** User should already create a lecture. |
| **Exit Condition:** An exam is created. |
| **Quality Requirement:**   1. System response to the actor arrives in a few seconds. 2. Instructor can select day from a calendar to make his job easy. |

**Use Case 7**

|  |
| --- |
| **Use Case Name:** Add Question |
| **Participant actor instances:** Instructor |
| **Flow of events:**   1. This event starts with Instructor’s click to the add question button. 2. System directs Instructor to question adding page. 3. Instructor first selects a question type. 4. Instructor adds question itself. 5. Instructor adds answer(s). 6. Instructor clicks done. 7. System adds question to the exam and database. |
| **Entry Condition:** User should be created an exam. |
| **Exit Condition:** A question is added to exam. |
| **Alternative flow:**  **6a.** Instructor doesn’t click done and left page.  **1.**System cancelled add student function. |
| **Quality Requirement:**  **1.** System response to the actor arrives in a few seconds. |

**Use Case 8**

|  |
| --- |
| **Use Case Name:** Delete Question |
| **Participant actor instances:** Instructor |
| **Flow of events:**   1. This event starts with Instructor’s click to the delete question button. 2. System directs Instructor to exam page. 3. Instructor first selects a question. 4. Instructor clicks delete button. 5. System deletes question from the exam and database. |
| **Entry Condition:** User should be created an exam. |
| **Exit Condition:** A question is deleted from exam. |
| **Quality Requirement:**  **1.** System response to the actor arrives in a few seconds. |

**Use Case 9**

|  |
| --- |
| **Use Case name:** Update Exam |
| **Participant actor instances:** Instructor |
| **Flow of events:**   1. This event starts with Instructor’s click to the update exam button. 2. System directs Instructor to exam page. 3. Instructor updates exam information’s (date, time etc.). 4. Instructor clicks done. 5. System updates exam information and database. |
| **Entry Condition:** User should be created an exam. |
| **Exit Condition:** Exam is updated. |
| **Quality Requirement:**  **1.** System response to the actor arrives in a few seconds. |

**Use Case 10**

|  |
| --- |
| **Use Case Name:** Update Question |
| **Participant actor instances:** Instructor |
| **Flow of events:**   1. This event starts after Instructor’s selecting a question. 2. System opens Instructor to question page. 3. Instructor clicks update question. 4. Instructor updates question or answer(s). 5. Instructor clicks done. 6. System updates question and database. |
| **Entry Condition:** User should be created an exam. |
| **Exit Condition:** Question is updated. |
| **Quality Requirement:**  **1**. System response to the actor arrives in a few seconds. |

**Use Case 11**

|  |
| --- |
| **Use Case Name:** Remove Exam |
| **Participant actor instances:** Instructor |
| **Flow of events:**   1. This event starts after Instructor clicks remove exam button. 2. System deletes exam from lecture and database. |
| **Entry Condition:** User should be created an exam. |
| **Exit Condition:** Exam is removed. |
| **Quality Requirement:**  **1.** System response to the actor arrives in a few seconds. |

**Use Case 12**

|  |
| --- |
| **Use Case Name:** Remove Lecture |
| **Participant actor instances:** Instructor |
| **Flow of events:**   1. Flow starts when Instructor clicks to the remove lecture button. 2. System removes lecture from Instructor’s account and database. |
| **Entry Condition:** User should be created a lecture. |
| **Exit Condition:** Lecture is removed, database changed. |
| **Quality Requirement:**  **1.** System response to the actor arrives in a few seconds. |

**Use Case 13**

|  |
| --- |
| **Use Case Name:** List students results |
| **Participant actor instances:** Instructor |
| **Flow of events:**   1. Flow starts with Instructor’s click to show results button. 2. System opens the student grade list for the lecture to Instructor. |
| **Entry Condition:** Instructor should be logged in. |
| **Exit Condition:** Instructor has seen grade list of the exam. |
| **Quality Requirement:**  **1.** System response to the actor arrives in a few seconds. |

**Use case 14**

|  |
| --- |
| **Use case name:** Login |
| **Participant actors:** Initiated by Administrator |
| **Flow of events:**   1. Administrator displays the login screen of the application. 2. Administrator enters username and password to the related areas. 3. Administrator clicks "Login" button. 4. System checks the information’s from database. 5. System fetches the main screen of administrator. |
| **Entry Condition:** 1. Administrator displays the login screen of the system.  Administrator saves his/her information’s to the database. |
| **Exit Condition:** Administrator clicks "Login" button and system displays the administrator main page. |
| **Quality Requirements**: 2\*. If administrator leaves one or more fields empty, system displays a warning message, like "This area cannot be empty.".  4\*. If the information’s checked from database are not true, system displays a warning message, like "Wrong username or password, please retry.” |

**Use case 15**

|  |
| --- |
| **Use case name:** ListInstructors |
| **Participant actors:** Initiated by Administrator |
| **Flow of events:**   1. Administrator clicks the "List Instructors" button on administrator main page. 2. System fetches the screen which list of personnel’s can be displayed. 3. Administrator clicks "Back" button. 4. The system directs administrator to his/her main page. |
| **Entry Condition:** Administrator logs in to the Examination System.  Administrator displays his/her main page. |
| **Exit Condition:** Administrator clicks "Back" button and system directs administrator to his/her main page. |
| **Quality Requirements:** None. |

