

# Mobile Content Generator

# A.T.O.M.

1560093 **A**yşenur Durgut

1630987 **T**alha Koruk

1630771 **O**khan Durmaz

1631019 **M**erve Mitik

Detailed  
Design  
Report

# TABLE OF CONTENTS

---

1 INTRODUCTION .....	3
1.1 PROBLEM DEFINITION .....	3
1.2 PURPOSE .....	4
1.3 SCOPE .....	4
1.4 OVERVIEW.....	5
1.5 DEFINITIONS AND ABBREVIATIONS.....	5
1.6 REFERENCES .....	5
2 SYSTEM OVERVIEW .....	6
3 DESIGN CONSIDERATIONS .....	9
3.1 DESIGN ASSUMPTIONS, DEPENDENCIES AND CONSTRAINTS.....	9
3.2 DESIGN GOALS AND GUIDELINES .....	10
4 DATA DESIGN .....	10
4.1 ER DESIGN .....	10
4.2 DATA DICTIONARY.....	12
4.2.1 ANNOUNCEMENT TABLE.....	12
4.2.2 CHAPTER TABLE .....	12
4.2.3 COMMENT TABLE.....	13
4.2.4 COURSE TABLE .....	14
4.2.5 IMAGE TABLE .....	14
4.2.6 PAGE TABLE .....	15
4.2.7 RATE TABLE.....	15
4.2.8 TEMPLATE TABLE.....	16
4.2.9 TEXT TABLE .....	16
4.2.10 USER TABLE.....	17
4.2.11 USER_ROLE TABLE .....	17
4.2.12 VIDEO TABLE .....	18
5 SYSTEM ARCHITECTURE.....	19
5.1 ARCHITECTURAL DESIGN.....	19
5.2 DESCRIPTION OF COMPONENTS.....	19
5.2.1 GUI.....	19
5.2.2 AUTHENTICATION .....	21
5.2.3 USER .....	23
5.2.4 APPLICATIONS .....	24
5.2.6 DATA RETRIEVAL .....	25
5.2.7 DATA STORAGE.....	26
5.2.7 KALTURA.COM .....	27

5.3 DESIGN RATIONALE.....	28
5.4 TRACEABILITY OF REQUIREMENTS .....	28
6 USER INTERFACE DESIGN.....	29
6.1 OVERVIEW OF USER INTERFACE.....	29
6.2 SCREEN IMAGES.....	29
6.2.1 LOGIN .....	29
6.2.2 SIGN UP.....	30
6.2.3 LEARNER HOMEPAGE.....	31
6.2.4 COURSE SCREEN FOR LEARNER .....	31
6.2.5 CHAPTER SCREEN FOR LEARNER.....	32
6.2.6 PAGE SCREEN FOR LEARNER.....	33
6.2.7 INSTRUCTOR HOMEPAGE .....	36
6.2.8 COURSE SCREEN FOR INSTRUCTOR.....	37
6.2.9 CHAPTER SCREEN FOR INSTRUCTOR.....	38
6.2.10 PAGE SCREEN FOR INSTRUCTOR.....	39
6.2.11 PAGE GENERATION.....	40
6.2.12 ADMIN HOMEPAGE .....	42
6.3 SCREEN OBJECTS AND ACTIONS .....	43
7 DETAILED DESIGN.....	44
7.1 USER PACKAGE DETAILED DESIGN .....	44
7.1.1 LEARNER CLASS DETAILED DESIGN .....	45
7.1.2 INSTRUCTOR CLASS DETAILED DESIGN .....	46
7.1.3 ADMIN CLASS DETAILED DESIGN .....	47
7.1.4 AUTHENTICATION .....	48
7.2 GRAPHIC USER INTERFACE.....	51
7.3 DATA MANAGER.....	54
7.3.1 DATA RETRIEVAL .....	55
7.3.2 DATABASE.....	56
8 LIBRARIES AND TOOLS .....	58
9 TIME PLANNING.....	59
9.1 TERM 1 GANTT CHART.....	59
9.2 TERM 2 GANTT CHART.....	60
10 CONCLUSION .....	60

---

# 1 INTRODUCTION

---

This documentation has aim to indicate initial design strategies and structural properties of e-learning project. This document contains system architecture and implementation phases in terms of software components, interfaces, and data. All information about those steps is explained clearly for better code development; thus, programmer can understand and write code through following steps.

---

## 1.1 PROBLEM DEFINITION

---

Technology develops in rapid speed in all over the world. Every invented technology and developed technology comes with also new information. At this point, people have to learn this new information to keep pace with developing technology closely. As a result, people have tried variety education forms. They use school, seminar and special course. However, these are not enough in terms of rapid change of technology. If today conditions are considered, then these results are emerged. Firstly, time is important for people but transportation is also loss of time. Secondly, there is one person who knows this course perfectly but even bigger class has 200 people. This number seems bigger; however, it is known that there are more than one billion people all over the world. This professional person is not enough to teach everybody who wants to learn something about this field. Thirdly, money becomes problem. People spend lots of money on education. They are major problems as the best known. At this point, e-learning become as a major solution for companies, employees, teacher, students and other such systems which has educational relationship. On the other hand, e-learning is important for companies in terms of improving training cost, decreasing material cost and increasing productivity. Also it is important for learner in terms of accessing real-time, having freedom to fail, improving retention, providing personalized learning.

It is known that while e-learning become major education tool, there have still some problems. Actually, there are lots of problems but these three are considered as majors.

- **Privacy** is the one of major problem of e-learning education. Since, information is really valuable and it should be protected when it is necessary. For example, any company does not want to share structural information with anybody except people who work for this company. Even there are strict boundaries inside some company and there is no share between its departments.
- **Compatibility** is the second major problem of e-learning education. Since, there are lots of different mobile devices, operating systems, platforms and even programming languages. As a result, there emerge compatibility problem. In other words, one program works on Windows but not Linux. Some websites are shown Firefox wonderfully but another browser Safari makes it nasty.
- **Simplicity** is the third major problem of e-learning education. Since, instructor has to create and design website or program. This is not easy for person who is master of any course but uses computer only to surfing and reading e-mail.

There can be developed better e-learning content generator tools which does not have these three major problems such as incompatibility, difficulty and protection of information. In order to overcome these problems of e-learning, we have been developing mobile content generator tool. By designing such a tool, people can create content of e-learning program easily through simple interface. This program works on any web browser because this program has been developing with using new technologies. Also certain people can receive this education as instructor desired so that privacy is provided. On the other hand, people who receive this education also can use this program easily, on any web browser and in secure. Education can be a video, a text or mixed of them. What's more, some features of tool gives opportunity to instructor create some feedback area such as comment, rating etc.

---

## 1.2 PURPOSE

---

This document is the software design description report of the project Mobile Content Generator. The purpose of this document is to explain briefly data design and system architecture of MCG. Moreover, this document explains all functionality clearly. Target group of this document is developers and testers. Any developer can use this detailed design report to develop this program better.

---

## 1.3 SCOPE

---

This document covers design report as in more detailed. This document contains more detailed modules which compose our system's main part. These modules are decomposed small modules and they have explained. This report contains system overview, design consideration, data design, system architecture, user interfaces, detailed design, tools, libraries and time planning. This report indicates general description of the overall system and its design. Design consideration part includes approximately system performance as dynamically and statically. Also design consideration indicates our goals and guidelines. Data design part contains description of data which explains that how world information transforms to data structure of system. Also data description part includes how data and system entities are stored and where they are stored such as database. This report contains modules about our architecture which we have developed. System architecture includes explanation of the relationship between the modules, explanation of components roughly and how to manage complete functionality of our system. This document indicates functionality of our system from user's perspective through user interfaces. User interface part shows and explains clearly interfaces which are prepared for interactive user. Detailed design part contains the internal details of each design component. Each design has classification, definition, responsibilities, constraints, composition, uses, interactions, resources, processing, interface and necessity diagrams such as sequence diagram. Libraries and tools part contains information about which languages and programs are used to develop this software system. Also languages and programs contain explanation. Time planning part includes process of software development in terms of Gantt chart.

---

## 1.4 OVERVIEW

---

This document contains six additional chapters. In the next chapter we will give a general overview of the system. Then the Design Considerations follows in which assumptions, dependencies, constraints and guidelines of our system will be explained. The chapter after that is the Data Design which will include information and data domain of the system and its organization. In the fifth chapter, we will architecture of our system in detail with description of components. Then we will present images for our user interface. After, libraries and tools will be specified. Gantt charts of term1 and term2 will be presented and finish with conclusion.

---

## 1.5 DEFINITIONS AND ABBREVIATIONS

---

**E-Learning:** The delivery of a learning, training or education program by electronic means. E-learning involves the use of a computer or electronic device (e.g. a mobile phone) in some way to provide training, educational or learning material [2].

**Kaltura:** An Open Source Video Platform

**Learner:** The user authorized to reach the learning content but not authorized to change it.

**Instructor:** The user who can generate learning material and authorize e-learners to it.

**MCG:** Mobile Content Generator

**SDD:** Software Design Description

---

## 1.6 REFERENCES

---

[1] *Software Engineering Standards Committee of the IEEE Computer Society*, IEEE Std 830-1998

[2] D. Stockley. (2011, Jan. 19). *What is E-Learning?* [Online]. Available: <http://derekstockley.com.au/elearning-definition.html>

[3] A. H. Doğru, "Software Engineering Concepts," in *Component Oriented Software Engineering*, 1<sup>st</sup> ed. Dallas, TheAtlas Publishing, 2006.

[4] Scacchi, W. "Process Models in Software Engineering", Irvine: University of California, 2001.

[5] <http://www.articulate.com/rapid-elearning/why-e-learning-is-so-effective>

[6] [http://en.wikipedia.org/wiki/Cascading\\_Style\\_Sheets](http://en.wikipedia.org/wiki/Cascading_Style_Sheets)

[7] [http://en.wikipedia.org/wiki/Microsoft\\_SQL\\_Server](http://en.wikipedia.org/wiki/Microsoft_SQL_Server)

[8] [http://en.wikipedia.org/wiki/Microsoft\\_Visual\\_Studio](http://en.wikipedia.org/wiki/Microsoft_Visual_Studio)

[9] <http://en.wikipedia.org/wiki/ASP.NET>

[10] <http://en.wikipedia.org/wiki/HTML5>

[11] <http://en.wikipedia.org/wiki/JavaScript>

[12] <http://www.rosettastone.eu>

[13] Software Requirements Specification for Mobile Content Generator

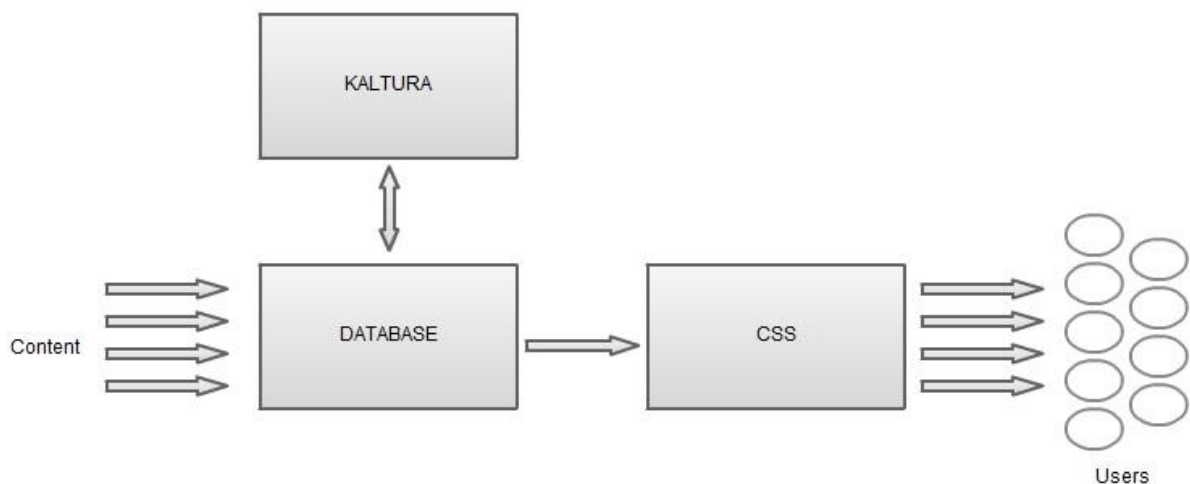
The SRS document for Mobile Content Generator, prepared according to the IEEE Std 830-1998: IEEE Recommended Practice for Software Requirements Specifications

---

## 2 SYSTEM OVERVIEW

---

Mobile Content Generator is basically an e-learning system with instructors who are adding courses and learners who are accessing these courses' contents. Instructors can add courses only by using desktop computers however; learners can access these courses via desktop computers or mobile devices like tablets and mobile phones.

