Online Exam

System Design Document

v0.1

08.12.2016

Muhammed Bilal Kutlu

Güneycan Demir

Musa Serdar Akmehmet

Çağla Çolak

Prepared for

SE301 Software Engineering



Table of Contents

[1. Introduction 1](#_Toc433996772)

[1.1. Purpose of the System 1](#_Toc433996773)

[1.2. Design Goals 1](#_Toc433996774)

[1.3. Definitions, Acronyms, and Abbreviations 1](#_Toc433996775)

[1.4. References 1](#_Toc433996776)

[2. Current Software Architecture 1](#_Toc433996777)

[3. Proposed Software Architecture 1](#_Toc433996778)

[3.1. Overview 1](#_Toc433996779)

[3.2. System Decomposition 1](#_Toc433996780)

[3.3. Hardware Software Mapping 2](#_Toc433996781)

[3.4. Persistent Data Management 2](#_Toc433996782)

[3.5. Access Control and Security 2](#_Toc433996783)

[3.6. Global Software Control 2](#_Toc433996784)

[3.7. Boundary Conditions 2](#_Toc433996785)

[4. Subsystem Services 2](#_Toc433996786)

[5. References 2](#_Toc433996787)

SYSTEM DESIGN DOCUMENT

# Introduction

Design is the abstraction of an answer; it's the common description of the answer to an issue without a details. Design is view sample seen within the evaluation part to be a sample in a design part. After design part we are able to reduce the time required the implementation.

## Purpose of the System

As said within the Requirements Analysis Doc, the aim of the system is to offer system administrator, faculties, departments and college students with a central location for organizing varied occasions. The aim of this technique supplies the next causes: No physical presence wanted for examinations, No wastage of time throughout analysis, Prompt availability of the outcomes, beginning and completion time are each recorded by the server, A collection of on-line assessments are provided by the software program for the good thing about the scholars. Normally the system is far more preferred as a consequence of these causes across various organizations and the world.

## Design Goals

The design objectives signify the specified qualities of on-line examination system and supply a constant set of standards that should be thought of when making design choices. Based on non-functionality requirements the next design targets must achieved as a way to qualify the system as profitable:

* **Security**

The system security is one of the most important non-functional requirements.

* **Reliability**

The system has to carry out the online examination operations with no errors. The web site developed needs to be extremely dependable and safe in order that details about any questions and many others shouldn’t be leaked earlier than the precise examination is held.

* **End User**

Students can take exam only throughout the previously allotted time slots, however can open website anytime to access different data. System should be able to deal with a number of users. This system should run on a number of operating systems and support windows operating system.

* **Performance**

The system has to be sturdy enough to manage any valid input from the users.

## Definitions, Acronyms, and Abbreviations

**Student**: The system user who will attends to the exams.

**Admin**: The system administrator who will manage all data system data and user controls.

**Instructor**: The user who will prepare and present the questions to the students and calculate their scores by entering the answers of the questions.

## References

**LON-CAPA**

LON-CAPA has allowed my department to build a stable, easily scalable, and highly customizable presence in the world of virtual university courses. By using LON-CAPA's world-leading testing engine and its support for our custom videos, my department has turned our virtual university presence into a state-of-the art teaching and learning environment. Many thousands of students in the USA and around the world have already taken advantage of our course offerings, and our internet presence still keeps expanding.

**- Wolfgang Bauer**, Chairperson, Department of Physics and Astronomy Michigan State University, USA

# Current Software Architecture

According to the system we have shown as a reference, we have seen many limitations to the users and teachers. Teachers were not able to create lessons, so they had to ask their students questions consisting of limited question banks. When teachers asked questions to their students, students had a limited answer. This limit was set at 5, and if the student answered 5 times wrongly, the student's right to answer the question was completely abolished. We have begun to improve our system by addressing these problems.

# Proposed Software Architecture

Online exam system is a web-based online application. This project would be very useful for educational institutes. Our system has a difference of user-friendly interface and reliable, than the others.