**Use case 16**

|  |
| --- |
| **Use case name:** AddNewInstructor |
| **Participant actors:** Initiated by Administrator |
| **Flow of events:**   1. Administrator clicks "Add New Instructor" button. 2. System fetches the screen that new personnel can be added. 3. Administrator fills the related areas with the personal information of the new instructor.   a) If administrator leaves a field empty, system displays a warning message, like "This area cannot be empty.".   1. Administrator clicks "Save" button.   a) If administrator closes the page without saving, system does not save the informations.   1. System saves the new instructor and his/her information’s to the database. 2. System displays a "Saved successfully!" message on a dialogue box.   a) If system fails to save information’s to the database, it displays a "saving is unsuccessful" message   1. System redirects administrator to the administrator main page. |
| **Entry Condition:** Administrator logs in to the Examination System.  Administrator displays his/her main page. |
| **Exit Condition:** System saves the new personnel and his/her information’s to the database, then system displays a "Saved successfully!" message and system redirects administrator to the administrator main page. |
| **Quality Requirements:** 3\*. If administrator leaves a field empty, system displays a warning message, like "This area cannot be empty.".  4\*. If administrator closes the page without saving, system does not save the information’s.  6\*. If system fails to save information’s to the database, it displays a "saving is unsuccessful" message. |

**Use case 17**

|  |
| --- |
| **Use case name:** Delete Instructor |
| **Participant actors:** Initiated by Administrator |
| **Flow of events:**   1. Administrator selects an instructor that is going to be deleted.   a) If administrator does not select a instructor from instructor list, system displays a warning message, like "A selection must be done to delete"   1. Administrator clicks the "Delete Instructor" button. 2. System deletes the selected instructor record from database. 3. System displays a "Deleted successfully!" message on a dialogue box.   a) If system fails to delete information’s from the database, it displays a "deleting is unsuccessful" message.   1. System reloads the instructor list screen. |
| **Entry Condition:** 1. Administrator logs in to the Examination System.  2. Administrator displays the screen that list of personnel’s is displayed. |
| **Exit Condition:** System displays a "Deleted successfully!" message and system reloads the personnel list screen. |
| **Quality Requirements:** 1\*. If administrator does not select a personnel from personnel list, system displays a warning message, like "A selection must be done to delete".  4\*. If system fails to delete information’s from the database, it displays a "deleting is unsuccessful" message. |

**Use case 18**

|  |
| --- |
| **Use case name:** EditInstructorInformations |
| **Participant actors:** Initiated by Administrator |
| **Flow of events:**   1. Administrator selects a instructor that is going to be edited.   a) If administrator does not select a instructor from instructor list, system displays a warning message, like "A selection must be done to edit".   1. Administrator clicks the "Edit Instructor Information’s" button. 2. System fetches the screen that displays the information’s of selected personnel. 3. Administrator changes necessary areas with the instructor information of the selected instructor.   a) If administrator leaves a field empty, system displays a warning message, like "This area cannot be empty.".   1. Administrator clicks the "Save" button.   a) If administrator closes the page without saving, system does not save the information’s.   1. System saves the changes to the database. 2. System displays a "Saved successfully!" message on a dialogue box.   a) If system fails to save information’s to the database, it displays a "saving is unsuccessful" message.   1. System redirects administrator to the administrator main page. |
| **Entry Condition:** Administrator logs in to the Examination System.  Administrator displays the screen that list of personnel’s is displayed. |
| **Exit Condition:** System saves the changes to the database and displays a "Saved successfully!" message, then redirects administrator to the administrator main page |
| **Quality Requirements:** 1\*. If administrator does not select a instructor from instructor list, system displays a warning message, like "A selection must be done to edit".  4\*. If administrator leaves a field empty, system displays a warning message, like "This area cannot be empty.".  5\*. If administrator closes the page without saving, system does not save the informations.  7\*. If system fails to save information’s to the database, it displays a "saving is unsuccessful" message. |

**Use case 19**

|  |
| --- |
| **Use case name:** ApproveRequestofInstructor |
| **Participant actor instances:** Initiated by Administrator |
| **Flow of events:**   1. Administor clicks the "Request of Instructors" button. 2. System shows List of Requests of Instructors. 3. Administor choose instructor name and click "Approve" button. 4. Administor clicks the "Save" button. 5. System shows a "Saved successfully!" message on a dialogue box. |
| **Entry Condition:** Administrator logs in to the Examination System.  Administrator displays his/her main page. |
| **Exit Condition:** Administrator clicks "Logout" button and system redirects administrator to the login page. |
| **Quality Requirements:** 3\* If administrator does not select any request of instructor, and click approve system displays a warning message, like "You should choose a request".  5\*. If administrator closes the page without saving, system does not save the information’s.  6\*. If system fails to save information’s to the database, it displays a "saving is unsuccessful" message. |

**Use case 20**

|  |
| --- |
| **Use case name:** Logout |
| **Participant actors:** Initiated by Administrator |
| **Flow of events:**   1. Administrator clicks "Logout" button. 2. System redirects administrator to the login page. |
| **Entry Condition:** Administrator logs in to the Examination System.  Administrator displays his/her main page. |
| **Exit Condition:** Administrator clicks "Logout" button and system redirects administrator to the login page. |
| **Quality Requirements:** None. |

**Use case 21**

|  |
| --- |
| **Use case name:** Student Login |
| **Participant actors:** Initiated by Student |
| **Flow of events:**   1. Student types [www.es.com](http://www.es.com) on browser. 2. ES System responds by displaying the login screen of the application. 3. Student enters SSN and password to the login fields. 4. Student clicks "Login" button. 5. If the SSN and password match with the SSN and password which is stored in database. Student authenticates. 6. ES System fetches the dashboard of student. |
| **Entry Condition:** Student visits the URL page of the system. |
| **Exit Condition:** Student clicks "Login" button and entered information should be correct. |
| **Quality Requirements:**  3\*. If Student leaves one or more fields empty, system displays a warning message, like "This area cannot be empty.”  5\*. If the information’s checked from database are not true, system displays a warning message, like "Wrong username or password, please retry.” |

