Campus Online++

System Design

Version 1.0

16.11.2015

Can Kavaloğlu

Kadir Tuğberk Arköse

Aslı Kübra Çamlıca

Nazlı Türk

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[1]

# Introduction

## Purpose of the System

Purpose of the system is to provide an effective and online way to students to register and choose their courses easily and to manage their courses and personal information. Also, the other purpose of the system to provide some services to instructors, quota managers and admins. Instructors can manage their offered courses, quota managers can edit and change quotas of courses, and also admins can add courses, students or instructors. Shortly, purpose of the system is to provide efficient way to organize and manage to students, instructors, quota managers and admins about their courses, their offered courses, and personal information and so on.

## Design Goals

Campus Online++ is an online registration and course information platform for students, instructors, quota managers and admins. Thus, it should provide some constraints, functional and non-functional requirements. For example, on students’ perspective, our Campus Online++ design should provide that students can see their schedules, transcripts and quotas of courses and so on by using Campus Online++. In addition, our important goal is to provide to register to students, this is main and important goal of our design. And also, our design should provide that instructors can manage their offered courses and can see their schedules, available rooms and so on by using Campus Online++. Moreover, the other goals of our design Campus Online++ should be run 90% of the time, should allow 1500 parallel users, should be able to maintained easily, should be implemented on Visual Studio platform , and should be web based and so on.

## Definitions, Acronyms, and Abbreviations

SDD- System Design Document.

MVC- Model View Controller

## References

We take current Campus Online system as a reference during our Campus Online++ design.

# Current Software Architecture

Information of the architecture of the system is unknown. Thus, this section cannot provide the architecture information of the current (existing) system.

# Proposed Software Architecture

Our proposed model of Campus Online++ is MVC. Model subsystems maintain domain knowledge, View subsystems display it to the user, and Controller subsystems manage the sequence of interactions with the user. We used MVC model because, in our system, entity objects and data will be in Model, and Controller can be called bridge. Controller provides communication and interaction between Model and View. View can be call interface between users and our system, thus it has boundary objects.

## Overview

In our design, the system has 16 subsystems, User Interface, Quota Manager Interface, Admin Interface, Student Interface, Instructor Interface are the subsystems for the ‘Presentation Layer’. Subsystems for the ‘Business Layer’ are, Login Subsystem, System Management Subsystem, Course Info Subsystem, Instructor Course Subsystem, Registration Subsystem, Update Info Subsystem, Quota Management Subsystem, Messaging Subsystem, Evaluation Subsystem, and Academic Info Subsystem. Finally, subsystem for the ‘Data Layer’ is Data Access.

**User Interface:** provides services for users for common interfaces, it contains Login Form, Message Form, Password Change Form, Display Quotas Option and etc. That is to say, User Interface provides services to display common forms such as log in, password change to all users.

**Instructor Interface:** provides services for instructors for instructors’ interfaces, it contains Free Hours Form, Class List Form, Final Grades Form, Attendance Form and so on. Therefore, Instructor Interface provides services to display forms which are related instructors’ functions such as final grades, offered courses to instructors.

**Student Interface:** provides services for students for students’ interfaces, it contains Curriculum Notice, Schedule Notice, Course Form, Evaluation Form and so on. Therefore, Student Interface; provides services to display forms which are related students’ functions such as evaluation form, course form to students.

**Admin Interface:** provides services to display admins’ forms such as Add Course Form, Add Schedule Form and so on.

**Quota Manager Interface:** provides services to display quota managers’ forms such as Set Quota Form, Course Quota Form and so on.

**Login Subsystem:** provides services for users (student, instructor, and admin) to login.

Update Info Subsystem; provides services for users (student, instructor) to update their information and to change password.

**Academic Info Subsystem:** provides services for students to display their CCR, schedule, and transcript. And also it provides services for instructors to display their schedule and student’s schedule and transcript.

**Evaluation Subsystem:** provides services for students to evaluate their courses and instructors.

**Instructor Course Subsystem:** provides services for instructors to display offered courses, class lists, and free hours of group, attendance lists, and consent requests and to submit final grades.

**Course Info Subsystem:** provides services for users to display quotas of courses and prerequisites of courses.

**Messaging Subsystem:** provides services for users (instructor and student) to send messages. Instructor can send message student and student can send message instructor.