Our system will not only have university teachers and students. It is accessible to anyone who wants to make an exam for their students who feel themselves as teachers and have taught themselves with their own efforts. This includes primary school, junior high school and high school. All they have to do is tell us they want to register as a teacher. It should not be forgotten that there is no need to create a permission request to register as a student. Anyone who wants to register can register as a student. Anyone who has registered as a teacher can make exams for their students using two types of questions. According to the old system, we tried to make our system easy to use, open and easy to develop.

## Overview

During the system design modeling of online exam, we divided our system into subsystems. This provides us a strong coherence. Our subsystems are ; Account management system, which has account transactions. Student management system, has functions of student actor, Instructor management system which has instructor’s functions and database subsystem to control storage.

## System Decomposition

The decomposition shows the existence of the following subsystems:

* Account management subsystem
* Student management subsystem
* Instructor management subsystem
* Database subsystem

**Account management subsystem**

This subsystem managing user accounts. It offers perform for creating an account, updating an, approve and close an account for admin side. Admin is the only actor who has permission to access close and approves functions. Create and update functions are accessed by instructor and student. This subsystem uses login services of the student management and instructor management subsystems

**The operations provided by this subsystem are:-**

* Login ()
* Change password ().
* Create account ()
* Update account ()
* Close account ()
* Approve account ()

**Student management subsystem**

This subsystem is managing student actor’s function, offers student side to its functions after authenticate. Managing students access to taking exam, answering questions, seeing their results and displaying answers. Displaying and seeing results functions are useable after submitting exam.

**The operations provided by this subsystem are:-**

* Take exam ()
* Answer questions ().
* Submit exam ()
* List exam result ()
* View answers ()

**Instructor management subsystem**

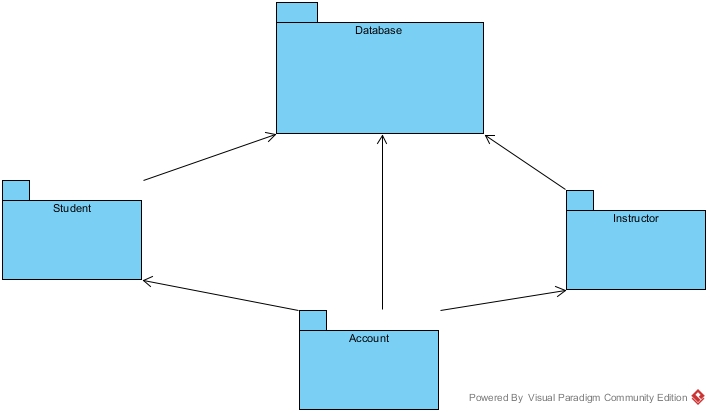
This subsystem is managing instructor actor’s function, offers instructor side to its functions after authenticate. The subsystem managing access of instructor to creating lecture, creating class of the lecture which is connected with database subsystem thus the students which are added class list are available to access exam, add or delete student, also creating and editing exam, adding question, update and delete options.

**The operations provided by this subsystem are:-**

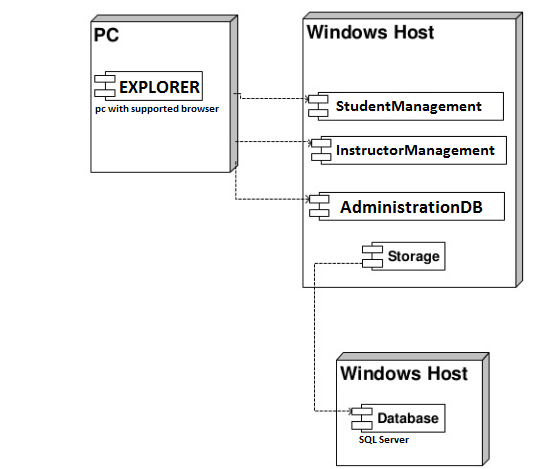
* Create new lecture ()
* Create new class ().
* Add student ()
* Delete student ()
* List class ()
* Create exam ()
* Add question ()
* Delete question()
* Update exam ()
* Remove exam()
* Remove Lecture()

**Database subsystem**

This subsystem will be implemented by relational database management system used to store the persistent data. All subsystems are related and having service with this subsystem