**Use case 22**

|  |
| --- |
| **Use case name:** Student Register |
| **Participant actors:** Initiated by Student |
| **Flow of events:**   1. Student types [www.es.com](http://www.es.com) on browser. 2. ES System responds by showing the main page of ES. 3. Student clicks the register button. 4. ES System responds by redirecting to the register page and displaying the page. 5. Student fills the required information to be registered. 6. Student clicks submit button. 7. System inserts the information to the database which is provided by Student. 8. System redirecting student to login page. |
| **Entry Condition:** Student visits the URL page of the system. |
| **Exit Condition:** Student clicks "Register" button and entered information should be correct. |
| **Quality Requirements:**  5\*- If Student leaves one or more fields empty, system displays a warning message, like "This area cannot be empty.”  6\*-The information that user entered is checking and verified by ES system in real-time. |

**Use case 23**

|  |
| --- |
| **Use case name:** Take Exam |
| **Participant actors:** Initiated by Student |
| **Flow of events:**   1. Student clicks upcoming exams section. 2. System checks the database if is there any exam which is related to the current student, fetching data from database (if there any) and displays the upcoming exams page. 3. Student clicks an exam to take. 4. ES System checks the exam date and if the exam data has come, system redirects the students to the [www.es.com/ExamIdentificationNumber](http://www.es.com/ExamIdentificationNumber) 5. ES System displays the brief information of the current exam. 6. Student clicks start button to start exam. 7. ES system starts the time duration for the current exam. 8. Student answers the questions. |
| **Entry Condition:** Student should be authenticated. |
| **Exit Condition:** Student has displayed the exam questions. |
| **Quality Requirements:**  2\*- If there is no upcoming exam for the initiated student, ES system displays a message like “There is no upcoming exams”  4-\*If the exam date has come or its already finish, ES system shows an error message.  6\*-If the student to be late to click “start” button, ES system shows an error message and redirects student to main page.  8\*-ES system checks the duration while every single question answered. If the student is late to answer the question, the student will be directed to the homepage and the entry will be blocked. |

**Use case 24**

|  |
| --- |
| **Use case name:** List Exam Results |
| **Participant actors:** Initiated by Student |
| **Flow of events:**   1. Student clicks list exam results section. 2. System checks the database if is there any exam result which is related to the current student, fetching data from database (if there any) and displays the exam results section. 3. System fills the fetched data to the table in an order. 4. Student checks his result. |
| **Entry Condition:** Student should be authenticated. |
| **Exit Condition:** Student has displayed its own exam result. |
| **Quality Requirements:**  **2\*-** If there is no exam that the student can enter, “there is no exam waiting for the student” to be shown by the ES system.  3\*- If there is no exam waiting for the student, an empty table is shown by ES system. |

**Use case 25**

|  |
| --- |
| **Use case name:** View Answers |
| **Participant actors:** Initiated by Student |
| **Flow of events:**   1. Student clicks the submit exam button to finish process. 2. ES System inserts all the answers to the database and responds by displaying “View Correct Answer” button. 3. Student clicks “View Correct Answers” button. 4. ES System fetch the correct answers from the database which is related to the current exam and redirecting student to the [www.es.com/CorrectAnswers-ExamIdentificationNumber](http://www.es.com/CorrectAnswers-ExamIdentificationNumber) page. 5. Student checks the correct answers and clicks back button to finish process. |
| **Entry Condition(s):**   1. Student should be authenticated. 2. Student should have submitted an exam. |
| **Exit Condition(s):**   1. ES system successfully calculated the exam result of the student. 2. ES system should have inserted the student to the attended list. |
| **Quality Requirements:**  **1-\*** If students are late to finish the exam test is considered invalid and is directed to the homepage.  3\*- Students cannot access this section without finishing the exam.  4.\* If the answer of the exam is not entered by the Instructor, this part is not accessible. |