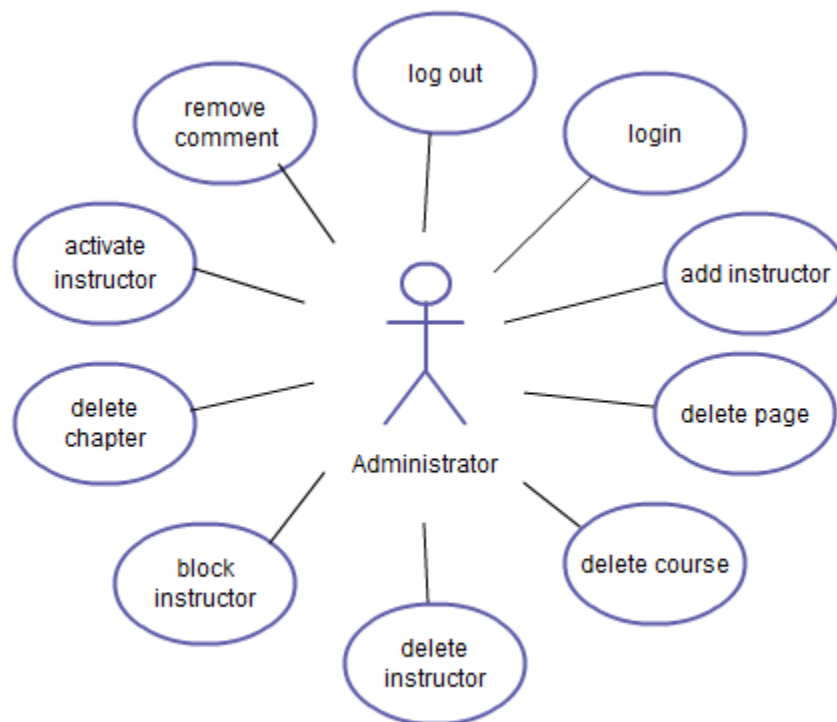


**Fig1. Overall System Block Diagram**

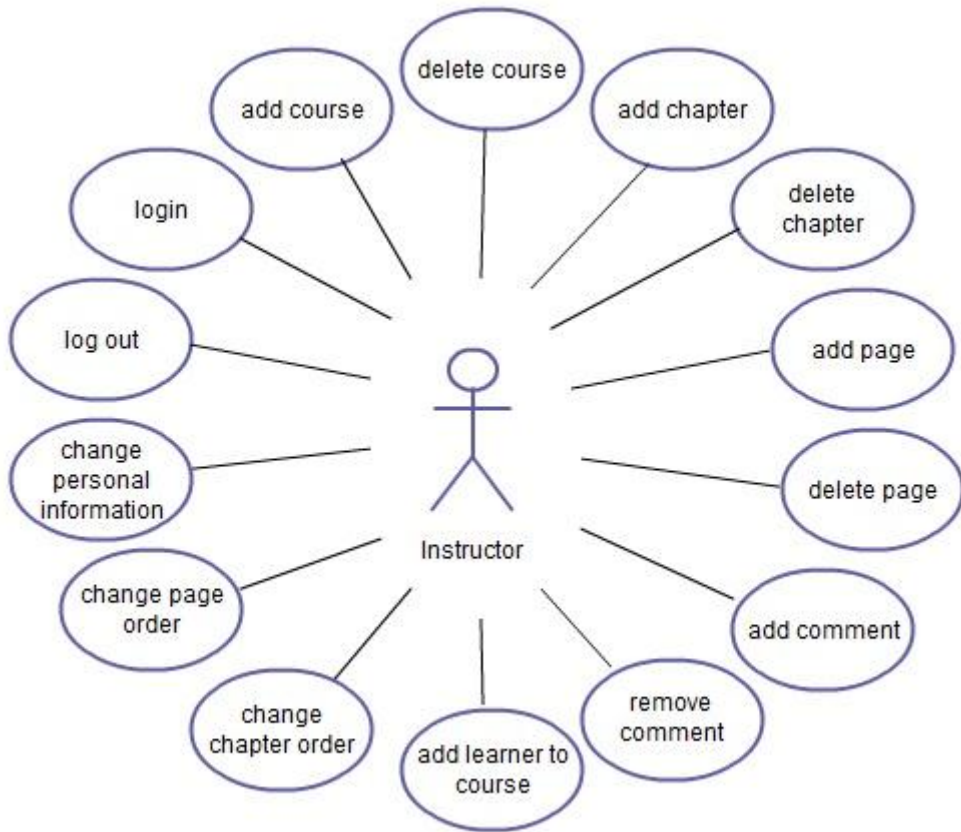
Fig.1 is the overall diagram of the system. The contents which are added to the system by instructors are sent to the database and then to Kaltura's servers for further processing. Contents coming back to database from Kaltura's servers are designed with CSS and served to users.

Our system has 3 different types of users as administrator, instructor and learner. With the administrator rights the user is able to add, delete, block and activate instructors, delete

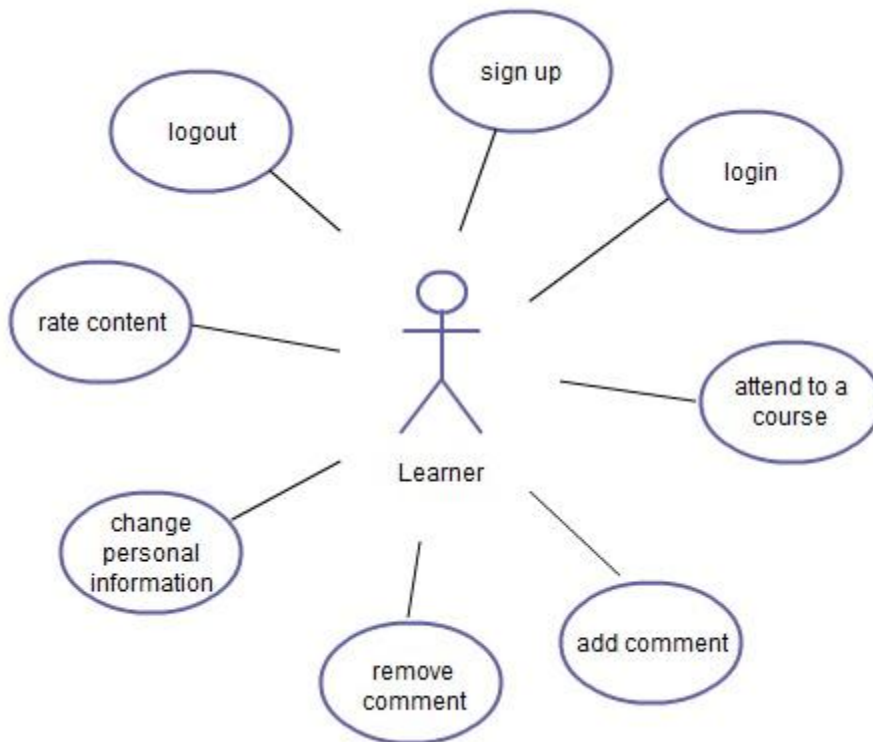
course, comment and page. The user who has instructor rights can add/delete course, chapter, page, comment, assign learners to courses and change the orders of chapters/pages. The last type of user is learners and they have the rights of attending to a course, adding/deleting comments and rating contents. Basically, administrators are able to change database in terms of contents and users of the system, instructors are able to change database in terms of contents and finally learners are able to reach to contents. The main functions of these users shown by use case diagrams in Fig.2, Fig.3 and Fig.4.



**Fig2. Administrator Use Case**



**Fig3. Instructor Use Case**



**Fig4. Learner Use case**

---

## 3 DESIGN CONSIDERATIONS

---

---

### 3.1 DESIGN ASSUMPTIONS, DEPENDENCIES AND CONSTRAINTS

---

In the beginning, this application is web application. Therefore, there are some restrictions as a result of server computer, network connection and developers. Server computer should be fast and provide to open more sessions simultaneously. Network connection should be better to acquire information from server faster. Files' size and number also affect the performance. Then they should be minimal. There is also another performance problem which is database. There are lots of query for database. Better optimization means better database works. Externally, "kaltura.com" has also performance effect on working process. This performance effect cannot be optimized or changed. There are two types of performance effects such as static performance and dynamic performance.

- **Static Performance**

Some of features depend on server performance. For example; number of computer which are connected to website, number of people can connect at the same time, called capacity of system features. Assuming that server has average performance, number of computer which connected to website will be 4000-5000. Making simultaneously connection number will be 90-100. Called capacity of system will be related RAM space and size of code. This size of code is directly related dynamic growing and developer's optimization. Therefore, assuming optimized code, approximately 4500 people each for a computer can call features unlimited.

- **Dynamic Performance**

Response time depends on code efficient, server performance and number of people who make operation on server. This response time will be less than one second when number of people who connect server is less than %60 of limit number. However, there will be another performance effect such as "kaltura.com". Website will acquire also video if this page contains any video. This makes response time between 2-3 second depending on "kaltura.com" response time. Since, this website prepares desired video as desired size and quality. There is also database query performance. Optimized database and fast server will make progress less than 0.01 second for a query for %70 of operations. If there are lots of operations on multiple tables, then this time will increase maximal under 0.1 second.

- **Design Constraints**

This web application contains programming languages such as ASP.NET, CSS, HTML5, MSSQL, JAVA and JAVASCRIPT. Also there is framework which name is VISUAL STUDIO. There will no hardware constraints. This web application is cross platform. Any hardware, which can run HTML5, runs also this application. Therefore, mobile phones, desktop computers and tablet computers can show this application.

System will be realizable. Since, this system will just work on server and show on HTML5. System is portable. Since, this is web application and it will work on any servers. Also, it can be transported to other server easily. Security is also important. There is login screen. There is no other entrance way. Sign up form includes image verification to protect from spam. Instruction is created directly admin and inappropriate videos directly delete from admin. Server part is also secure by itself. There are user level permissions that they just follow the course content.

---

## 3.2 DESIGN GOALS AND GUIDELINES

---

MCG works on web services; therefore, it depends much more on database. Security is important for database. In order to make secure system, database must have better design. Nobody can reach database without permission. Each profile must have own area in database. Queries must be designed considering the privacy. Queries' limited are determined clearly. As a result, only authorized people reach their area and this makes a shield for foreign attacks.

MCG must be designed to work every web browser without any problem. In this manner, learner can receive education anywhere. MCG provides portability in terms of working on every web browser.

Speed must be emphasized on because device's scope of MCG also includes mobile devices. This type of devices has slower processor according to standard computer.

KISS principle must be used for its interfaces. As indicated, this tool must provide simplicity. Since, instructor and learner have different technological and educational background. So that, there must exist simple interface for creating content or receiving education. On the other hand, MCG must be designed different interfaces for different devices. Since, mobile devices screen are relatively small, they need simple interfaces. Other computer has large screen and there can be more button or text area.

---

## 4 DATA DESIGN

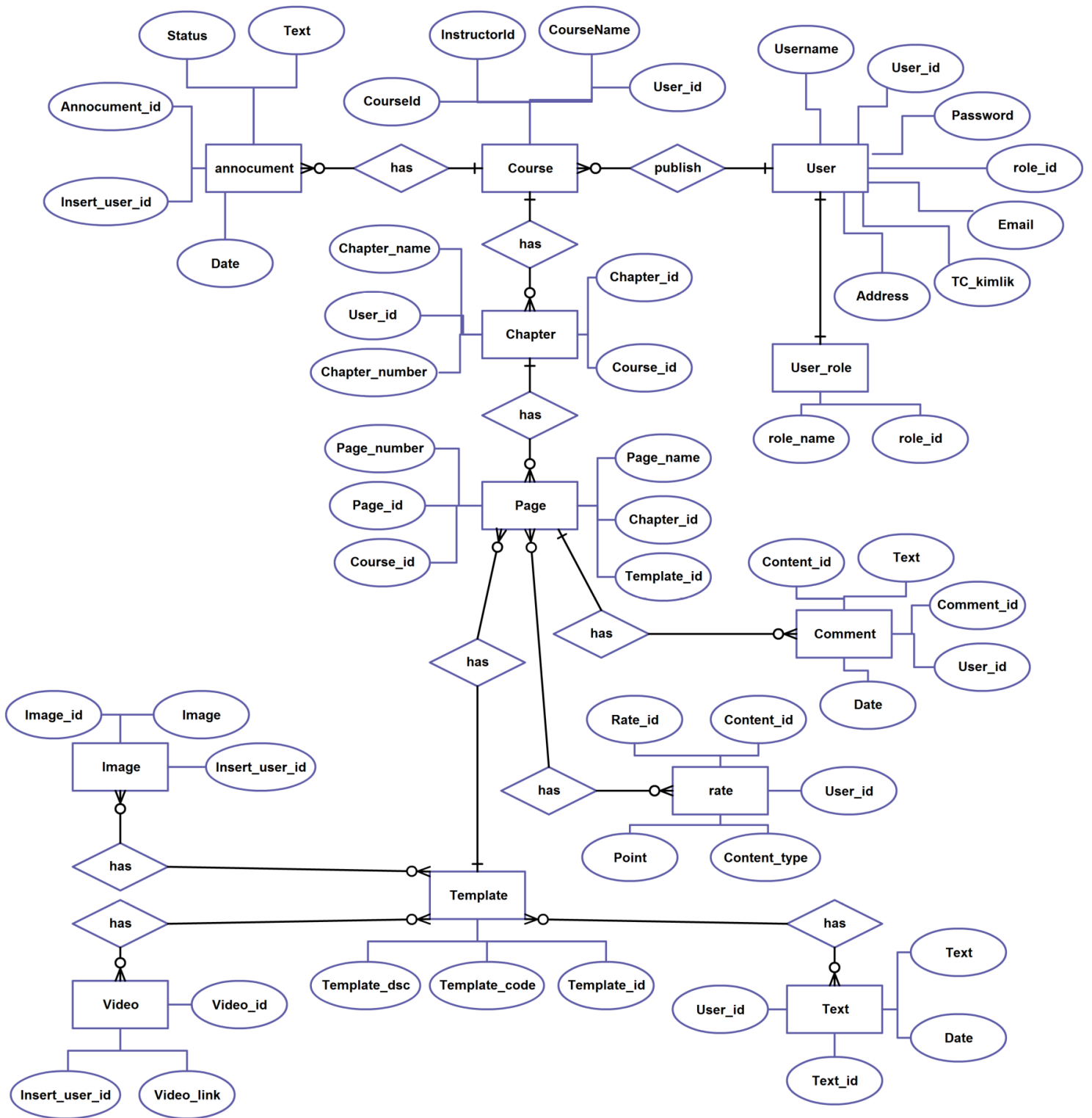
---

---

### 4.1 ER DESIGN

---

Mobile Content Generator will run on application server. It will store all information and data in relational database. All data will be stored in database tables and used by using appropriate queries. All data manipulations will be done by database management system. MS-SQL will be used to handle database operations. ER Diagram is shown in the Fig.5.



**Fig5. ER Diagram**

---

## 4.2 DATA DICTIONARY

---

To store data, we designed following tables. We will store user information and all contents except videos in the database. Videos will be stored in www.kaltura.com web page. We will only store the code that includes video's ID.

---

### 4.2.1 ANNOUNCEMENT TABLE

---

Field	Type	Nullable	Foreign Key	References
<u>announcement_id</u> (P.K.)	integer(10)	No	No	-
insert_user_id	integer(10)	No	Yes	user_id(User)
text	varchar(500)	No	No	-
date	date	No	No	-
status	integer(1)	No	No	-

**Table 1. Announcement Table**

In Table 1, we store announcements that will be created by system admin or instructor. There is a primary key announcement\_id and a foreign key that references User table's user\_id.

announcement\_id: Stores a unique number for every announcement.

insert\_user\_id: Stores the id of the user that inserts the announcement record to the system.

text: Stores the text of the announcement.

date: Stores the insertion date of the announcement to the system.

status: Stores a number to indicate status of announcement (i.e. active, passive, etc.)

---

### 4.2.2 CHAPTER TABLE

---

Field	Type	Nullable	Foreign Key	References
<u>chapter_id</u> (P.K.)	integer(10)	No	No	-
chapter_name	varchar(250)	No	No	-
course_id	integer(10)	No	Yes	course_id(Course)
chapter_number	integer(3)	Yes	No	-
user_id	integer(10)	No	Yes	user_id(User)

**Table 2. Chapter Table**

In Mobile Content Generator, a course will consist of chapters. To handle this need, we store chapters in Chapter table shown in Table 2. Chapter table(Table 2) keeps basic attributes of a

chapter and its primary key is chapter\_id. There are two foreign keys, first one is course\_id that refers to Course table(Table 4) and second one is user\_id that refers to User table(Table 10).

chapter\_id: Stores a unique number for every data in table. That is also used for primary key.

chapter\_name: Stores name of the stored chapter in database and system.

course\_id: Stores the id number of the course in which the chapter placed in.

chapter\_number: Stores the number of chapter in the course.

user\_id: Stores the id number of the user that inserts the chapter record to the system.