**Registration Subsystem:** provides services for students to register for courses, to send with draw request. And also, it provides services to instructors to display with draw requests, approved programs and to open add drop.

**Quota Management Subsystem:** provides services for quota manager to set quotas of courses.

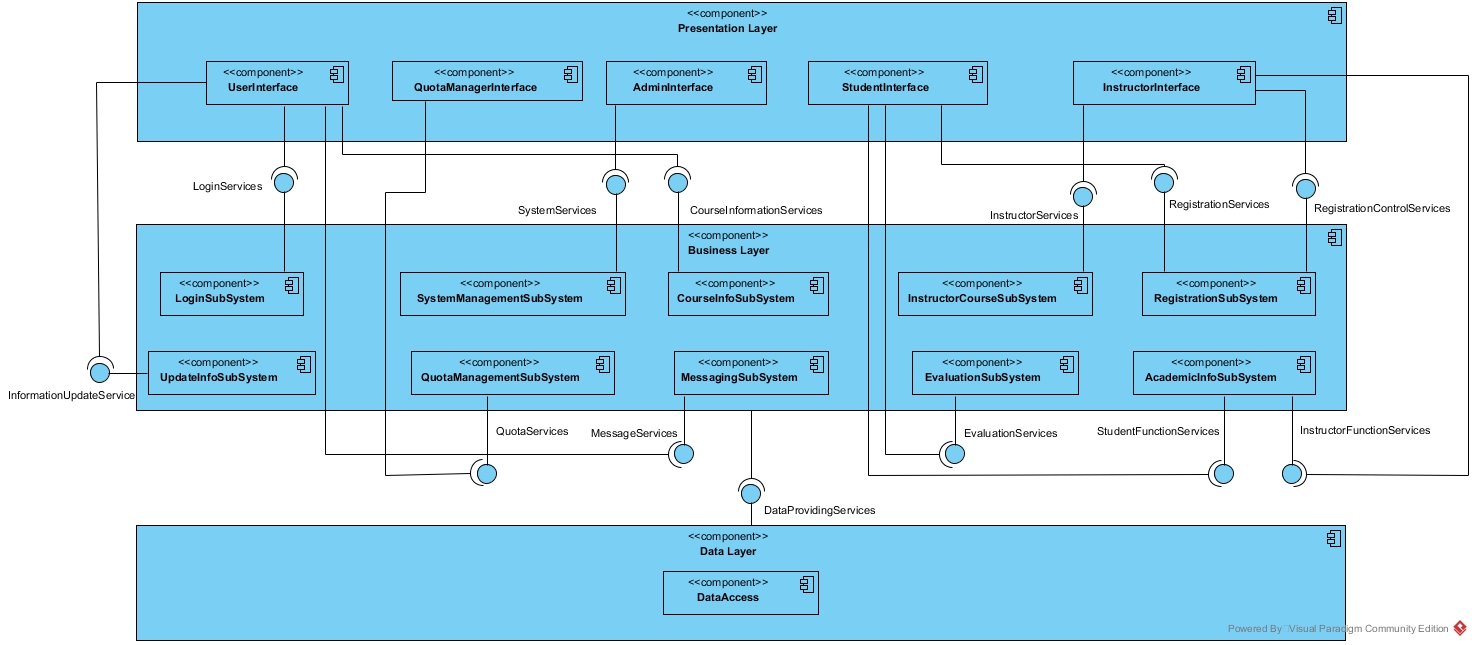
**System Management Subsystem:** provides services for admin to add student, instructor, course, and schedule.

**Update Info Subsystem:** provides services for student, instructor, and admin to update their personal information. In addition, the subsystem also provides the service of changing password for student, instructor and admin as well.