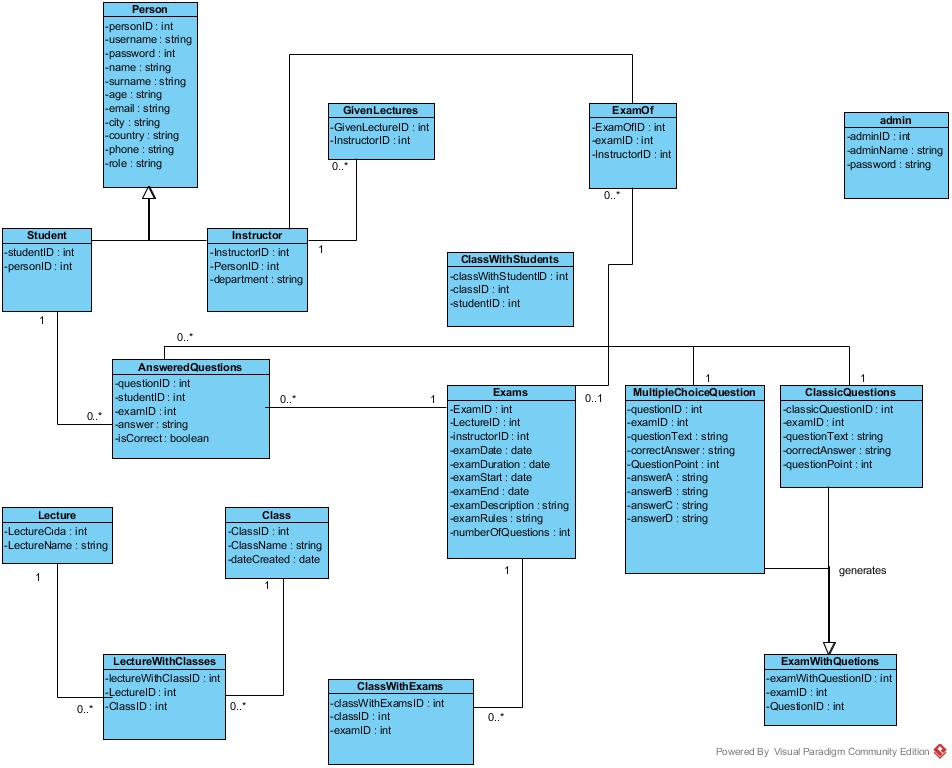
**Use case 26**

|  |
| --- |
| **Use case name:** Answer a Question |
| **Participant actors:** Initiated by Student |
| **Flow of events:**   1. Student clicks start button to start the exam. 2. ES System checks the exam identification number and displays it on [www.es.com/ExamID-QuestionID](http://www.es.com/ExamID-QuestionID) 3. Student picks its answer by clicking the one of the radio buttons. 4. ES System responds by saving the answer to an Array. |
| **Entry Condition(s):**   1. Student should be authenticated. 2. The exam duration time is should be alive. |
| **Exit Condition:**   1. ES system should have added the answer to the array. |
| **Quality Requirements:**  3-\* If any chic is not selected, the answer will not be added to the ES system. |

**Use case 27**

|  |
| --- |
| **Use case name:** Submit Exam |
| **Participant actors:** Initiated by Student |
| **Flow of events:**  1. Student answers all the questions for the current exam.   1. Student clicks the submit exam button. 2. ES System inserts all the answers which are provided by the student to the database. 3. ES System adds the current student to the attended list as attended for the current exam. |
| **Entry Condition(s):**   1. Student should be authenticated. 2. The exam requirements must be fulfilled. |
| **Exit Condition(s):**   1. ES system successfully calculated the exam result of the student. 2. ES system should have inserted the student to the attended list. 3. ES system should have deleted the current exam from upcoming exam list. |
| **Quality Requirements:**  3\*- Unanswered questions will be recorded as “null” to the ES system database. |

### Object model



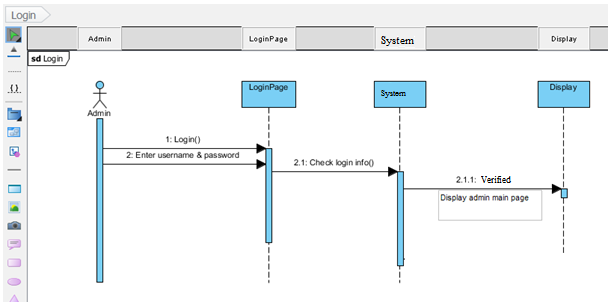
### 

### Use Case Diagram of Online Examination SystemC:\Users\Bilal\Documents\Visual Studio 2012\Projects\Online Exam\online-exam\Online Exam Full Use Case.jpg