---

### 4.2.3 COMMENT TABLE

---

Field	Type	Nullable	Foreign Key	References
comment_id(P.K.)	integer(10)	No	No	-
text	varchar(1000)	No	No	-
content_id	integer(10)	No	No	-
content_type	integer(3)	No	No	-
date	date	Yes	No	-
user_id	integer(10)	No	Yes	user_id(User)

**Table 3. Comment Table**

In Mobile Content Generator, user will be able to comment contents of a course. Content word means course, chapter or page. Actually, content\_id refers to these tables (Table 2, Table 4 and Table 6). To keep contents type, there is content\_type attribute. There is one foreign key, user\_id that refers User table(Table 10) and keeps information of the user who adds that comment to the system. Primary key of the table(Table 3) is comment\_id.

comment\_id: Stores a unique number for every data in table. That is also used for primary key.

text: Stores the comment in text form.

content\_id: Stores id number of the content to which the comment belongs.

content\_type: Stores a number to determine the type of the content (page, chapter, course etc.).

date: Stores the insertion date of the comment to the system.

user\_id: Stores the id number of the user that inserts the comment record to the system.

---

#### 4.2.4 COURSE TABLE

---

Field	Type	Nullable	Foreign Key	References
<u>course_id</u> (P.K.)	integer(10)	No	No	-
course_name	varchar(250)	No	No	-
instructor_id	integer(10)	No	Yes	user_id(User)

**Table 4. Course Table**

Course is main heading of our tool. It includes chapters and chapters include pages which keep all needed education information. Course table (Table 4) has one foreign key, instructor\_id to refer user\_id in User table (Table 10) and its primary key is course\_id.

course\_id: Stores a unique number for every data in table. That is also used for primary key.

course\_name: Stores the name of the course.

instructor\_id: Stores the id number of the user that inserts the course record to the system.

---

#### 4.2.5 IMAGE TABLE

---

Field	Type	Nullable	Foreign Key	References
<u>image_id</u> (P.K.)	integer(10)	No	No	-
image	image	No	No	-
insert_user_id	integer(10)	No	Yes	user_id(User)

**Table 5. Image Table**

Images are needed to show some visual elements in courses. Instructors can add schemas, charts or pictures formatted as image. Image table (Table 5) stores image itself and needed information for image. It has one foreign key, insert\_user\_id to refer user\_id in User table (Table 10). Primary key of the table (Table 5) is image\_id.

image\_id: Stores a unique number for every data in table. That is also used for primary key.

image: Stores the formatted image.

insert\_user\_id: Stores the id number of the user that inserts the image record to the system.

---

#### 4.2.6 PAGE TABLE

---

Field	Type	Nullable	Foreign Key	References
<u>page_id</u> (P.K.)	integer(10)	No	No	-
page_number	integer(4)	Yes	No	-
page_name	varchar(250)	Yes	No	-
chapter_id	integer(10)	Yes	Yes	chapter_id(Chapter)
template_id	integer(5)	No	Yes	template_id(Template)
course_id	integer(10)	Yes	Yes	course_id(Course)
user_id	integer(10)	No	Yes	user_id(User)

**Table 6. Page Table**

All needed information for learners will be hold in pages. To keep these information, he have page table (Table 6). The information of how page will be shown to user will keep in template table (Table 8). Template\_id in Page table(Table 6) will refer to Template table (Table 8). There are also three foreign keys, chapter\_id and course\_id refer Chapter (Table 2) and Course (Table 4) table to determine which course and chapter that the page belongs to; and user\_id to keep the instructors information who adds the page to the system. Primary key of the table (Table 6) is page\_id.

page\_id: Stores a unique number for every data in table. That is also used for primary key.

page\_number: Stores the number of the page in a chapter, or a course.

page\_name: In some conditions, a page would have a name instead of a number. This page\_name attribute stores the name of the page.

chapter\_id: Stores the id number of the chapter in which the page placed in.

template\_id: Stored the id number of the template that the page uses.

course\_id: Stores the id number of the course in which the page placed in.

user\_id: Stores the id number of the user that inserts the page record to the system.

---

#### 4.2.7 RATE TABLE

---

Field	Type	Nullable	Foreign Key	References
<u>rate_id</u> (P.K.)	integer(10)	No	No	-
point	integer(2)	No	No	-
content_id	integer(10)	No	No	-
content_type	integer(2)	No	No	-
user_id	integer(10)	No	Yes	user_id(User)

**Table 7. Rate Table**

In Mobile Content generator, users will be able to rate every page, chapter or course. To store these ratings, we have Rate table (Table 7). Primary key of this table is rate\_id. There is one foreign key user\_id to refer User table (Table 10) to keep inserted user's information. In content\_id attribute holds information about which rating belongs to which chapter, course or page. To distinguish these contents, content\_type will hold type of the content.

rate\_id: Stores a unique number for every data in table. That is also used for primary key.

point: Stores the point that is given by the user to content.

content\_id: Stores id number of the content to which the rate belongs.

content\_type: Stores a number to determine the type of the content (page, chapter, course etc.).

user\_id: Stores the id number of the user that inserts the rate record to the system.

---

#### 4.2.8 TEMPLATE TABLE

---

Field	Type	Nullable	Foreign Key	References
<u>template_id</u> (P.K.)	integer(5)	No	No	-
template_code	varchar(5000)	No	No	-
template_dsc	varchar(250)	Yes	No	-

**Table 8. Template Table**

In Template table (Table 8), we store html5 codes of pages. Template codes will include ids of videos, images and texts and be stored in template\_code attribute. Primary key of the table is template\_id.

template\_id: Stores a unique number for every data in table. That is also used for primary key.

template\_code: Stores the CSS code of the template. Code does not include any specific information like video\_id or image\_id. That attributes will be inserted to the code dynamically to create the page.

template\_dsc: Stores the description of the template (i.e. video\_text, image\_text or else.)

---

#### 4.2.9 TEXT TABLE

---

Field	Type	Nullable	Foreign Key	References
<u>text_id</u> (P.K.)	integer(10)	No	No	-
text	varchar(5000)	No	No	-
date	date	No	No	-
user_id	integer(10)	Yes	Yes	user_id(User)

**Table 9. Text Table**

Text table (Table 9) will include texts of pages. There is one foreign key user\_id to refer User table (Table 10) to store the person who adds text to the system, and the primary key of the table is text\_id.

text\_id: Stores a unique number for every data in table. That is also used for primary key.

text: Stores the text that used in pages.

date: Stores the insertion date of the comment to the system.

user\_id: Stores the id number of the user that inserts the text record to the system.

---

#### 4.2.10 USER TABLE

---

Field	Type	Nullable	Foreign Key	References
<u>user_id</u> (P.K.)	integer(10)	No	No	-
username	varchar(25)	No	No	-
password	varchar(10)	No	No	-
role_id	integer(2)	No	Yes	role_id(User_role)
address	varchar(500)	Yes	No	-
email	varchar(50)	No	No	-
tckimlik	varchar(11)	Yes	No	-

**Table 10. User Table**

User table (Table 10) will store personal information of all kind of users. There is one foreign key role\_id that refers to User\_role table (Table 11) and stores the role of this user in system. Primary key of the table (Table 10) is user\_id attribute.

user\_id: Stores a unique number for every data in table. That is also used for primary key.

username: Stores the username of the user. That username will be used to login to the system.

password: Stores the password of the user.

role\_id: Stores an id number to determine the role of the user (admin, learner, instructor or else)

address: Stores the address information of the user.

email: Stores the e-mail information of the user.

tckimlik: Stores the T.C. Kimlik No. of the user.

---

#### 4.2.11 USER\_ROLE TABLE

---

Field	Type	Nullable	Foreign Key	References
<u>role_id</u> (P.K.)	integer(10)	No	No	-
role_name	varchar(50)	No	No	-

**Table 11. User\_Role Table**

We will define different roles to assign users and give different authorizations. User\_role table (Table 11) will store these role information. It has two attributes, role\_id as primary key and role\_name to identify the role.

role\_id: Stores a unique number for every data in table. That is also used for primary key.

role\_name: Stores the name of the role (admin, instructor, learner, etc.)

---

**4.2.12 VIDEO TABLE**

---

Field	Type	Nullable	Foreign Key	References
video_id(P.K.)	integer(10)	No	No	-
video_link	varchar(50)	No	No	-
insert_user_id	integer(10)	No	Yes	user_id(User)

**Table 12.Video Table**

Our system will not be stored videos in database. Instead, we will upload all videos to www.kaltura.com web-site and take the link of the video. In video\_link attribute, we store this data. There is one foreign key insert\_user\_id that refers to User table (Table 10) and keep information about user who adds this video to system. Primary key of the table (Table 12) is video\_id.

video\_id: Stores a unique number for every data in table. That is also used for primary key.

video\_link: Stores the link of the video that comes from Kaltura web site.

insert\_user\_id: Stores the id number of the user that inserts the video record to the system.

---

## 5 SYSTEM ARCHITECTURE

---

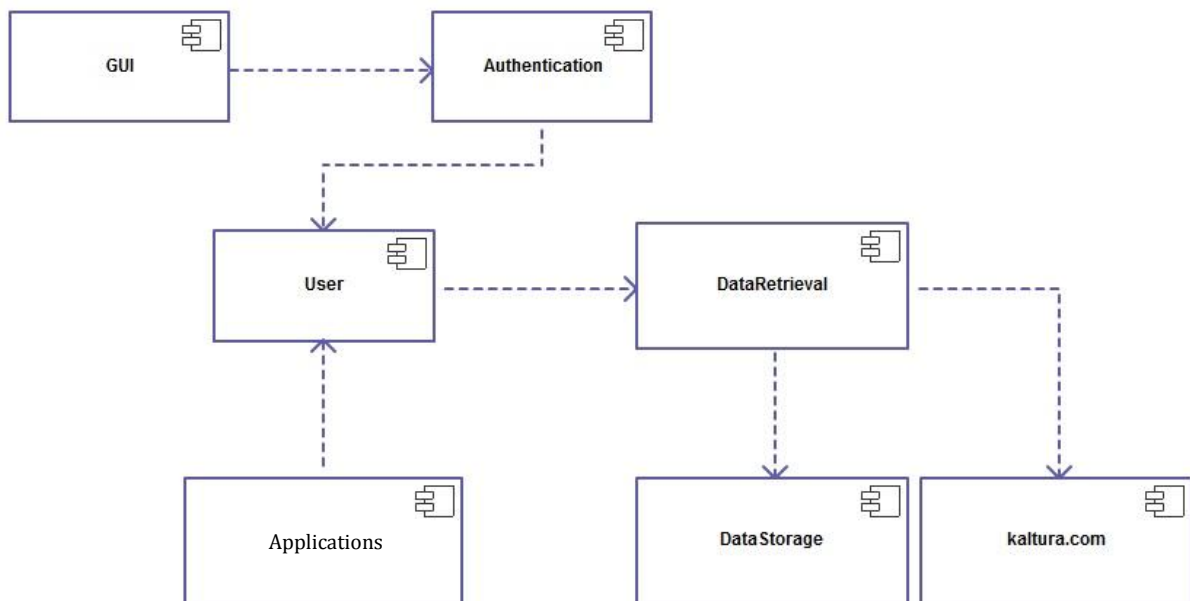
A general description of the Mobile Content Generator system architecture is presented in the following sections.

---

### 5.1 ARCHITECTURAL DESIGN

---

Mobile Content Generator is a web-based software which is composed of seven components; GUI, Authentication, User, DataRetrieval, BackEndApplication, DataStorage, Kaltura.com. The component diagram illustrates structure of the system. Each component will be explained separately in the next section.



**Fig5. Component Diagram of the System**

---

### 5.2 DESCRIPTION OF COMPONENTS

---

In this section, components mentioned in 5.1 are described individually a section. Each description contain short definition, processing narrative for component, interface description, component processing detail and dynamic behavior.

---

#### 5.2.1 GUI

---

Graphical user interface provides user to use system easily. This is obtained by using button, text box etc.

---

##### 5.2.1.1 PROCESSING NARRATIVE FOR GUI

---

This component comprises all the objects and shows them as appropriate kind. This shows all hard part clearly as everybody can understand what s/he should do. This component works in

client side through HTML5. Server side sends desired information and client side interpret as desired. This component manages this overflow by creating dynamic HTML5 webpage through ASP.NET.

---

#### 5.2.1.2 GUI INTERFACE DESCRIPTION

---

There are inputs from client side. Then, this input stack sends to server side with necessity information. Server side processes this information and also makes some information transformation with other components and return desired information as output. On the other hand, this output stack reaches the client side successfully. Client side creates dynamically desired webpage through using output information.

---

#### 5.2.1.3 GUI PROCESSING DETAIL

---

The complete procedural activities related to this component are as follow;

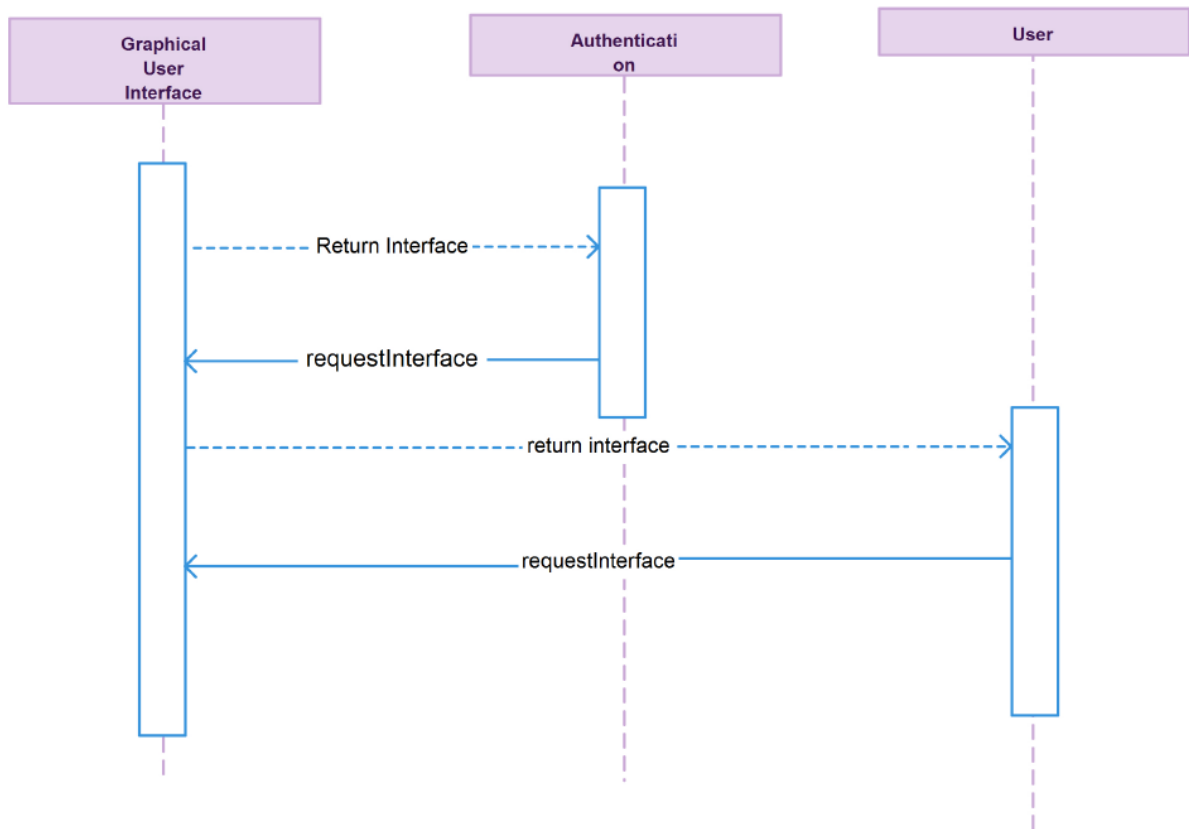
1. User/client requests a page from the system through the internet
2. Server captures the request successfully
3. Server evaluates this request inside
4. Server gets necessary information from data storage component
5. Server returns information as output
6. Client side takes information successfully
7. Client evaluates the server's output
8. Client creates webpage dynamically and show to user

---

#### 5.2.1.4 GUI DYNAMIC BEHAVIOR

---

This component has relation with authentication and user components. This component interacts with authentication component as authentication component calls this component. User component also calls this component to show itself on webpage. This component depends on receiving information from these two components.