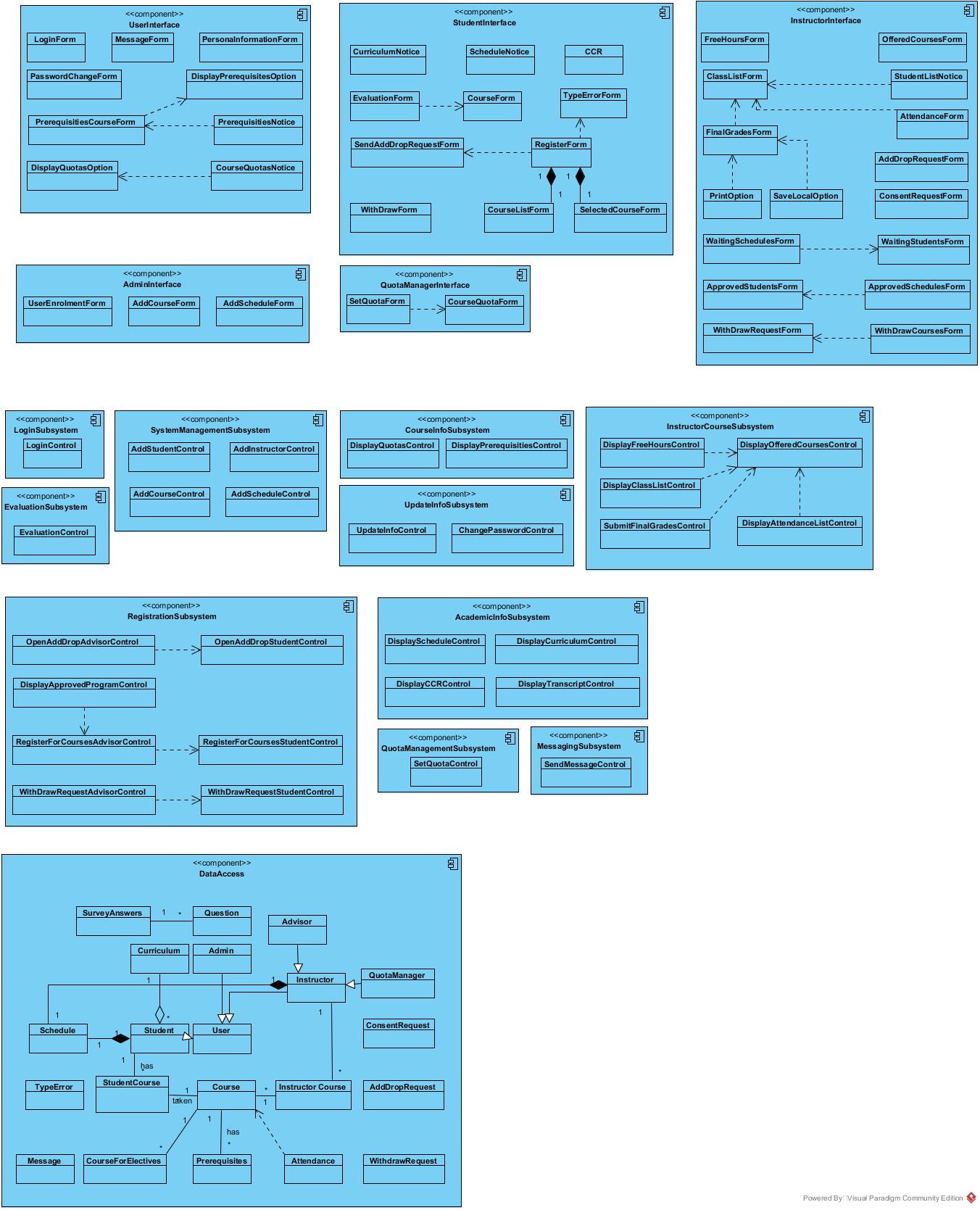
**Data Access Subsystem:** contains all our persistent objects, this part could be called Model of MVC.

## System Decomposition

Coupling view of Subsystem Decomposition:



Cohesion view of Subsystem Decomposition:



Our system decomposition, the subsystems are: User Interface, Student Interface, Instructor Interface, Admin Interface, Quota Manager Interface, Login Subsystem, Update Info Subsystem, Academic Info Subsystem, Evaluation Subsystem, Instructor Course Subsystem, Course Info Subsystem, Messaging Subsystem, Registration Subsystem, Quota Management Subsystem, System Management Subsystem, and Data Access Subsystem. User Interface contains Login Form, Message Form, Password Change Form, Display Quotas Option and etc. That is to say, User Interface provides services to display common forms such as log in, password change to all users. Instructor Interface contains Free Hours Form, Class List Form, Final Grades Form, and Attendance Form and so on. Therefore, Instructor Interface provides services to display forms which are related instructors’ functions such as final grades, offered courses to instructors. Student Interface contains Curriculum Notice, Schedule Notice, Course Form, and Evaluation Form and so on. Therefore, Student Interface provides services to display forms which are related students’ functions such as evaluation form, course form to students. In addition, Admin Interface provides services to display admins’ forms such as Add Course Form, Add Schedule Form and so on. And also, Quota Manager Interface provides services to display quota managers’ forms such as Set Quota Form, Course Quota Form and so on. Moreover, our system decomposition has 10 subsystem. Login Subsystem provides services for users (student, instructor, and admin) to login, it contains Login Control object. Update Info Subsystem provides services for users (student, instructor) to update their information and to change password, it contains Update Info Control object to update and Change Password Control object to change password. Academic Info Subsystem provides services for students to display their CCR, schedule, and transcript. And also it provides services for instructors to display their schedule and student’s schedule and transcript. It contains Display Schedule Control object, Display Curriculum Control, Display CCR Control, Display Transcript Control objects to display all of it. Evaluation Subsystem provides services for students to evaluate their courses and instructors. It contains Evaluation Form Control object to evaluate. Instructor Course Subsystem provides services for instructors to display offered courses, class lists, and free hours of group, attendance lists, and consent requests and to submit final grades, it contains Display Free Hours Control, Display Class List Control, Submit Final Grades Control, Display Attendance Control, Display Offered Courses objects to display all of it. Course Info Subsystem provides services for users to display quotas of courses and prerequisites of courses. Messaging Subsystem provides services for users (instructor and student) to send messages. Instructor can send message student and student can send message instructor. Registration Subsystem provides services for students to register for courses, to send with draw request. And also, it provides services to instructors to display with draw requests, approved programs and to open add drop. Quota Management Subsystem provides services for quota manager to set quotas of courses. System Management Subsystem provides services for admin to add student, instructor, course, and schedule. Data Access Subsystem; contains all our persistent objects, this part could be called Model of MVC.

## Hardware Software Mapping

In Campus Online++ we decide to run three different nodes, one for Data Access layer one for Presentation Layer and one for Business Layer. Data Access layer will run on Microsoft SQL Server based machine. Presentation Layer will be onboard computer and Business Layer will be deployed to a Microsoft Windows Server. Business layer will compile on Microsoft Windows Server, since presentation layer has only Html and JavaScript files it will render in on board computer which are web browser Mozilla Firefox and Google chrome. Since we distributed our system on different nodes, they need to communicate with each other. We did not create a subsystem for communication. We rely on Asp.net MVC for communication. During registration time system will be overloaded, lots of clients will try to access to Campus Online++. Data layer node and business layer node will get huge amount of request. To prevent crashes, we might distribute our system on different servers. We also use some external software libraries and frameworks. For Object-Relational mapping we use Asp.net Entity Framework, for security and authentication we use Microsoft OWIN, for encryption we use simple crypto and for architecture we use Asp.net MVC 5 framework.

## Persistent Data Management

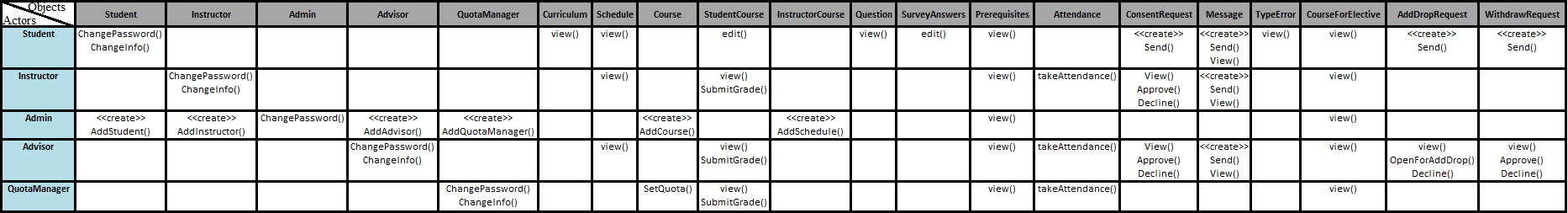
There are some persistent data stored by the system so that the data can outlive a single execution of the system. For example, in Campus Online++, we decide to store persistent data in a database. The persistent data are; Users of the system (Student, Instructor, Admin, Advisor, Quota Manager), Course data which may be offered by different instructor in a different semester. Related to the Course data, we also store Prerequisites data for each Course which has a prerequisite or be a prerequisite for another course. Schedule data of both student and instructor are stored in the system as well. In addition, Question data which are relevant to surveys of course and instructors are stored in the database. Similarly, Survey Answers of the questions are stored in the database as well. There are also Curriculum data for each student according to their departments, and they are stored and retrieved from the database. Lastly, there are such persistent data which are need to be stored in database such as Add Drop Request, With Draw Request, Attendance of the students relevant to the courses they have enrolled, and Message data which are sent by students or instructors to communicate among each other. These are the persistent data that are need to be stored in database so that the data can outlive a single execution of the system. In Campus Online++, because of its advantages and suitability for the Campus Online++ system, we decided to use relational database for the data management infrastructure. Here are some advantages of relational database and why we decided to choose to use it;