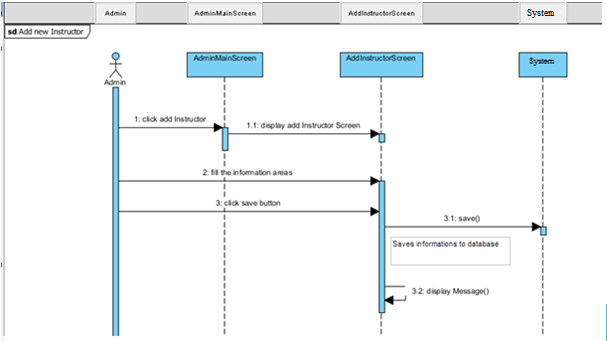
### Dynamic models

**ADMIN DYNAMIC MODEL**

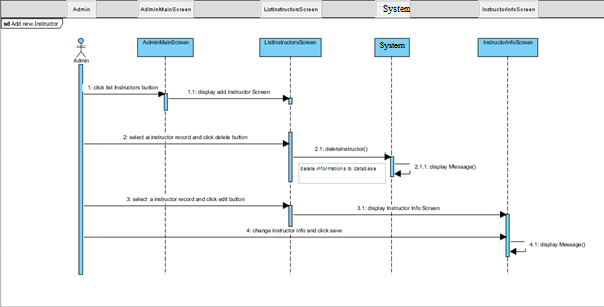
LoginAdmin



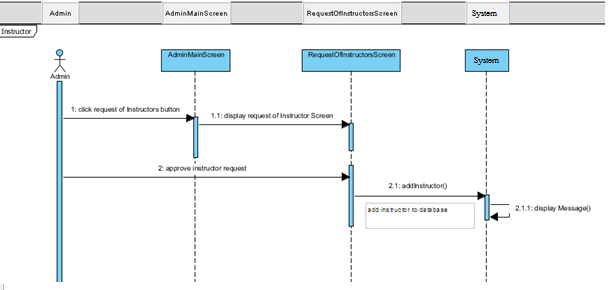
Add new Instructor



List and edit and delete Instructor

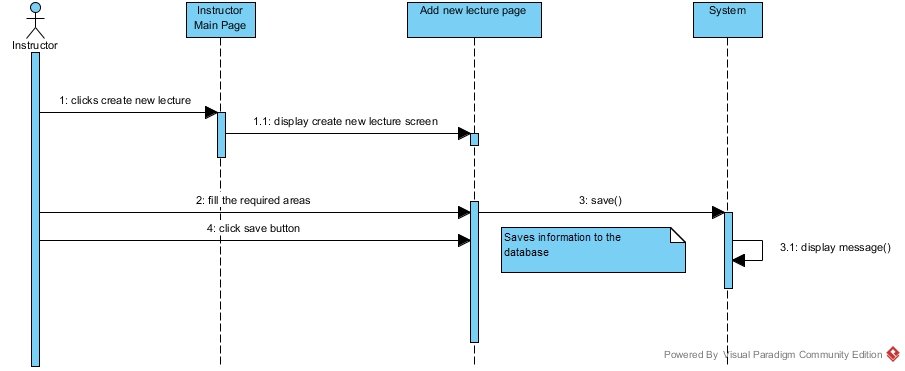


Request of Instructor

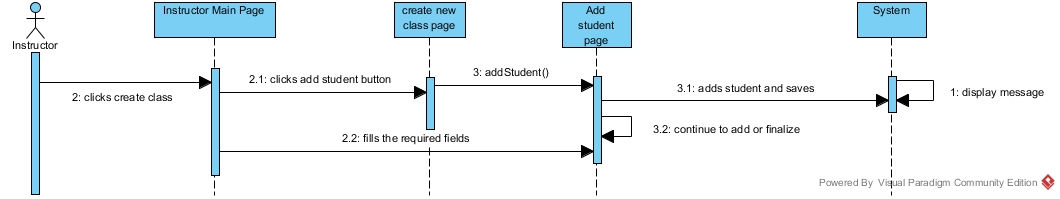


**INSTRUCTOR DYNAMIC MODEL**

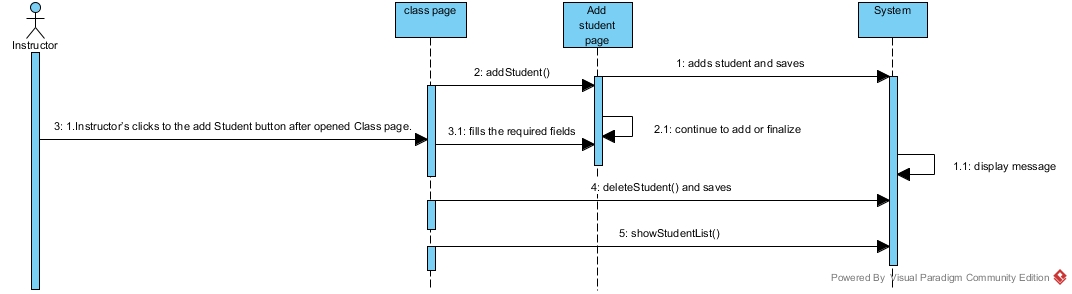
1. **Create New Lecture**
   1. Instructor main page is boundary lifeline.
   2. Add new lecture page is boundary lifeline.
   3. System is controller lifeline.



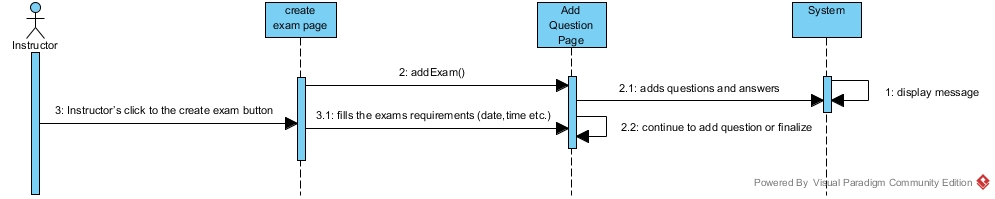
1. **Create Class**
   1. Instructor main page is boundary lifeline.
   2. Create new class page is boundary lifeline.
   3. Add student page is boundary lifeline.
   4. System is controller lifeline.



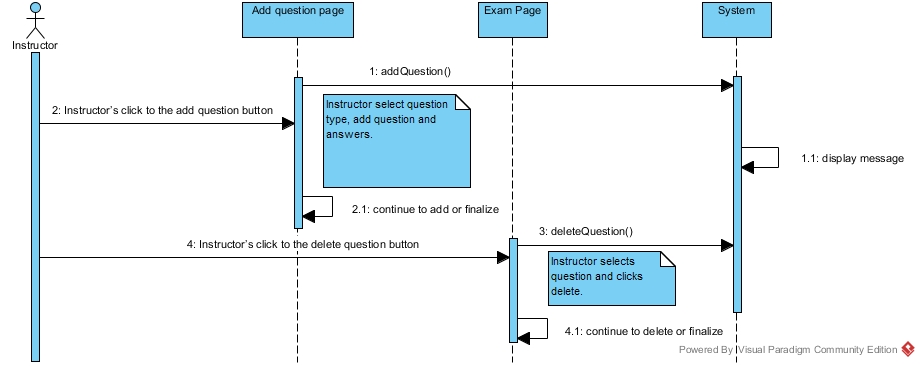
1. **Add, delete and show student**
   1. Class page is boundary lifeline.
   2. Add student page is boundary lifeline.
   3. System is controller lifeline.



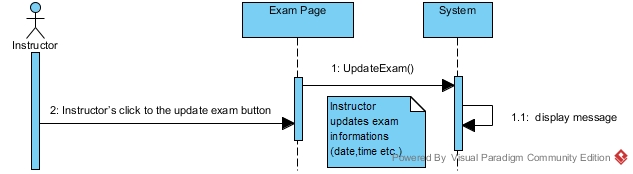
1. **Create Exam**
   1. Create exam page is boundary lifeline.
   2. Add question page is boundary lifeline.
   3. System is controller lifeline.