**Fig6. Sequence diagram of GUI**

---

## 5.2.2 AUTHENTICATION

---

This component provides to separate people who have permission to enter system and also separate people who are administrator, instructor and learner.

### 5.2.2.1 PROCESSING NARRATIVE FOR AUTHENTICATION

---

This component is responsible for checking personal requests with the permission of the clients. This prevents people who are not related with the system. This is implemented as username and password. At this point, everybody who is related with system have unique username and password. On the other hand, this component separates profile of people who is related with system. Learner cannot reach or display instructor profile.

### 5.2.2.2 AUTHENTICATION INTERFACE DESCRIPTION

---

This has two areas. First one is username, second one is password. Also there is log in button under these two text boxes. Log in button sends all information in the text box to server side to check. Output depends on whether information is correct or not. User can go next step or try again.

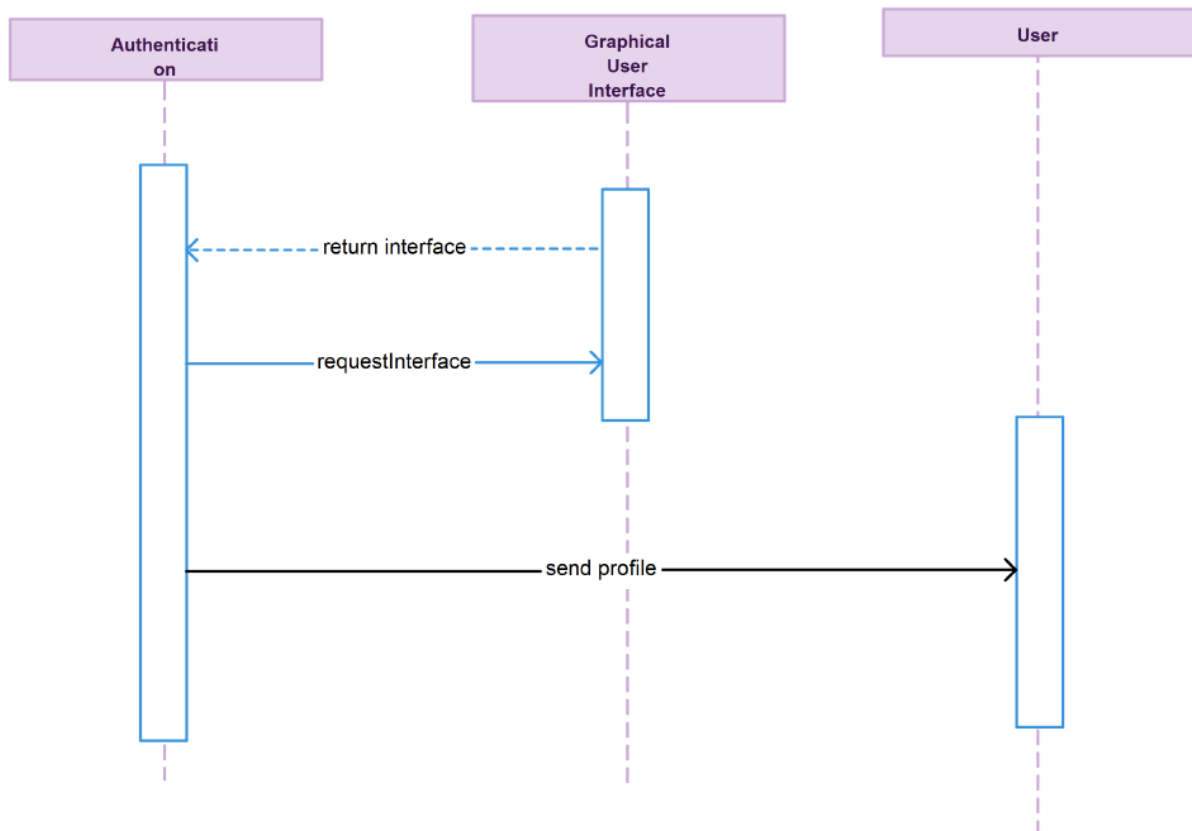
### 5.2.2.3 AUTHENTICATION PROCESSING DETAIL

The complete procedural activities related to this component are as follow;

1. User/client fills the username and password area
2. User/client clicks the log in button under this area
3. Client side sends these information to server side to check
4. Server side takes these information successfully
5. Server side check information from database
6. Server side return different information stack whether correct or not
7. Client side gets information successfully
8. Client side interprets received information
9. Client side create dynamically webpage according to received information
10. User see the desired webpage

### 5.2.2.4 AUTHENTICATION DYNAMIC BEHAVIOR

This component has relation with GUI component and user component. This calls GUI component to show itself as webpage. On the other hand, this component directs client to user component according to receiving information from server side.



**Fig7. Sequence diagram of Authentication**

---

## 5.2.3 USER

---

### 5.2.3.1 PROCESSING NARRATIVE FOR USER

---

This component is responsible for sending requests to DataRetrieval. Only the authenticated components become user components. It has 3 kinds of sub-components as administrator, instructor and learner. User component has corresponding applications. It acts as a bridge between applications and data retrieval. User can request a data from DataRetrieval or can request to store data to DataStorage.

### 5.2.3.2 USER INTERFACE DESCRIPTION

---

It generates data requests for DataRetrieval. By applications requests are processed with corresponding program. Processes like uploading video, text, audio or accessing to these contents are requested and results of these requests are returned from DataRetrieval.

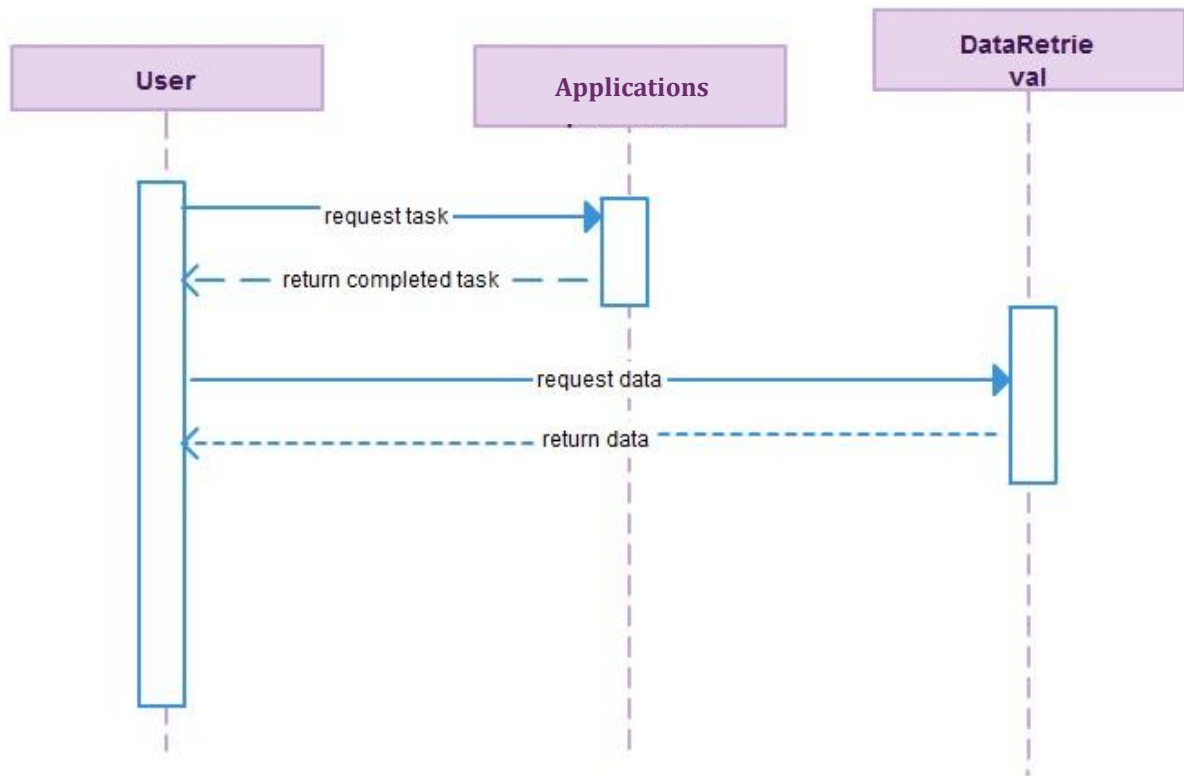
### 5.2.3.3 USER PROCESSING DETAIL

---

It works as follows;

1. It generates requests for loading from or storing to database by using Applications.
2. Results of requests are returned to user component from DataRetrieval.

### 5.2.3.4 USER DYNAMIC BEHAVIOR



**Fig8. Sequence diagram of User**

## 5.2.4 APPLICATIONS

### 5.2.4.1 PROCESSING NARRATIVE FOR APPLICATIONS

This component is for handling different tasks using technologies available. These technologies include ASP.net, JavaScript, HTML5 and CSS.

### 5.2.4.2 APPLICATIONS INTERFACE DESCRIPTION

It generates result for User using the User's request as input and produces output.

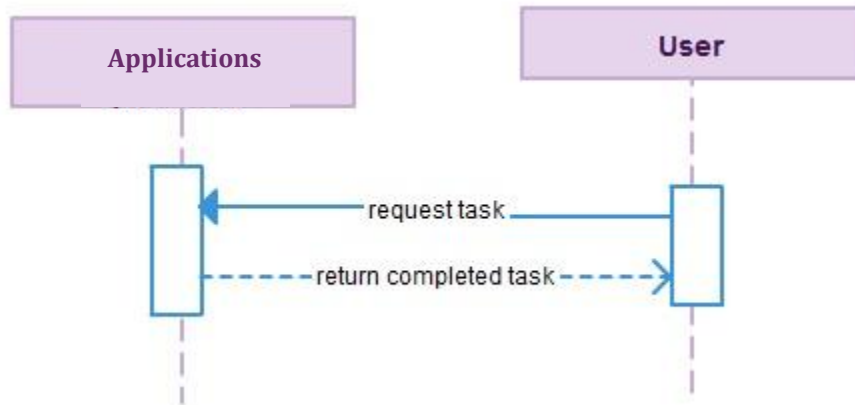
### 5.2.4.3 APPLICATIONS PROCESSING DETAIL

It works as follows;

1. User requests a task and triggers an appropriate task for application.
2. The application returns processed task's result to user.

#### 5.2.4.4 BACKEND APPLICATIONS DYNAMIC BEHAVIOR

---



**Fig9. Sequence diagram of Backed Application**

---

#### 5.2.6 DATA RETRIEVAL

---

##### 5.2.5.1 PROCESSING NARRATIVE FOR DATA RETRIEVAL

DataRetrieval is responsible for accessing data from and storing data to DataStorage and Kaltura.com. It provides a connection between users and database. Users requests are sent to DataStorage or Kaltura.com via DataRetrieval.

##### 5.2.5.2 DATA RETRIEVAL INTERFACE DESCRIPTION

It receives data requests from users of the system. Then, these requests are converted into SQL commands and sent to DataStorage or Kaltura.com for getting data objects. The result obtained from DataStorage or Kaltura.com is reported to the component which sent request for it.

##### 5.2.5.3 DATA RETRIEVAL PROCESSING DETAIL

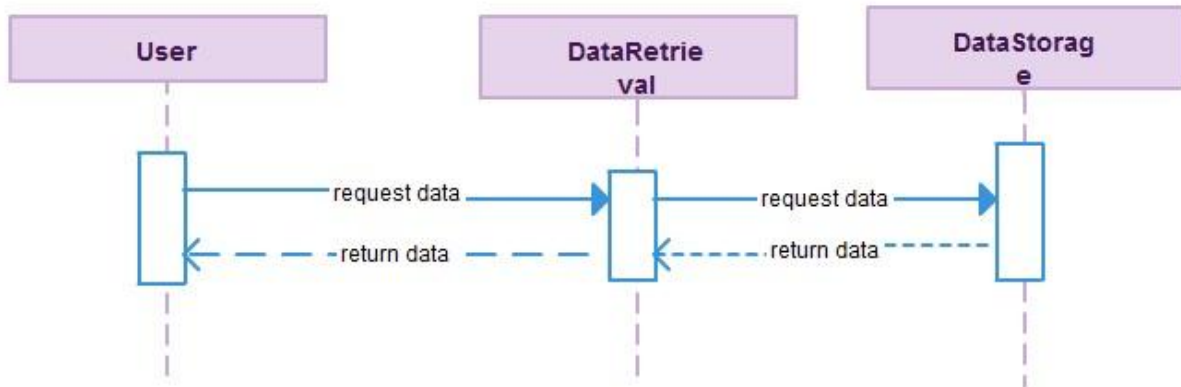
---

It works as follows;

1. Receive a request from User component.
2. Request is converted into SQL commands.
3. Send commands to DataStorage or Kaltura.com components
4. Received object from DataStorage or Kaltura.com is reported to the component which sent request for it.