1. **Abstraction:** a relational database provides data abstraction at a higher level than flat files.
2. **Ease of use:** The revision of any information as tables consisting of rows and columns is much easier to understand.
3. **Flexibility:** Different tables from which information has to be linked and extracted can be easily manipulated by operators such as project and join to give information in the form in which it is desired.
4. **Precision:** The usage of relational algebra and relational calculus in the manipulation of the relations between the tables ensures that there is no ambiguity, which may otherwise arise in establishing the linkages in a complicated network type database.
5. **Security:** Security control and authorization can also be implemented more easily by moving sensitive attributes in a given table into a separate relation with its own authorization controls. If authorization requirement permits, a particular attribute could be joined back with others to enable full information retrieval.
6. **Data Independence:** Data independence is achieved more easily with normalization structure used in a relational database than in the more complicated tree or network structure.
7. **Data Manipulation Language:** The possibility of responding to query by means of a language based on relational algebra and relational calculus e.g., SQL is easy in the relational database approach. For data organized in other structure the query language either becomes complex or extremely limited in its capabilities.

Looking at all these advantages that relational database provides us, we have come to a consensus that the relational database is the closest data management infrastructure for our system.

## Access Control and Security

The user model of the system in terms of an access matrix:



Campus Online++ suffers from security issues which are very common among the web applications. Those are, threat of an injection to the database, JavaScript injections, exploiting cookies (session hijacking) and DDOS attacks. To prevent database injections we will use prepared statements also we will use stored procedures if it is possible for that case and of course we will try to escape all user supplied input. In case of an injection we will store all our sensitive information encrypted. To prevent JavaScript injections we will obfuscate or hash variable names and values. To prevent session hijacking we will use SSL alongside with HTTPS. Preventing DDOS attacks is a bit tricky, since we won't have a large bandwidth we will use a dos mitigation service to keep our web site up and online. For Authentication we will use Asp.net forms authentication provider, we will keep our users' passwords encrypted and to do that we will use SimpleCrypto library provided by Asp.net. We will store encryption keys elsewhere on the same server.

## Global Software Control

Campus Online++ is thread safety. That is to say, our system Campus Online++ is multi-threaded program. Because, Campus Online++ should provide services for many users at the same time, and also some functions of Campus Online++ should be synchronized. In addition, threads are more intuitive, they currently introduce many problems during debugging and testing. For example, in our design, when student chooses a course in registration time, quotas of courses must be synchronized, because many student could try to choose same course in the same time. Thus, quota of course must be synchronized. Otherwise, when student wants to withdraw from a course by using Campus Online++ and student’s advisor confirms that withdraw request of the student, instructor of the course should know this arrangement and Campus Online++ must be synchronized. That is to say, if student withdraws from the course, instructor of the course should realize this situation, and also the instructor should not any operation of the student anymore. Moreover, if student wants to drop a course in add-drop week, instructor of the course should realize this situation, and also the instructor should not any operation of the student, thus it should be synchronized anymore like withdraw example.