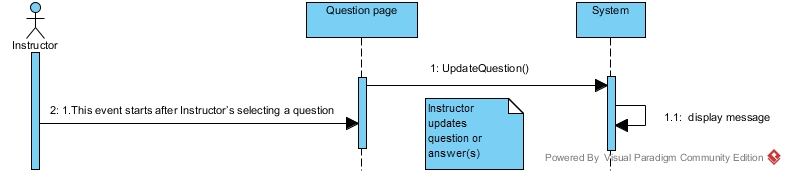
1. **Add and delete question**
   1. Add question page is boundary lifeline.
   2. Exam page is boundary lifeline.
   3. System is controller lifeline.



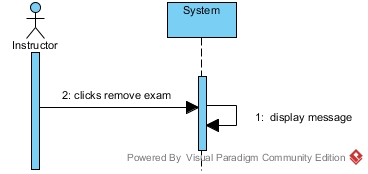
1. **Update Exam**
   1. Exam page is boundary lifeline.
   2. System is controller lifeline.



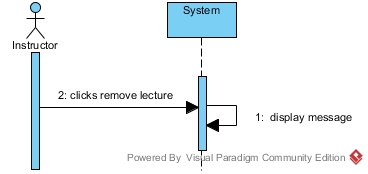
1. **Update Question**
   1. Question page is boundary lifeline.
   2. System is controller lifeline.



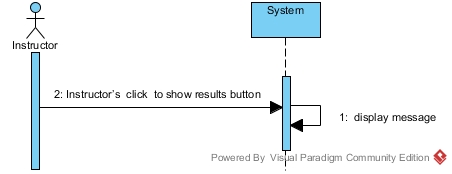
1. **Remove exam**
   1. System is controller lifeline.



1. **Remove lecture**
   1. System is controller lifeline**.**

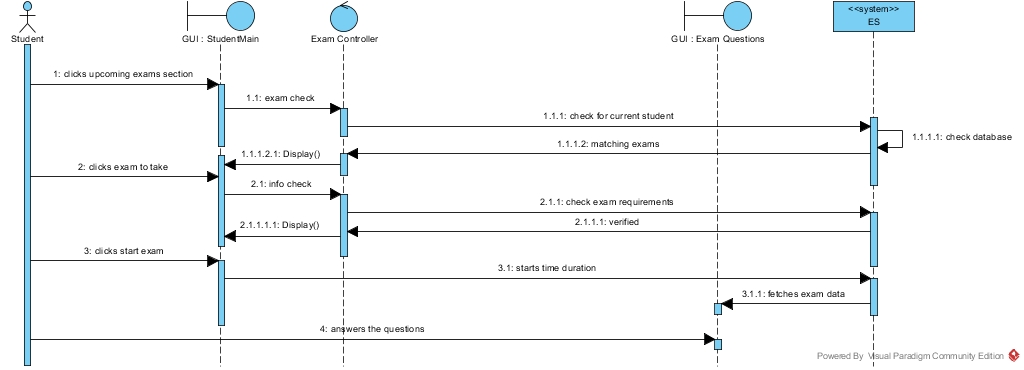


1. **List students results**
   1. System is controller lifeline.