#### 5.2.5.4 DATA RETRIEVAL DYNAMIC BEHAVIOR

---



**Fig10. Sequence diagram of Data Retrieval**

---

#### 5.2.7 DATA STORAGE

---

##### 5.2.6.1 PROCESSING NARRATIVE FOR DATA STORAGE

---

Data Storage component is responsible for creating and storing data objects. It provides the data which is requested from DataRetrieval. ADO.NET is used as a connector between data storage and ASP.net to get data from database and create corresponding data objects.

##### 5.2.6.2 DATA STORAGE INTERFACE DESCRIPTION

---

DataRetrieval sends data requests to DataStorage for further processing. Then, DataStorage processes these commands and with the help of ADO.NET, the processed SQL commands are put to objects for future use.

##### 5.2.6.3 DATA STORAGE PROCESSING DETAIL

---

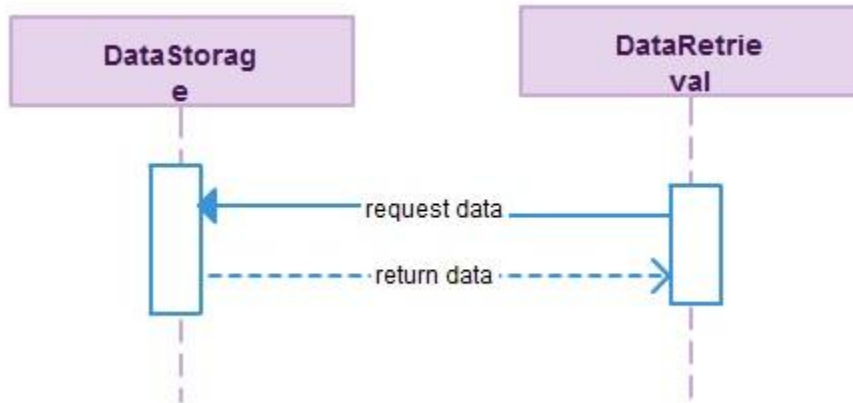
It works as follows;

1. Receive a data request from DataRetrieval component.
2. Converted SQL queries are processed with ADO.NET.
3. Results of queries are put to objects.

---

#### 5.2.6.4 DATA STORAGE DYNAMIC BEHAVIOR

---



**Fig11. Sequence diagram of Data Storage**

---

#### 5.2.7 KALTURA.COM

---

##### 5.2.7.1 PROCESSING NARRATIVE FOR KALTURA.COM

---

This component is responsible for video upload and video streaming. When an instructor upload a video, system will connect [www.kaltura.com](http://www.kaltura.com) , upload the video and give back video link. Then the video link will be saved to our database. When a user wants to watch a video, parameters will be sent to [www.kaltura.com](http://www.kaltura.com) and video will be retrieved in appropriate way

##### 5.2.7.2 KALTURA.COM INTERFACE DESCRIPTION

---

It receives data inputs and as JSON objects. It will encode video in its own server and send outputs as form of JSON objects. Then JSON objects will be parsed and saved in database if the request is video upload request done by instructor. On the other hand, if the request is a video streaming request, then output will be sent to user's browser.

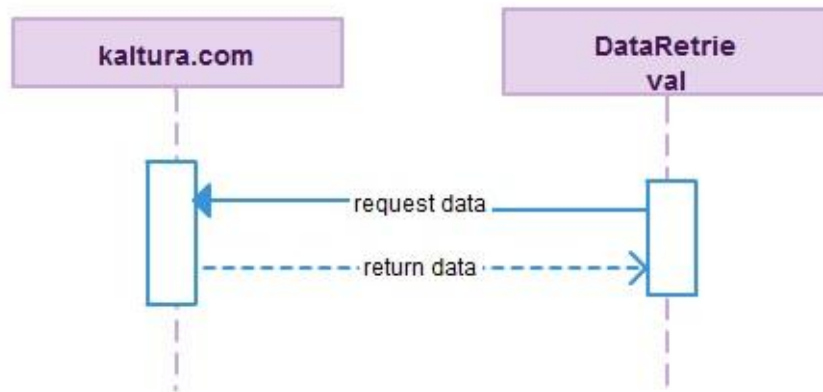
##### 5.2.7.3 KALTURA.COM PROCESSING DETAIL

---

It works as follows;

1. It receives a request as form of JSON object.
2. It decodes and applies needed processes.
3. When process finished, it sends appropriate JSON objects according to type of the request it received.

#### 5.2.7.4 KALTURA.COM DYNAMIC BEHAVIOR



**Fig12. Sequence diagram of Kaltura.com**

### 5.3 DESIGN RATIONALE

Firstly, login page of system appears on the screen as webpage. Then, user who is learner, instructor or administrator enters his/her information into text box and click log in button. According to profile, login page directs him/her to own profile page.

If s/he is administrator, there appears control panel; his/her own information and other applications. S/he can delete comment, rate, video, text, course, instructor and learner. S/he can see everything about system as desired and interfere in this situation.

If s/he is instructor, there appears his/her courses and control panel. S/he uses control panel to add course. Adding course also directs him/her new page. Firstly, s/he enters information about courses, then selects appropriate template, lastly; if necessary add video using kaltura.com, text, audio, image; then publish this course. Later, s/he selects people or group who can see this course.

If s/he is learner, there appears his/hers courses. S/he selects any course which has been allowed for his/her and starts or continues to receive education. S/he can write comment or rate.

Any user profile can exit from system anytime by clicking logout button.

### 5.4 TRACEABILITY OF REQUIREMENTS

3.2.3.3	5.2.2
3.2.16.3	5.2.2
3.2.17.3	5.2.3
3.2.21.3	5.2.3

## 6 USER INTERFACE DESIGN

In this section, user interface of Mobile Content Generator is explained in detail.

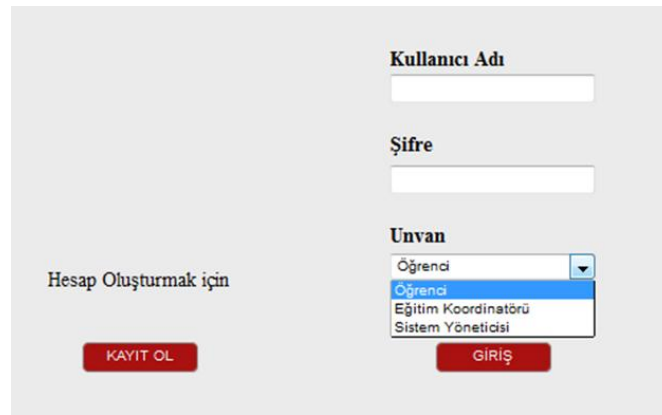
### 6.1 OVERVIEW OF USER INTERFACE

System has three types of users; Learner, Instructor, Admin. These users have different authorization and capabilities. Login mechanism is same for all users. They will choose their status as whom they want to login and submit valid password and ID. An additional function for Learner is sign up. A Learner who is not registered must sign up and fill requested form. After his account is approved, he can login the system. Admin accounts of the system are authorized when the system is setup. Instructors will be added by admins of the system. Therefore, admins and instructors do not sign up. Just after login, these users will be directed to different pages. All types of user can use the system through desktop environment, however only Learner is also able to reach it via mobile devices.

### 6.2 SCREEN IMAGES

All screen images of MCG will be shown in the following subsections. All screens are supported by Desktop environment and all Learner Screens also supported by Mobile environments.

#### 6.2.1 LOGIN



The screenshot displays a login form with the following elements:

- Kullanıcı Adı**: A text input field for the username.
- Şifre**: A text input field for the password.
- Unvan**: A dropdown menu with options: Öğrenci (selected), Öğrenci, Eğitim Koordinatörü, and Sistem Yöneticisi.
- Hesap Oluşturmak için**: Text indicating the sign-up option.
- KAYIT OL**: A red button for registration.
- GİRİŞ**: A red button for login.

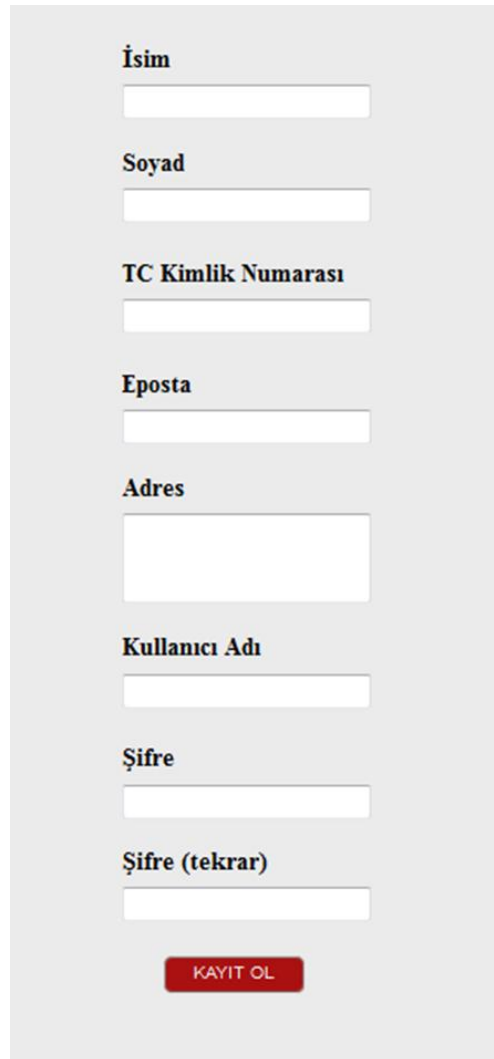
*Fig13. Login Screen*

When the webpage of the Mobile Content Generator is open, user faces with the login screen (Fig13). This screen can be seen through desktop and mobile platforms. Users (learners, instructors, admins) which are authenticated previously, can login the system with accurate username and password that matches with the title of the person. For unauthorized users there is another option which is sign up. Only Learners can use this action. Instructors are added to system by Admins and admins are predefined by the system.

---

## 6.2.2 SIGN UP

---



The image shows a vertical sign-up form on a light gray background. The form consists of several input fields, each with a label above it: 'İsim' (Name), 'Soyad' (Surname), 'TC Kimlik Numarası' (TC Identification Number), 'Eposta' (Email), 'Adres' (Address), 'Kullanıcı Adı' (Username), 'Şifre' (Password), and 'Şifre (tekrar)' (Password (repeat)). Each label is in bold black text. Below the last two input fields is a red button with the white text 'KAYIT OL' (REGISTER).

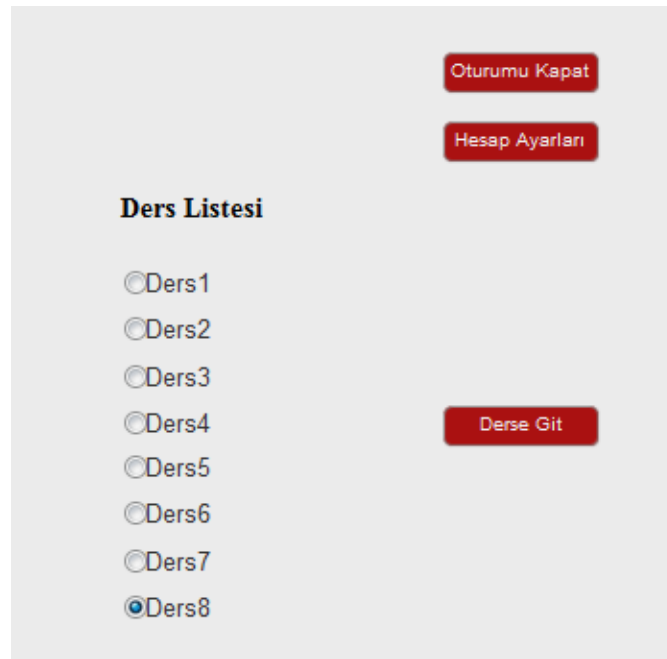
**Fig14. Sign Up Screen**

When a user clicks the “Kayıt Ol” button at Login page(Fig13), he is directed to Sign Up page (Fig14). This page requests necessary information to enroll the system. When a user is signed up the system with filling this form, he can login to system as Learner type of user. Sign Up page (Fig14) works for both Desktop and Mobile environments.

---

### 6.2.3 LEARNER HOMEPAGE

---



**Fig15. Learner Homepage Screen**

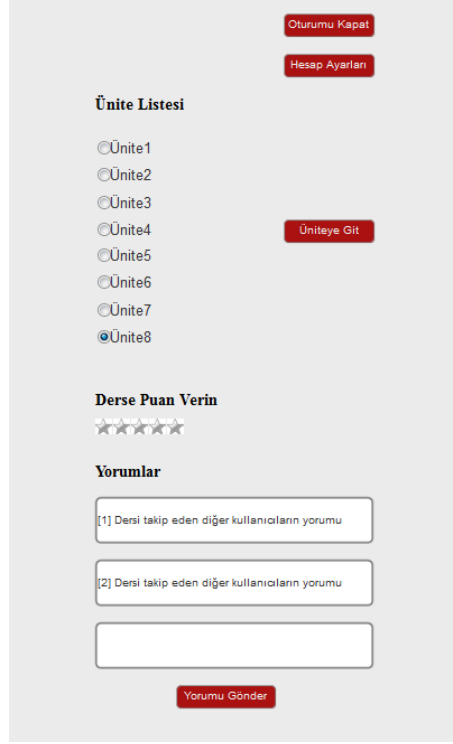
When a user is logged in the system as Learner, he is directed to Learner homepage (Fig15). Learner sees list of all courses that he is assigned to. He can choose one and go to course page with clicking “Derse Git” button. He also sees to buttons “Oturumu Kapat” and “Hesap Ayarları”. If Learner clicks “Oturumu Kapat”, he logs out from system and web page is directed to Login page (Fig13). If Learner clicks “Hesap Ayarları” button, he is directed to the form page same as Sign Up page (Fig14) filled with user’s information. He can do changes at that page.

---

### 6.2.4 COURSE SCREEN FOR LEARNER

---

When Learner selects a course to go from Learner homepage (Fig15), he is directed to course page for Learner (Fig16). There is a list of Chapters belonging to current Course. He can select a Chapter from this list to go. Learner can give rate to overall course. At the bottom of the page, comments done by other Learners for current course exists. If learner wants to send his thoughts, he can use the form and his comment will be able to seen by all the users authorized to course. “Oturumu Kapat” and “Hesap Ayarları” buttons stand at the right upper corner like Learner Home Page (Fig15).



*Fig16. Course Screen for Learner*

## 6.2.5 CHAPTER SCREEN FOR LEARNER



*Fig17. Chapter Screen for Learner*

This page represents content of the current Chapter for Learner (Fig17). There is a list of Pages. One can be selected and directed to the Page with “Sayfaya Git” button. User can give rate to current Chapter and view other Learners comment. With Learners will, he can also submit comment for the Chapter. Functionality of “Oturumu Kapat” and “Hesap Ayarları” is the same with previous screens (Fig15, Fig 16).