## Boundary Conditions

Campus Online++ is initialized by the system admin invoking the ‘Initialize System’ use case. Once the initialization of the system is completed, the departments and the curriculum information are registered onto system by the system admin. In addition, the system admin initialize the server by invoking the ‘Start Server’ use case. After that, when the system is brought from non-initialized state to steady-state, the system is opened and free for the system users (Students, Instructors, and Admin) to login and perform their tasks. While the users of the system is online and perform their tasks to accomplish, there might be some errors occurring during the transaction processes of the tasks that performed by the users.

The mentioned errors would be originated from attempting to login with invalid credentials, trying to update the personal information (which are necessary to be filled) with empty information, or attempting to select a lab course without selecting its lecture course. All these exceptions are get caught by the system and handled properly. For example, for attempting to login with invalid credentials error, the use case named ‘Invalid Credentials’ is invoked to handle the exception. When the ‘Invalid Credentials’ use case is invoked, the system has already checked the user’s username and password that if the username and password matches with the username and password in the database of the system. The system realizes that they do not match with each other, then in the process of the ‘Invalid Credentials’ use case, the user ,who tries to login, is being informed with a proper message indicates that “Username or Password is Incorrect”. Thus, the system lets user to know that the credentials that the user uses to login in not correct so that the user can realize the situation and try again with the correct one. For trying to update the personal information (which are necessary to be filled) with empty information exception, the exception occurs when the current logged user tries to update his/her personal information but supply empty fields which are not supposed to be empty. Then Campus Online++ performs a checking to make sure that the necessary information fields are given and the information fields are proper to be updated. However, the system realizes that the necessary fields were not supplied, so the use case named ‘Missing Fields’ is invoked to handle the exception. When the ‘Missing Fields’ use case is invoked, a proper message (the message indicates the necessary fields which are not supposed to be empty, but left empty by the user) is display to the logged user to fill the necessary fields to accomplish the update task. For attempting to select a lab course without selecting its lecture course exception, the exception occurs when a student is in the process of his/her registration for courses and tries to submit his/her schedule for approval but did not take the lecture course of one of the lab course(s) that s/he chose. Campus Online++ first checks the mentioned situation if the exception occurs, then if it realizes that there occurred the mentioned exception, the use case named ‘Missing Lecture Course’ is invoked by the system. When the ‘Missing Lecture Course’ use case is invoked, a proper message (the message is composed of the course(s) that s/he forgot to enroll) is displayed to the student to indicate that s/he forgot to enroll the lecture course(s).

When the system is terminated by system admin invoking the ‘Shutdown Server’ use case. All users that currently logged in to the system will be disconnected. This situation occurs for instance, the system admin closes the system for a maintenance before the registration day(s). No user can login to the system until the system is initialized by the admin again.

The use cases mentioned above, namely ‘Start Server’, ‘Shutdown Server’ can be seen below:

|  |
| --- |
| *Use case name:* InitializeSystem |
| *Participant actors:* Initiated by Admin |
| *Flow of events:* 1. Upon successful login, the Admin executes the commands to register the departments and the curriculum information according to the university’s rules. |
| *Entry Condition:* The Admin logs into the database server that the Campus Online++ uses. |
| *Exit Condition:* Initialization of the system is completed and the department and curriculum information of the university are registered onto the database server. |

|  |
| --- |
| *Use case name:* StartServer |
| *Participant actors:* Initiated by Admin |
| *Flow of events:* 1. Upon successful login, the Admin executes the command that is used for the start the server. |
| *Entry Condition:* The Admin logs into the server machine that the Campus Online++ services built on. |
| *Exit Condition:* Campus Online++ services are available and waits for connections from user of the Campus Online++. |

|  |
| --- |
| *Use case name:* ShutdownServer |
| *Participant actors:* Initiated by Admin |
| *Flow of events:* 1. Upon successful login, the Admin executes the command that is used for the shut down the server. |
| *Entry Condition:* The Admin logs into the server machine that the Campus Online++ services built on. |
| *Exit Condition:* Campus Online++ services are unavailable and connections which were already activated are disabled. |

# Subsystem Services

Campus Online++ has 16 subsystems which are User Interface, Quota Manager Interface, Admin Interface, Student Interface, Instructor Interface are the subsystems for the ‘Presentation Layer’. Subsystems for the ‘Business Layer’ are, Login Subsystem, System Management Subsystem, Course Info Subsystem, Instructor Course Subsystem, Registration Subsystem, Update Info Subsystem, Quota Management Subsystem, Messaging Subsystem, Evaluation Subsystem, and Academic Info Subsystem. Finally, subsystem for the ‘Data Layer’ is Data Access.