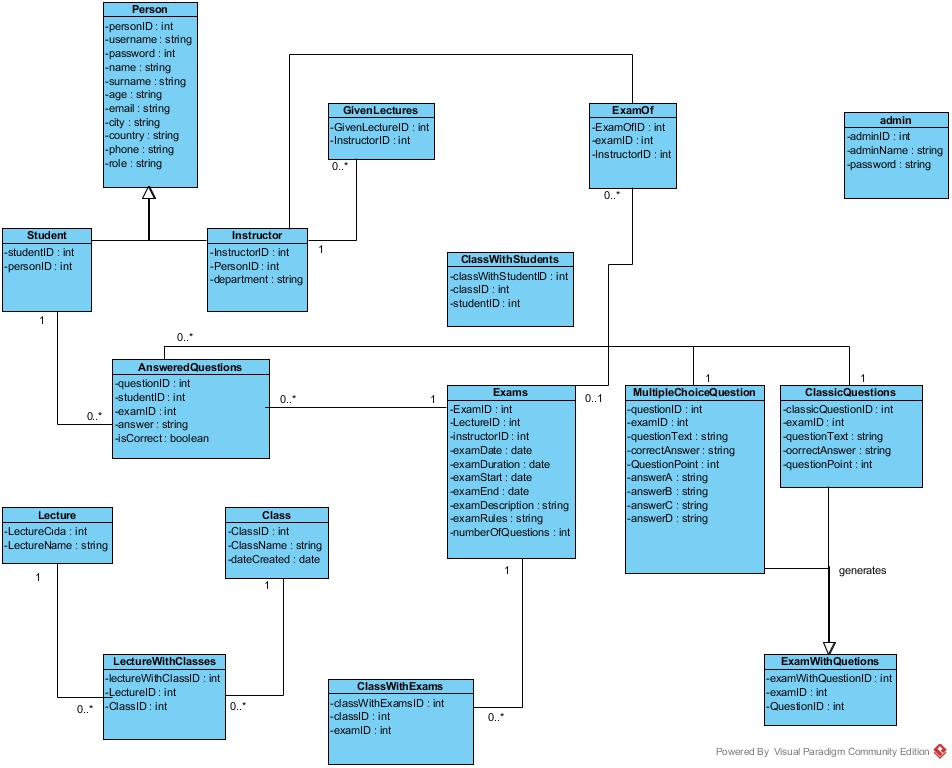


## Hardware Software Mapping



## Persistent Data Management

Our system will use the SQLSerever database engine to store data. This will allow the database to be easily integrated with and accessed by the rest of the system. The database will retain user information for functions such as login for user, show exams that are created by instructors for students to see etc. Our database structure is seen below with entity field’s relations etc.



## Access Control and Security

The application is a multiuser application so it consists of 3 types of users which are instructor, student and administrator. Because of this, the application will provide different interfaces for each user type.

First, the administrator will connect to the system with the membership interface, and will do the administrator's duties on the dating system, such as adding, updating, editing, deleting etc. By the way, registration is not necessary for the administrator through the system website, information will be entered manually into the database at the beginning of the system and the administrator will be the authority that will access the database directly. As a summary, the administrator does not have to register because it is initially registered in the database and the system. The administrator registers the instructor. Each registration process included the instructor registration made by the administrator, will be done with the user interfaces of the system. The system will store all the information in the database and in the login processes again the system will use them by collecting data from the database. The information in the database will use both the confirmation and the use of the system for users. All types of users must log in to the system with their username and password.

After these steps, the system will be ready for the appointment of the instructor. The instructors will register in the application, the system will send the request to admin and admin will approve the request of the instructor, the system will add a new user to the database. The instructor will then connect to the system with your information such as username and password. During the login procedure, the data in the user database table will be obtained and compared to the data entered by the user. Since this operation requires read-only access to the database, it can be performed from different access points simultaneously.

During registration, field filling does not require access to the database, while completion of the process requires the data to be written to the database, which requires read and write access to the database. In that case, the required database fields will be blocked and simultaneous access of multiple users will be denied.

For some situation like updating or deleting information it is necessary to update one of the tables in the database in its phase of completion and therefore must be handled with more care since several users can be the cause of updating the table at the same time. This will also be avoided by blocking.

Finally, viewing the information or lists again requires read-only access to the database. Therefore, multi-user access does not impose problems and new restrictions.