---

## 6.2.6 PAGE SCREEN FOR LEARNER

---

After Learner has chosen a Page at Chapter Screen for Learner (Fig17) and directed Page Screen for Learner, he faces with learning material. This screen can be in different forms. Instructor decides the template of Page which fits best for his purpose. These options are limited by number of four (Video-Text, Audio-Text, Image-Text and Image-Audio). Each possible template that can be faced by Learner will be shown in the following subsection.

---

### 6.2.6.1 VIDEO-TEXT

---



**Fig18. Video-Text Screen for Learner**

Video-Text is a template of Page screen which contains learning material provided by Instructor as shown at Fig18. Learner can watch the video and when he wants, he can stop. System records the information of which point Learner stopped and left from Page. By means of this feature of the system, when the Learner reenters the Page and press the play button, video continues from the point where Learner left. Learner does not have to deal with to find where he had come at the video. In addition to this feature, Learner can choose any point at timeline bar at the video player and continue to watch from that point. Text written by Instructor stands below video. Under these, Learner can rate learning material of the Page. Underneath, Learners comments exits and user can also submit his comment; after writing comment by clicking “Yorumu Gönder” button. After finishing work with current Page, Learner can go next Page of the learning material by “Sonraki Sayfa” button or go back to previous Page of the learning material by “Önceki Sayfa” button.

#### 6.2.6.2 AUDIO-TEXT



**Fig19. Audio-Text Screen for Learner**

Audio-Text (Fig19) is another template of Learner Page. The page contains an audio player and text below it. Like Video-Text Page(Fig18), this Page also contains rating, showing comments, submitting comments, going next or previous pages, account settings and log out buttons.

### 6.2.6.3 IMAGE-TEXT



**Fig20. Image-Text Screen for Learner**

As indicated in its name, this template's main components are an image and a text as shown at the Fig20. Learner can rate the Page, submit comment and see others comment about the material. Clicking "Önceki Sayfa" or "Sonraki Sayfa" changes current Page. Learner can log out or change account settings with the buttons on the right upper corner.

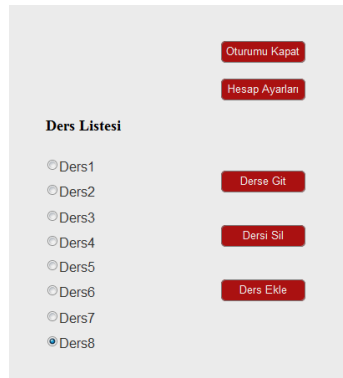
#### 6.2.6.4 IMAGE-AUDIO



**Fig21. Image-Audio Screen for Learner**

The learning material in this Page consists of an Image and an audio as shown at Fig21. There exists an audio player under the image. Like other pages (Fig18-20) it contains rating and comment functionalities and “Önceki Sayfa”, “Sonraki Sayfa”, “Oturumu Kapat” and “Hesap Ayarları” buttons.

#### 6.2.7 INSTRUCTOR HOMEPAGE



**Fig22. Instructor Homepage**

This is the first Screen related to Instructor. After Instructor is logged in the system(Fig 13) , he is directed to Instructor Homepage(Fig22). At the right upper corners there are “Oturumu

Kapat” and “Hesap Ayarları” buttons. If Instructor clicks “Oturumu Kapat”, he logs out from system and web page is directed to Login page (Fig13). If Instructor clicks “Hesap Ayarları” button, he is directed to the form page same as Sign Up page (Fig14) filled with his information. He can do changes at that page. Main content of the Instructor Homepage (Fig22) is List of Courses that Instructor created and leaded. Instructor has the authorization to delete course by clicking “Dersi Sil” or creating a new course by clicking “Ders Ekle”. If he wants to see or change the content of a course, he can select a course, go Course by “Derse Git” Button.

## 6.2.8 COURSE SCREEN FOR INSTRUCTOR

The screenshot displays the instructor's course management interface. At the top right, there are two red buttons: "Oturumu Kapat" and "Hesap Ayarları". Below these, the "Ünite Listesi" section contains a vertical list of units from Ünite1 to Ünite8. Each unit has a radio button next to it, and Ünite8 is selected. To the right of each unit is a red button: "Üniteye Git" for Ünite1, "Üniteyi Sil" for Ünite2, "Yukarı Taşı" for Ünite3, "Aşağı Taşı" for Ünite4, and "Ünite Ekle" for Ünite5. Below the unit list is a checkbox labeled "Ders kayıtlı öğrenciler tarafından erişilebilir". The "Kayıtlı Öğrenciler" section features a scrollable list of students (Öğrenci1 to Öğrenci5) and two red buttons: "Seçimleri Sil" and "Öğrenci Sil". The "Derse Verilen Ortalama Puan" section shows a star rating of 4.5 out of 5. The "Yorumlar" section contains two comment boxes, each with a "Yorumu Sil" button. The first comment is "[1] Dersi takip eden diğer kullanıcıların yorumu" and the second is "[2] Dersi takip eden diğer kullanıcıların yorumu".

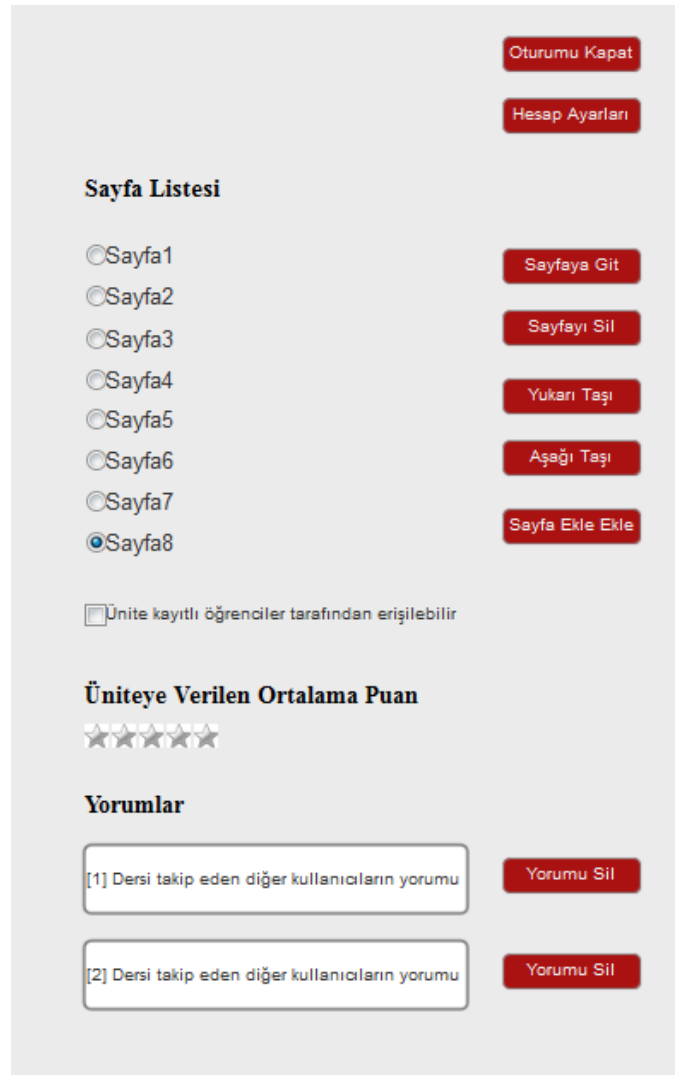
Fig23. Course Screen for Instructor

At this screen(Fig23), there is a list of chapters. Instructor can change ordering of the chapter with “Yukarı Taşı” or “Aşağı Taşı” buttons. He can delete a chapter with “Üniteyi Sil”, add a new chapter with “Ünite Ekle” or go to selected Chapter with “Üniteye Git” buttons. Instructor determines the visibility of the Course to Learners with the check box. Below Course list, there is a list of student enrolled to the course. Instructor can delete a Learner or multiple with “Seçilenleri Sil”. He can also add Learner to the Course. Downward, there is an informative part which shows the Rate of the Course given by Learners. At the bottom, comments done by Learners are shown. Instructor can delete any irrelevant or inappropriate comment by clicking “Yorumu Sil” button.

---

## 6.2.9 CHAPTER SCREEN FOR INSTRUCTOR

---



**Fig24. Chapter Screen for Instructor**

This screen (Fig24) is very similar to Course screen for Instructor(Fig23). He can select a Page from Page list and change order of it, delete it or go that page. Instructor can determine visibility of the Chapter with checking or unchecking the check box. He sees the overall rating from Learners about the Chapter. Instructor is able to delete any comment done by Learners. When Instructor clicks the “Sayfa Ekle” to create new Page, he should select a template from predefined templates; Video-Image(Fig18), Audio-Image (Fig19), Image-Text (Fig20) and Image-Audio (Fig21) than he is directed to a new screen(Video-Text Fig26, Audio-Text Fig27, Image-Text Fig28, Image-Audio Fig29) of the chosen template to compose Learners material.

---

### 6.2.10 PAGE SCREEN FOR INSTRUCTOR

---



**Fig25. Page Screen for Instructor**

In this screen (Fig25), Instructor sees the content of the page (templates explained in the sections 6.2.6.1-6.2.6.4). Below the content he can change visibility of the page with check box. He can delete any comment done by Learners.

---

## 6.2.11 PAGE GENERATION

---

When Instructor wants to create a new Page, he faces possible options, four Page template (shown at 6.2.6 from Learner's perspective). In this section, generation process of the Page is shown from Instructor's perspective.

### 6.2.11.1 VIDEO-TEXT

---



The screenshot shows a user interface for creating a page. At the top right, there are two buttons: "Oturumu Kapat" (Close Session) and "Hesap Ayarları" (Account Settings). In the center, there is a red button labeled "Video Yükle" (Upload Video). Below it is a text input field with the placeholder text "Ders metnini giriniz." (Enter course text). At the bottom center, there is a red button labeled "Kaydet" (Save).

**Fig26. Video-Text**

This template (Fig26) contains two components: video and text. Instructor should upload a video by clicking the button "Video Yükle". He should enter text through the text box. In order to save the changes he should click "Kaydet" button.

### 6.2.11.2 AUDIO-TEXT

---



The screenshot shows a user interface for creating a page. At the top right, there are two buttons: "Oturumu Kapat" (Close Session) and "Hesap Ayarları" (Account Settings). In the center, there is a red button labeled "Ses Dosyasını Yükle" (Upload Audio File). Below it is a text input field with the placeholder text "Ders metnini giriniz." (Enter course text). At the bottom center, there is a red button labeled "Kaydet" (Save).

**Fig27. Audio-Text**

This template (Fig27) requests uploading audio file. Instructor can upload audio file by clicking “Ses Dosyası Yükle” button. He can use the text box below this button to enter learning material as text. To save the Page, Instructor must click “Kaydet” button.

#### 6.2.11.3 IMAGE-TEXT

---



**Fig28. Image-Text**

When Instructor wants Learners to learn from the template described in 6.2.6.3, he uses this Page generator (Fig28). He can upload image with clicking “Resmi Yükle” button. Text of the learning material should be entered to text box. After forming page, Instructor must click “Kaydet” button to save the Page.

#### 6.2.11.4 IMAGE-AUDIO

---



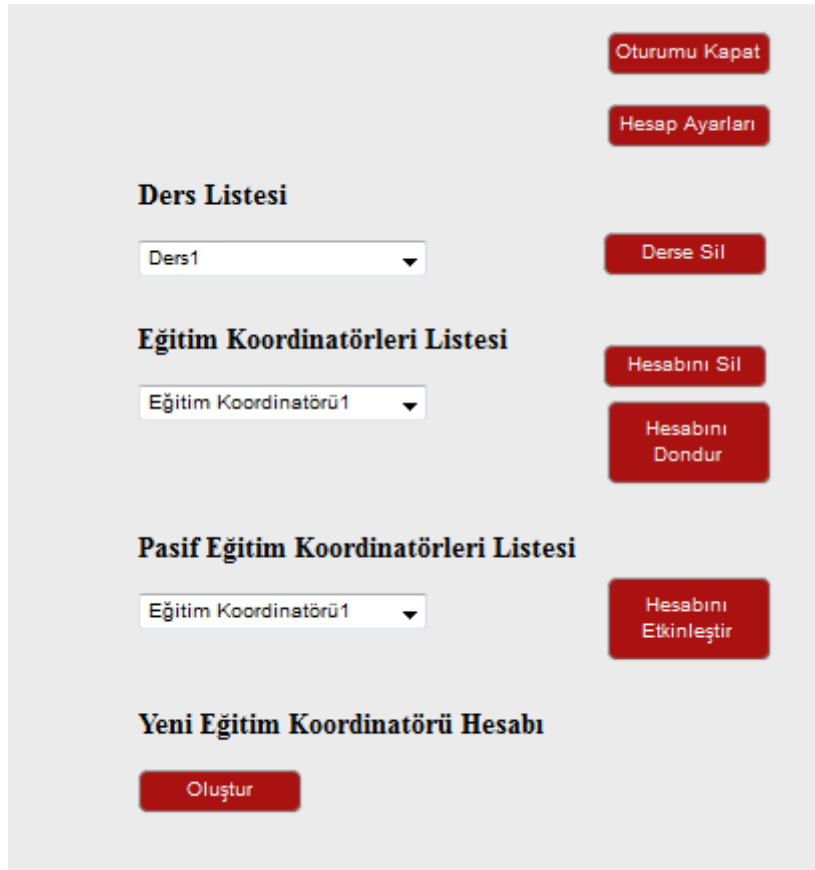
**Fig29. Image-Audio**

In order to form a Page shown at the 6.2.6.4, Instructor should use this generator (Fig29) . He should upload image file through “Resmi Yükle” button and audio file through “Ses Dosyasını Yükle” button. After successfully uploaded material, Instructor must save work by clicking “Kaydet” button.

---

## 6.2.12 ADMIN HOMEPAGE

---



*Fig30. Admin Homepage*

Admin is directed after logging in the system (Fig13) to Admin Homepage (Fig30). Admin’s permission is more than Instructor and Learner. He can delete course with selection from course list and clicking the “Dersi Sil” button. Admin can delete an Instructor account with selecting from Instructor List (shown as Eğitim Koordinatörleri Listesi) and clicking “Hesabı Sil” button. Another option is deactivation of Instructor account by clicking “Hesabı Dondur” button, thus the Instructor cannot use his account until Admin deactivate the account. There is another list of Instructors whose accounts are deactivated. Admin can chose Instructor from “Pasif Eğitim Koordinatörleri Listesi” and deactivates the account by clicking “Hesabı Etkinleştir” button. When Admins wants to add a new Instructor to the system, he can click “Oluştur” button.

## 6.3 SCREEN OBJECTS AND ACTIONS

---

Mobile Content Generator is interacted with user through a browser (workers whether on Desktop or on Mobile environment). Consequently screen object will be mainly HTML5 elements.