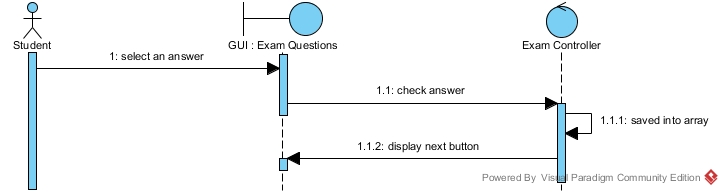


**STUDENT DYNAMIC MODEL**

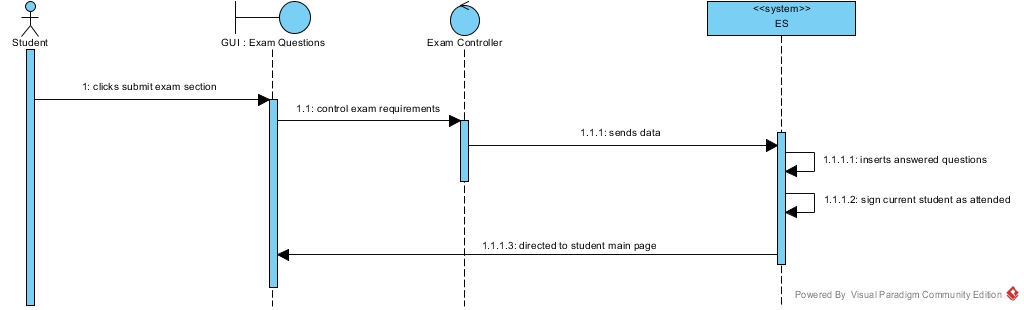
* **Take Exam**

****

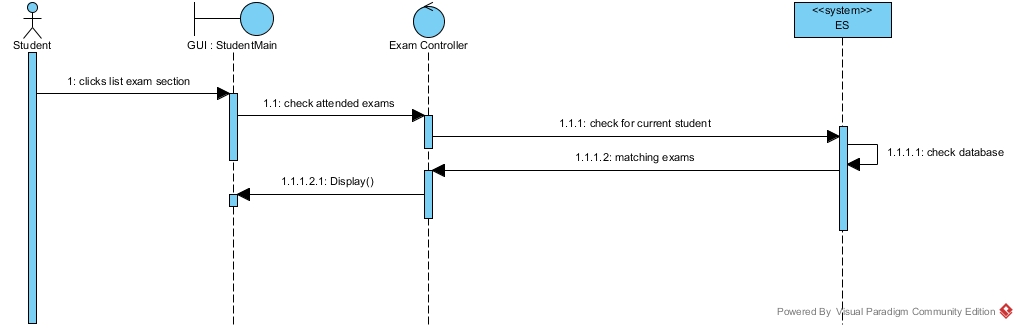
* **Answer Question**

****

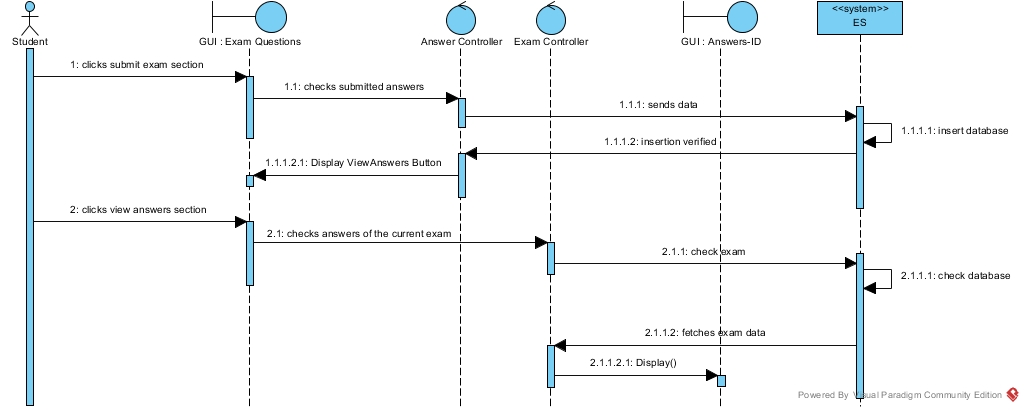
* **Submit Exam**

****

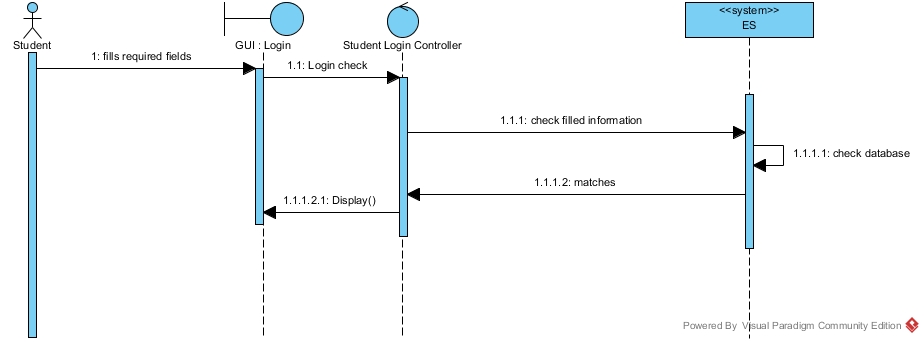
* **List Exam Results**

****

* **View Answers**

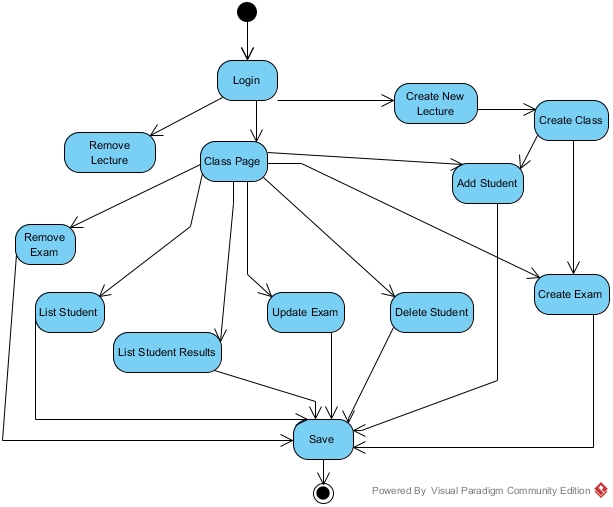
****

* **Authentication**

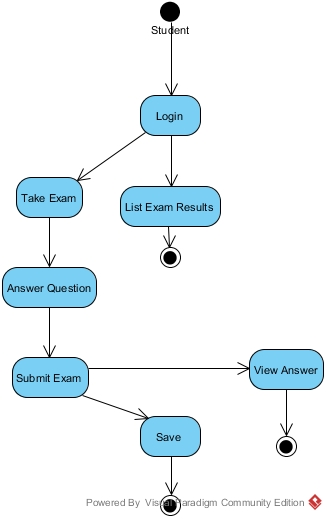
****

**ACTIVITY DIAGRAMS**

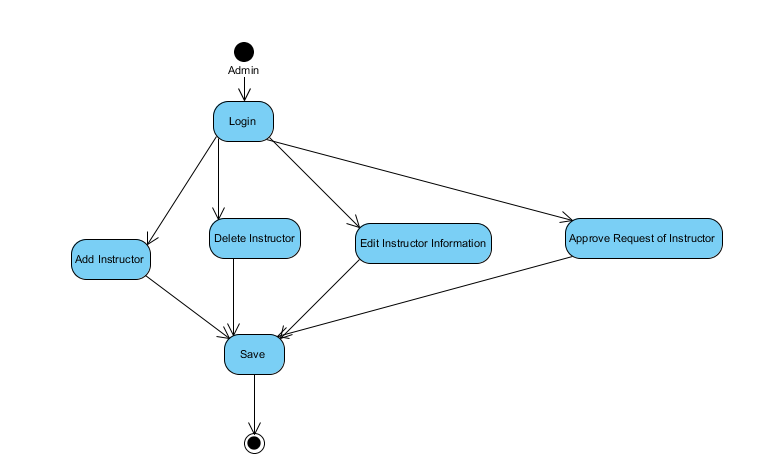
**Activity Diagram for Instructor**



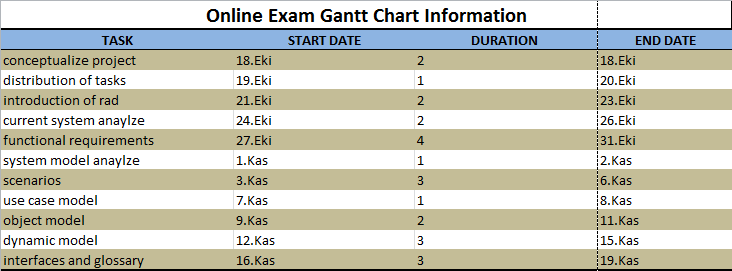
**Activity Diagram of Student**



**Activity Diagram of Admin**

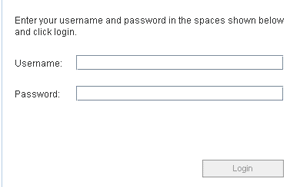


**GANTT CHART OF ONLINE EXAMINATION SYSTEM**

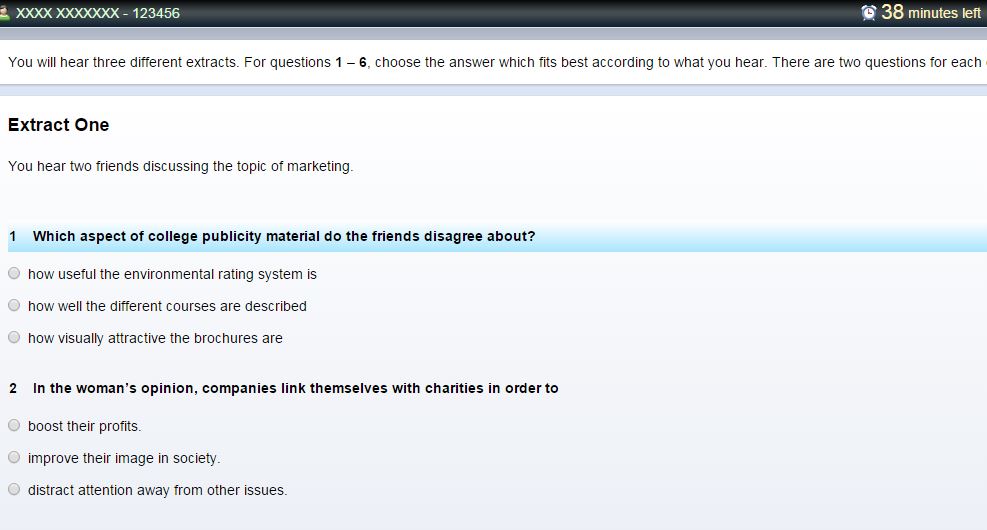
****

### HomepageC:\Users\Bilal\Desktop\home page.jpg

*Login*



*Questions & Answers*



# Glossary

**Actor:** External entity that needs to exchange information with the system. An actor can represent either a user role or another system. In other words; a person, software system, or hardware device that interacts with a system to achieve a useful goal.

**Analysis:** An activity during which developers ensure that the system requirements are correct, complete, consistent, unambiguous, and realistic.

**Authentication:** The process of associating a person with access rights.

**Class diagram:** UML notation representing the structure of the system in terms of objects, classes, attributes, operations, and associations. Class diagrams are used to represent object models during development.

**Functional Requirement:** An area of functionality the system must support. The functional requirements describe the interactions between actors and the system independent of the realization of the system.

**Login:** Procedure used to get access to an operating system, or application, usually in a remote computer.

**Nonfunctional Requirement:** A user visible constraint on the system. Nonfunctional requirements describe user visible aspects of the system that are not directly related with the functionality of the system.