As last words, the usernames and passwords of users will be stored in the user table. No one else accepts the administrator can have access to this information. Authentication interfaces are different for each type of user and will be directed to their own main pages after the login process.

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Actors/  Classes | User | Authentication | Exam | Question | Lecture |
| Admin | <<create>>  createInstructor()  updateInstructor()  deleteInsructor()  updateInfo()  listInstructor()  approveRequest() | Login()  logout() |  |  |  |
| Instructor | <<create>>  createClass()  addStudent()  deleteStudent()  listStudent() | Login()  logout()  resetPwd() | createExam()  updateExam()  removeExam() | addQuestion()  deleteQuestion()  updateQuestion()  removeQuestion() | createLecture() |
| Student | updateInfo()  requestPassRemainder() | Login()  logout()  resetPwd() | joinExam()  showExamResults() | answerQuestion()  viewAnswers() |  |

Figure 3.1 Access Control Matrix for Hospital Appointment System

Admin, User, {<<create>> createInstructor(), updateInstructor(), deleteInsructor(), updateInfo(), listInstructor(), approveRequest()}

Instructor, User, {<<create>> createClass(),addStudent(),deleteStudent(),listStudent()}

Admin, Authentication, {login(), logout()}

Student, Authentication, {login(), logout(), resetPwd()}

Instructor, Authentication, {login(), logout(), resetPwd()}

Instructor, Exam, { createExam(), updateExam(), removeExam()}

Student, Exam, { joinExam(),showExamResults()}

Instructor, Question, { addQuestion(),deleteQuestion(),updateQuestion(),removeQuestion()}

Student, Question, { answerQuestion(),viewAnswers()}

Instructor, Lecture, { createLecture()}

## Global Software Control

External Control Flow (Between Subsystems): ES system defined by the web application with a simple feature. Web server requests request submission of user data. Because the system is multi-user, simultaneous executions can occur. However, the control flow of a single user has a predefined form. After the login step, the system has a web page structure in the form of a tree formed by links or buttons.

Concurrent Control: Because the application is web-based, all subsystems and components can run simultaneously for different users in the application.

Internal Control (Within a Single Process): The process control is carried out by means of the designed forms on the web. The system is based on the page structure of the page request page. This makes the designed procedures simple and mostly linear. However, procedure calls can be made to other subsystems or to the current subsystem. Threads or multiple processes can be required for a process. The system uses a database so that the response time from the database should be minimized.

User Interface: The system user interface will be made through web pages. The control of the next step depends on the user. In addition to this, the flow is implemented within the web page. Most subsystems have a different web page. Due to the system event-driven design, subsystems cannot be considered to have their own event loop. However, events are controlled by web pages.

## Boundary Conditions

Startup: go to system URL and login

Shut Down: click log out and close browser

Error Conditions:

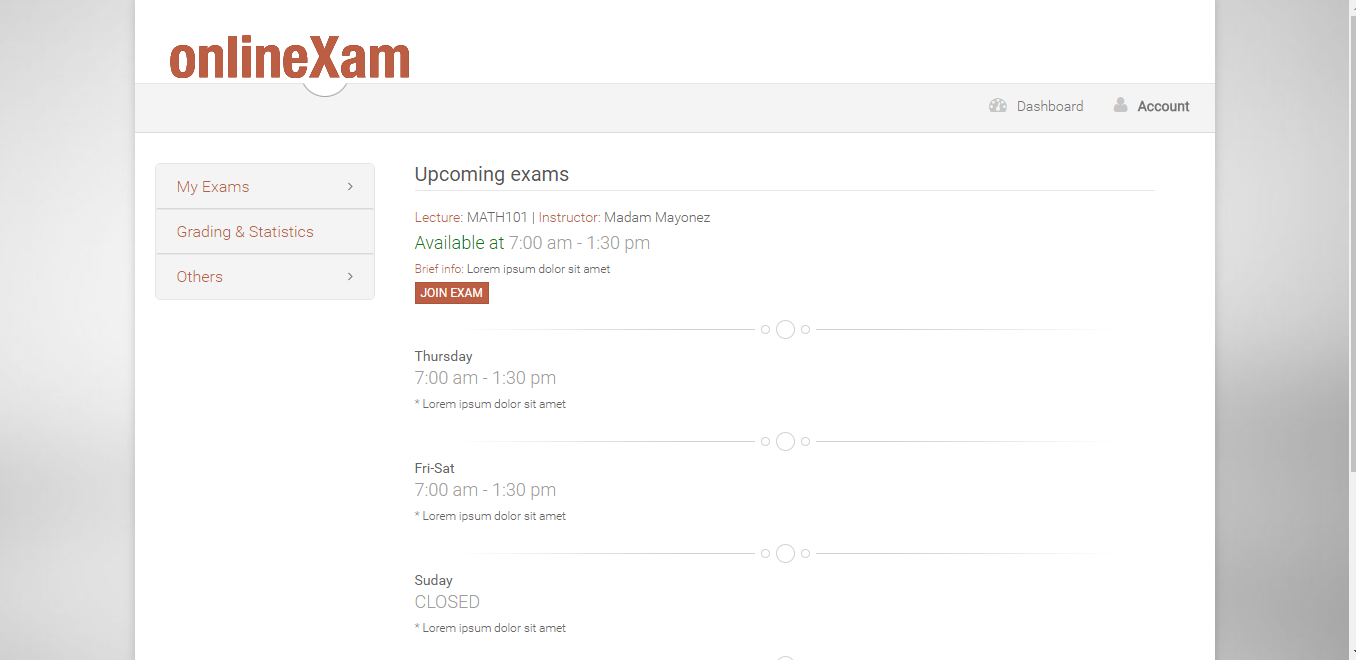
* Logging in:
  + Username or password field cant be blank.
  + Username is not a 5 digit decimal number.
  + Password is not 8 characters long.
  + Password and username don’t match.
  + Username is wrong or does not exist.
  + The welcome screen does not appear after logging in.

* User settings
  + User is unable to change certain settings or changes don’t reflect.
  + Between the time of editing and updating, the system crashes.
* Data Entry
  + The system fails when the dispatcher is entering information.

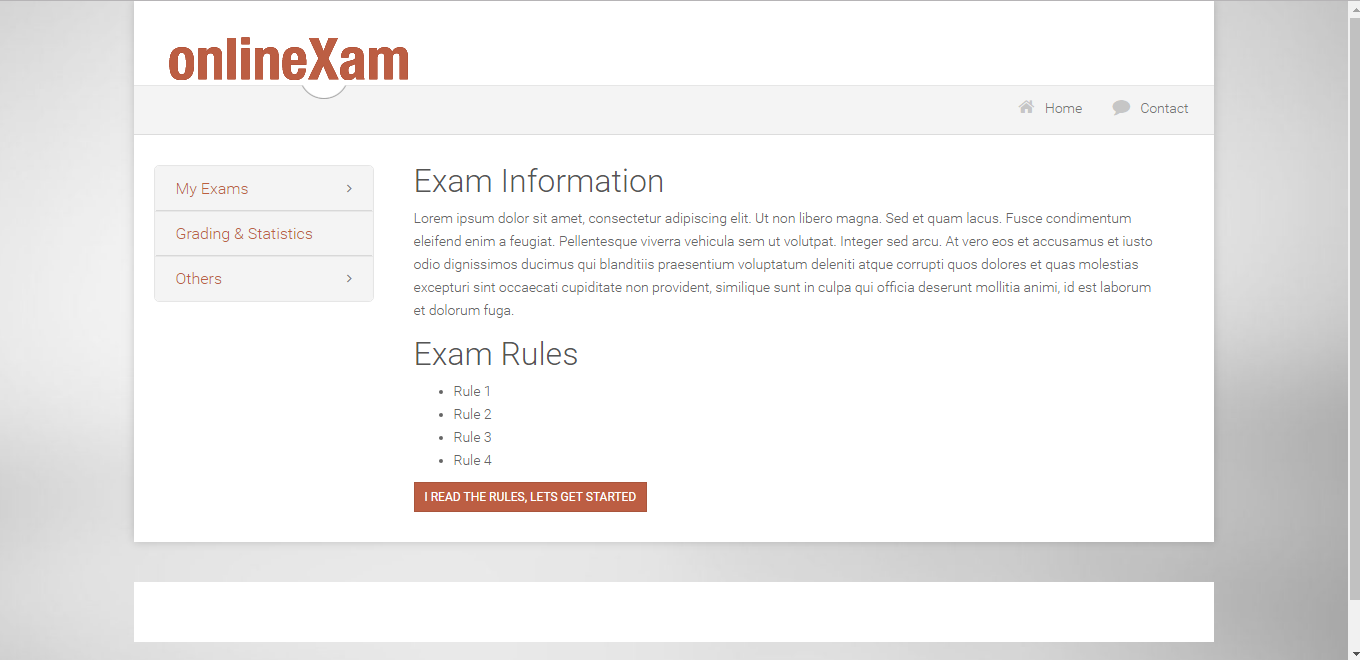
* Instructor Entry
* İnstructor informations cant be exist .
* İnstructor informations cant be blank .
* Exam entry
* Exam name, date, time cant be blank.
* Date time cant be a date before today.
* Question entry
* Question or answer field cant be blank.
* Class List entry
* Same class name is exist.
* Logging out
  + Dispatcher unable to logout.