- **Label:** It is used for input fields. Where the user enters a text, this element will be used (username, password, name, etc.). Mainly used in login screen, sign up screen and page generate templates.
- **Check Box:** Especially used visibility of the Page / Chapter / Course to change by Instructor. When is checked the visibility by Learner type of user get changed.
- **Radio Button:** This element used in lists (lists of Courses / Instructors /Learners / Pages, etc.) to choose one element from the list. Generally selecting one used for directing current page to another Page.
- **Select:** Another element is select which is used to create drop lists. It is used several screens in the previous section.
- **Submit Button:** This button is used in every screen. One action of it is sending information that is already selected (like sign up page-Kayıt button). Another action is directing current page to target page (“Önceki Sayfa”, “Sonraki Sayfa”, “Hesap Ayarları”, etc).
- **Image:** This component is used at Page Templates to show learning material as Image.
- **Audio:** In Audio-Text and Image-Audio there is a need for audio field. Audio tag meets the need of an audio player. Audio file can be listened without downloading it, stopped and replayed.
- **Video:** In the Video-Text template there is used a video player. HTML5 tag fulfills the requirement. All the functions can be done with is that is expected from an media player.
- **Pop-up Window:** Pop-up window is used to ease the upload period. When a user wanted to upload a material, pop-up window is opened and user doesn't have to wait until upload is finished to other things.
- **Rate:** Rate is used by both Instructor and Learner. Instructor sees the average rating of the current material. Student can submit a rate to current material, and the rating is send to database.

## 7 DETAILED DESIGN

---

### 7.1 USER PACKAGE DETAILED DESIGN

---

**Classification:** This is one of main packages of the whole system.

**Definition & Responsibilities:** This package keeps information about a user of the system and it contains admin, learner and instructor classes of the system. It retrieves information about a user in the system and creates corresponding learner, instructor or admin classes for a user.

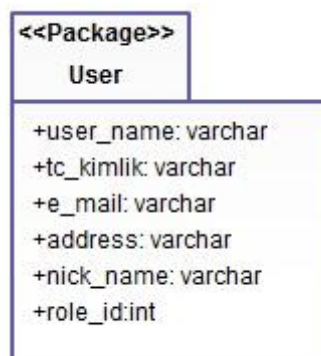
**Constraints:** This package does not have any methods. It holds only specified data fields required to hold a user.

**Composition:** User package consists of 3 classes which are admin, instructor and learner. The details of these classes are explained in 7.1.1, 7.1.2 and 7.1.2 subsections. Also, it consists of a sub-package called authentication which is explained in 7.1.4.

**Uses/Interactions:** User package interacts with GUI and Data Manager packages. GUI package creates appropriate graphical user interfaces for user. Graphical user interfaces of the system for users differs in kinds of admin, learner and instructors, and GUI package handles this difference between these interfaces. Data Manager package provides data for users.

**Resources:** -

**Processing:** Authentication sub-package returns information about user's title in the system and user package creates necessary admin, instructor or learner classes with these information. Then, it sends these created classes' information to Data Manager package to be stored in the database.



**Fig31. User Class Diagram**

- *user\_name: varchar*: stores name of the user.
- *tc\_kimlik:varchar*: stores TC Kimlik No of the user.
- *e\_mail: varchar*: stores e-mail address of the user.
- *address: varchar*: stores the address of the user.
- *nick\_name: varchar*: stores the nick name of the user s/he chooses.
- *role\_id:int*: stores the title of user in the system as admin, instructor or learner

---

### 7.1.1 LEARNER CLASS DETAILED DESIGN

---

**Classification:** This is a class within the User main package.

**Definition & Responsibilities:** This is a learner in the system and this class is designed to keep information about a learner and its functions in the database. It provides the functionality to a learner that s/he needs when using the system.

**Constraints:** Learner class only has methods for reaching contents, not for modifying any of the contents.

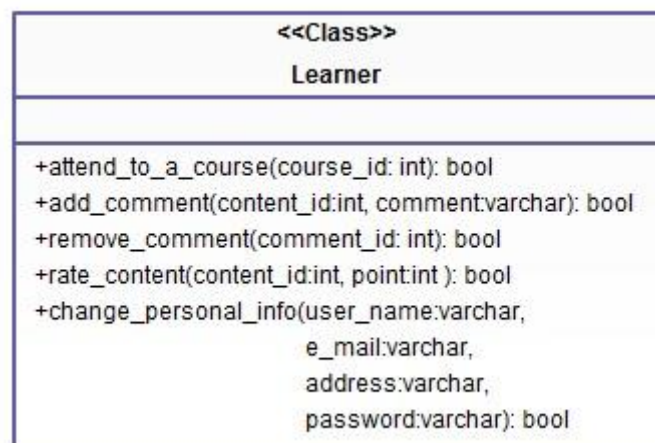
**Composition:** Learner class does not have any sub-component.

**Uses/Interactions:** Learner class only interacts with User package. It retrieves necessary information for creating a learner from User package. Also, it has an indirect interaction with Data Manager package through the User package.

**Resources:** User package is resource of Learner class.

**Processing:** Learner class retrieves learner information from User package and creates classes for them. When Learner has a request about modifying database like attending to a course, adding/removing comment, rating content or changing his/her personal information User package sends these requests to Data Manager package and Data Manager takes necessary steps for database management.

- *attend\_to\_a\_course(course\_id: int): bool* : attend to the course specified with the course\_id parameter
- *add\_comment(content\_id:int, comment:vchar):bool* : add a comment to the content specified with the content\_id
- *remove\_comment(comment\_id: int):bool* : remove the comment with the specified comment\_id
- *rate\_content(content\_id:int, point:int ):bool*: rate the content specified with the content\_id
- *change\_personal\_info(user\_name:vchar, e\_mail:vchar, address:vchar, password:vchar) :bool*: change the personal information with the specified parameters.



**Fig32. Learner Class Diagram**

---

## 7.1.2 INSTRUCTOR CLASS DETAILED DESIGN

---

**Classification:** This is a class within the User main package.

**Definition & Responsibilities:** This is an instructor in the system and this class is designed to keep information about an instructor and its functions in the database. It provides the functionality to an instructor that s/he needs when using the system.

**Constraints:** Instructor class has methods for reaching contents and also for modifying learning contents.

**Composition:** Instructor class does not have any sub-component.

**Uses/Interactions:** Instructor class only interacts with User package. It retrieves necessary information for creating an instructor from User package. Also, it has an indirect interaction with Data Manager Package gets and sets information to Database through the User package.

**Resources:** User package is resource of Instructor class.

**Processing:** Instructor class retrieves instructor information from User package and creates classes for them. When Instructor has a request about modifying database like viewing course, adding/removing course, chapter, page, comment, rating content or changing his/her personal information User package sends these requests to Data Manager Package and Data Manager takes necessary steps for database management.

- *addNewCourse ( object ) :bool:* compose a new course
- *addNewChapter ( object, courseID ) : bool:* add a new chapter to a course.
- *addNewPage(templated ) : bool:* add a new page to a chapter.
- *uploadVideo( Object ) : bool:* upload video to Kaltura through Data Manager Package
- *uploadAudio (Object ) : bool:* upload audio to database through Data Manager Package
- *uploadImage (Object ) : bool:* upload image to database through Data Manager Package
- *enterText ( String ) : bool:* save text to database through Data Manager Package
- *deleteCourse (contentID) : bool:* delete course from database through Data Manager Package
- *deleteChapter (contentID) :bool:* delete chapter from database through Data Manager Package
- *deletePage (contentID) : bool:* delete page from database through Data Manager Package
- *deleteComment (commentID) : bool:* delete comment from database through Data Manager Package
- *moveChapterUp (contentID) : bool:* rearrange order of the chapter to upper
- *moveChapterDown (contentID) : bool:* rearrange order of the chapter to lower
- *movePageUp (contentID) :bool:* rearrange order of the page to upper
- *movePageDown (contentID) : bool:* rearrange order of the page to lower
- *editPermission (contentID, bool) : bool:* change the visibility of the course to learners
- *enrollLearnerToCourse (learnerID) : bool:* add learners to course

**Interface/Exports:** Instructor class is responsible for functions of Instructor type user. The person can make changes about courses he is responsible and see overall content of the course via GUI than User (Instructor Class). Super of Instructor class (User) interacts with Data Manager.



**Fig33. Instructor Class Diagram**

---

### 7.1.3 ADMIN CLASS DETAILED DESIGN

---

**Classification:** This is a class within the User main package.

**Definition & Responsibilities:** This is an admin in the system and this class is designed to keep information about an admin and its functions in the database. It provides the functionality to an admin that s/he needs when using the system.

**Constraints:** Admin class has methods for regulate users and contents.

**Composition:** Admin class does not have any sub-component.

**Uses/Interactions:** Admin class only interacts with User package. It retrieves necessary information for creating an instructor from User package. Also, it has an indirect interaction with Data Manager Package gets and sets information to Database through the User package.

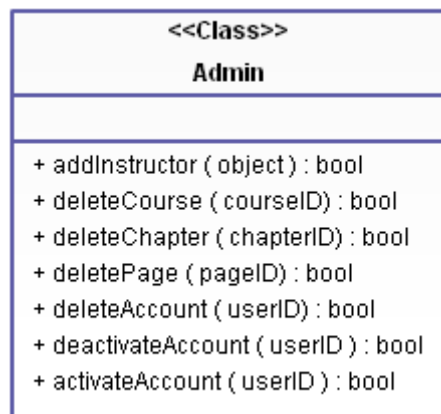
**Resources:** User package is resource of Admin class.

**Processing:** Admin class retrieves admin information from User package and creates classes for them. When Admin has a request about modifying database like adding/ deleting/ deactivating/ activating instructors/learners, User package sends these requests to Data Manager Package and Data Manager takes necessary steps for database management.

- *addInstructor ( object ) : bool*: composing an Instructor account

- *deleteCourse ( courseID ) : bool*: deleting a course from database through Data Management Package
- *deleteChapter ( chapterID ) : bool*: deleting a chapter from database through Data Management Package
- *deletePage ( pageID ) : bool*: deleting a page from database through Data Management Package
- *deleteAccount ( userID ) : bool*: deleting an User account from database through Data Management Package
- *deactivateAccount ( userID ) : bool*: deactivating an active User account from database through Data Management Package
- *activateAccount ( userID ) : bool*: activating a deactivate User account from database through Data Management Package

**Interface/Exports:** Admin class is responsible for functions of Admin type user. The person can make changes about courses, instructors and students via GUI than User (Admin Class). Super of Instructor class (User) interacts with Data Manager.



**Fig34. Admin Class Diagram**

---

## 7.1.4 AUTHENTICATION

---

**Classification:** Sub-package

**Definition:** Authentication is the process of determining whether someone is, who is declared to be.

**Responsibilities:** This sub-package gets necessity information from user and also gets necessity data information from database. Then, this compares these data information and confirms that valid user. After validation, this sub-package directs valid user the desired profile.

**Composition:** This sub-package contains two classes such as login and logout.

**Uses/Interaction:** Authentication interacts with data manager. Since, this sub-package needs user data information to confirm.

**Resources:** Authentication uses client-computer memory and processor. This consumption is so little.

**Processing:** This sub-package gets data information from user. Then, this sends this data information to data manager. After data manager processing, data manager retrieves necessity data information for graphic user interface.

**Interface/Exports:** This sub-package contains acceptance and refusing exceptions. When user is accepted from system, then user gets authentication is successful display. On the other hand, when user is refused from system, then user gets authentication is unsuccessful display.

---

#### 7.1.4.1 LOGIN

---

**Classification:** Class

**Definition:** This class checks user whether has profile or not on the database.

**Responsibilities:** Login class gets user information. Then, login class sends user information to data manager to get user's profile. After data manager processing, data retrieval returns data information about profile to graphic user interface.

**Constraints:** There is no constraint for this class.

**Composition:** This class is complete unit.

**Uses/Interactions:** Login class interacts directly data manager.

**Resources:** Login class uses only client-side computer resources such as memory, processor and internet. This usage is so little.

**Processing:** Login class gets data information from user through graphic user interface. This class sends necessity information to data manager.

**Interface/Exports:** Graphic user interface is provided for login class.

---

#### 7.1.4.2 LOGOUT

---

**Classification:** Class

**Definition:** Logout checks user whether exit or not.

**Responsibilities:** Logout class gets data information such as session id for user from graphical user interface. Then, this class destroys this session id and set cookies required data information. Also this class sends last changed data information to data manager to store user's profile.

**Constraints:** there is no constraint for this class.

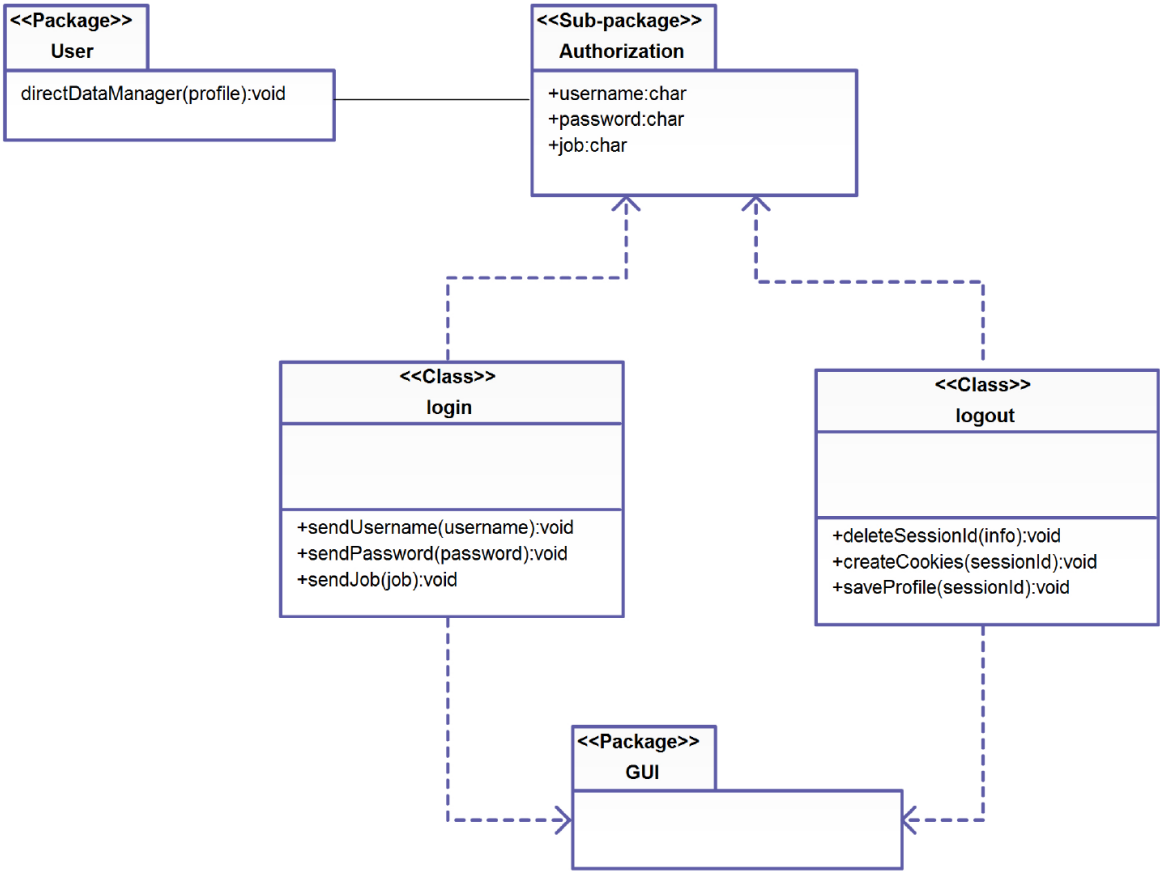
**Composition:** this class is complete unit.