**Class:** A description of a set of objects having common properties and behaviors, which typically correspond to real-world items (persons, places, or things) in the business or problem domain.

**Exception:** A condition that prevents a use case from successfully concluding. The use case’s post conditions are not reached and the actor’s goal is not satisfied.

**Scenario:** A description of a specific interaction between a user and a system to accomplish some goal. An instance of usage of the system. Often presented in the form of a story.

**Sequence Diagram:** An analysis model that shows the order in which messages passes in a system or the chronological sequence of steps that take place in an activity and the entities or classes involved in the activity.

**Unified Modeling Language:** UML is a modeling language used to define a system prior to construction, much like a blueprint is used prior to building a house. It allows the project team to specify, visualize, and document an application, including its structure and design, in a way that meets all of the user requirements.

**Use Case:** A description of an interaction between an actor and a system that results in an outcome that provides value to the actor.

**Use Case Diagram:** An analysis model that identifies the actors who can interact with a system to accomplish valuable goals and the various use cases that each actor will perform.

**Online Examination System**: Students use the system to join exam aminations which is prepared by the instructor.

**Instructor:** Instructors prepare exam in the system and they record students of class on the system. Also they can see students exam results.

**Student:** Students take examination on the System. They can see their exam results.

**Admin:**  Admin give the authority to instructor to create exam. It can delete ,edit and add instructor to the system.

# References

Following website sources :

1. <http://www.guru99.com/learn-software-requirements-analysis-with-case-study.html> (GURU99)
2. <https://www.ibm.com/developerworks/architecture/library/ar-analpat/> (IBM)