**User Interface:** provides services for users for common interfaces, it contains Login Form, Message Form, Password Change Form, Display Quotas Option boundary objects and etc.That is to say, User Interface provides services to display common forms such as log in, password change to all users. Log in, password change, display quotas option are same for every user which are student, instructor, quota manager and admin.

**Instructor Interface:** provides services for instructors for instructors’ interfaces, it contains Free Hours Form, Class List Form, Final Grades Form, Attendance Form boundary objects and so on. Therefore, Instructor Interface provides services to display forms which are related instructors’ functions such as final grades, offered courses to instructors. Instructor Interface does not provide other forms which are related student, admin or quota manager, it is only about instructors’ forms.

**Student Interface:** provides services for students for students’ interfaces, it contains Curriculum Notice, Schedule Notice, Course Form, Evaluation Form boundary objects and so on. Therefore, Student Interface; provides services to display forms which are related students’ functions such as evaluation form, course form to students. Student Interface does not provide other forms which are related instructor, admin or quota manager, it is only about instructors’ forms.

**Admin Interface:** provides services to display admins’ forms such as Add Course Form, Add Schedule Form boundary objects and so on. Admin Interface does not provide other forms which are related instructor, student or quota manager, it is only about admins’ forms.

**Quota Manager Interface:** provides services to display quota managers’ forms such as Set Quota Form, Course Quota Form boundary objects and so on. Quota Manager Interface does not provide other forms which are related instructor, admin or student, it is only about admins’ forms.

**Login Subsystem:** provides services for users (student, instructor, and admin) to login.

Update Info Subsystem; provides services for users (student, instructor) to update their information and to change password. It contains Update Info Control and Change Password Control objects.

**Academic Info Subsystem:** provides services for students to display their CCR, schedule, and transcript. And also it provides services for instructors to display their schedule and student’s schedule and transcript. It contains Display Schedule Control, Display CCR Control and Display Transcript Control objects.

**Evaluation Subsystem:** provides services for students to evaluate their courses and instructors. It contains Evaluation Form Control object to evaluate.

**Instructor Course Subsystem:** provides services for instructors to display offered courses, class lists, and free hours of group, attendance lists, and consent requests and to submit final grades.It contains Display Free Hours Control, Display Class List Control, Submit Final Grades Control, Display Attendance Control, Display Offered Courses objects to display all of it.

**Course Info Subsystem:** provides services for users to display quotas of courses and prerequisites of courses.It contains Display Quotas Control and Display Prerequisities Control objects.

**Messaging Subsystem:** provides services for users (instructor and student) to send messages. Instructor can send message student and student can send message instructor. It contains Sen Message Control object.

**Registration Subsystem:** provides services for students to register for courses, to send with draw request. And also, it provides services to instructors to display with draw requests, approved programs and to open add drop. It contains Open Add Drop Control, Open Add Drop Advisor Control, Display Approved Program Control, Register For Courses Advisor Control, Register For Courses Student Control, With Draw Request Advisor Control and With Draw Request Student Control objects.

**Quota Management Subsystem:** provides services for quota manager to set quotas of courses. It contains Set Quota Control object.

**System Management Subsystem:** provides services for admin to add student, instructor, course, and schedule. It contains Add Student Control, Add Instructor Control, Add Course Control and Add Schedule Control objects.

**Update Info Subsystem:** provides services for student, instructor, and admin to update their personal information. In addition, the subsystem also provides the service of changing password for student, instructor and admin as well. It contains Update Info Control and Change Password Control objects.

**Data Access Subsystem:** contains all our persistent objects, this part could be called Model of MVC.

# References

1. Bruegge B. & Dutoit A.H.. (2010). *Object-Oriented Software Engineering Using UML, Patterns, and Java*, Prentice Hall, 3rd ed.
2. <http://maxlogix.blogspot.com.tr/2009/09/advantages-and-disadvantages-of-using.html>
3. Current Campus Online system of Işık University
4. Lecture presentations of the course (the presentations were provided by the Instructor who offered the course).