# Subsystem Services

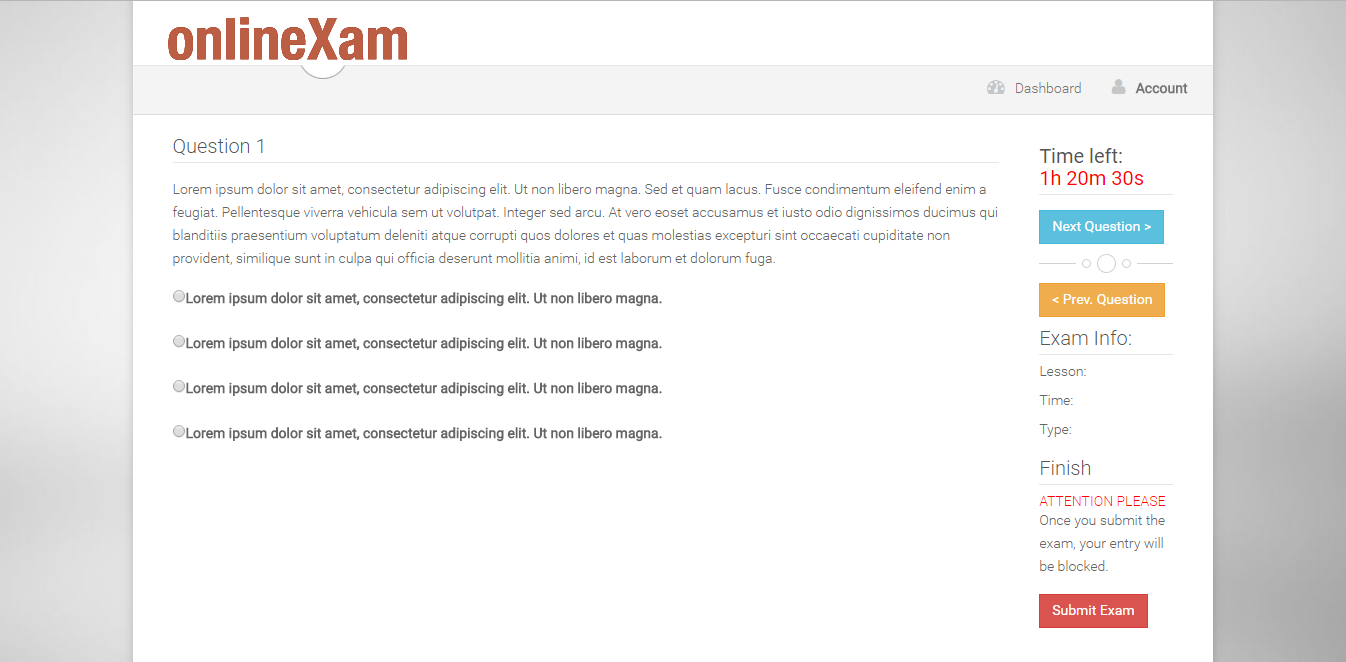
**Student Dashboard**



**Join Exam**



**Answer Question**

****

# References

1. Online homework & exam system, loncapa.isikun.edu.tr/adm/login?domain=bsu