**Uses/Interactions:** logout class interacts with graphical user interface and data manager.

**Resources:** Logout class cleans session data information from client-computer. However; cookies can use client-side computer memory. Cookies use so little memory in the temporary files.

**Processing:** Logout class gets session data information for specific user and sends necessity data information to data manager. Rest of data information is cleaned by logout class. By the way, if it is necessary for user, logout class creates cookies on the client-side computer memory.

**Interface/Exports:** Graphical user interface package provides successfully logout or problem on logout displays.



**Fig35. Authentication Class Diagram**

---

## 7.2 GRAPHIC USER INTERFACE

---

**Classification:** Package

**Definition:** This package provide interface that people can interact with computer. This makes easy usage of software for user.

**Responsibilities:** This package gets data structure, which contains detailed display information, from database manager through admin, learner and instructor sub-packages. This data structure of display information is interpreted from this package. These interpreted data is presented on the screen in order to make interaction with software for people.

**Constraints:** This package has dependency for JAVASCRIPT language and obtained display information. JAVASCRIPT language interprets on client side. In other words; this language interprets from user computer's browser. There is no support problem for today's browsers for this language. However; user can close support of JAVASCRIPT language on the browser. Then, if acquired data information contains part of JAVASCRIPT languages, then this package does not work properly and browser displays interface as corrupted. Another constraint is database manager. Database manager can provide corrupted data information because of more than maximum data retrieval request simultaneously, crash problem of server or internet connection problem. Graphic user interface uses session id as global variable.

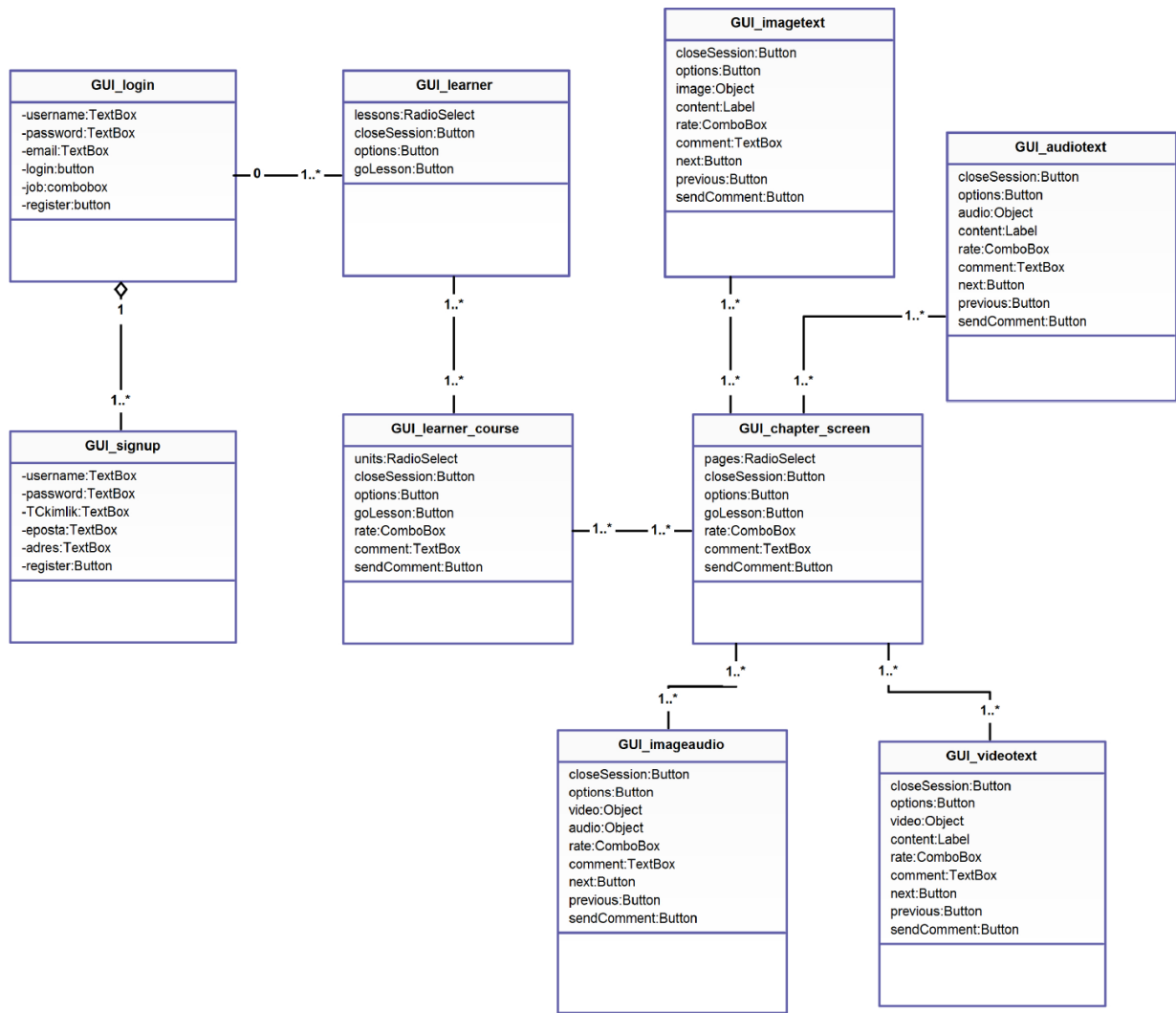
**Composition:** This package behaves as complete. This package gets and sends data information.

**Uses/Interactions:** This package interacts with user and data manager package. This package gets necessity data information from admin, instructor and learner sub-packages. In other words, user's sub-packages trigger graphic user interface package. Then, this package triggers data manager and gets required information from data manager.

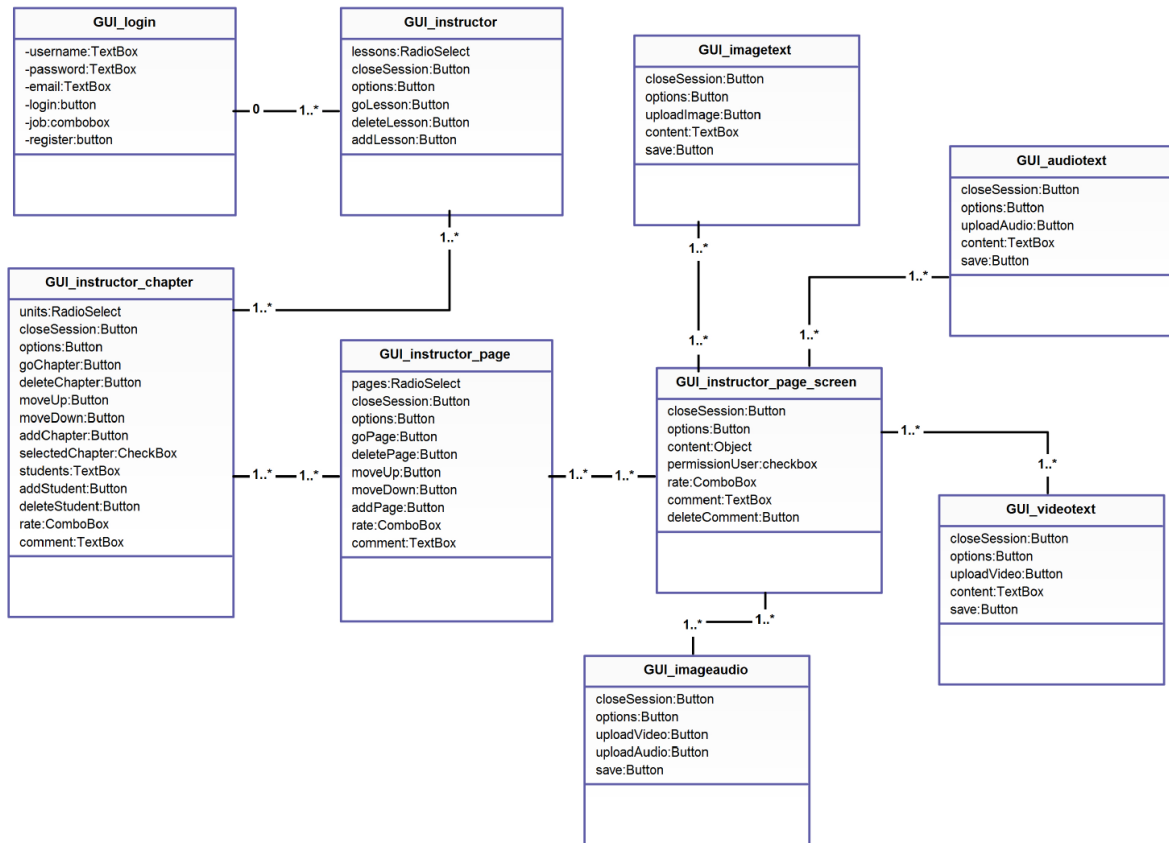
**Resources:** this package uses client-side computer resources and server-side computer resources. Client-side's memory, processor and graphic card are used. Client-side usage depends on only memory. Since, graphic user interface package gets data information from data manager and loads this information to client-side computer memory. At this point, if client-side memory is not enough, then there emerge problem to display screen. There are no other problems for client-side. If client-side browser works properly and there is enough temporary memory, then page is loaded directly. Server-side usage depends on simultaneously maximum connection number. If this maximum number is exceeded, then there emerges problem such as incomplete data information is sent to client-side. Another resource is internet connection. There emerges a problem if internet connection does not work properly. Client-side does not obtain proper data information. This causes problems such as client-side gets either no data information or corrupted data information.

**Processing:** This package is triggered firstly by server. Software user writes link on browser search bar and sends request for this package to server-side. Then server-side creates this package and sends back to client-side. Client-side browser interprets this data information. If there is script language in part of data information, then this part of data information is interpreted by client-side computer's web browser. When user interacts with graphic user interface, this package collects information given from user and sends server-side.

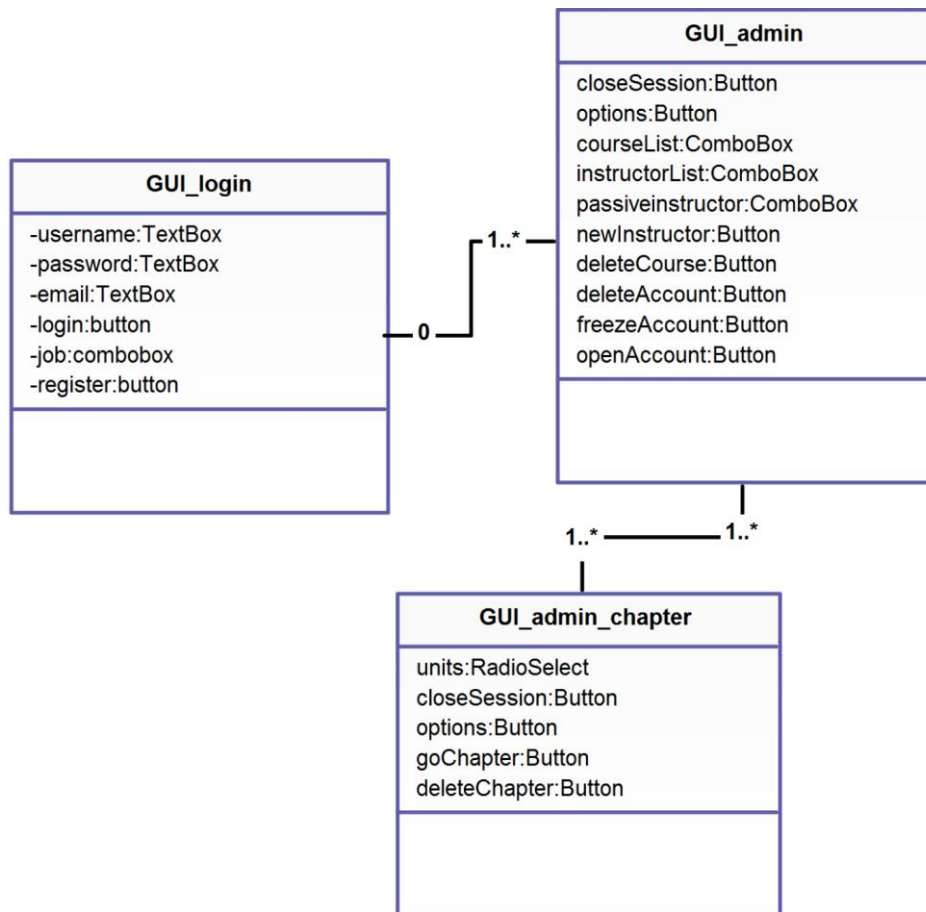
**Interface/Exports:** This package provides interaction interface for usage of user, data manager package and other components. User can get, modify or delete any necessity information, which is contained by system, through graphic user interface package depending on given permission.



**Fig36. Graphic User Interface Class Diagram1**



**Fig37. Graphic User Interface Class Diagram2**



**Fig38. Graphic User Interface Class Diagram3**

### 7.3 DATA MANAGER

**Classification:** Package

**Definition:** Data manager is a connector of front application and back application. Front application means displayed part of software that can be seen by user and back application means necessity data information that cannot be seen by user.

**Responsibilities:** Data manager provide database for software and other package of software can reach this data storage through data manager.

**Constraints:** Data manager includes server therefore, server constraints are valid. These constraints are limited number of connection at the same time, database crash and server computer. Server computer also has processor, memory limit such as each user creates profile on server computer and if its memory is exceed then it does not work and become unavailable.

**Composition:** Data manager has two sub-packages. They are data retrieval and "kaltura.com". Also this includes a subsystem which is database.

**Uses/Interactions:** Data manager is the base package. Other all packages get or set data information by using data manager. Especially, graphic user interface package send data information to data manager through user package. User package has relationship with data manager directly. User package tell data manager what it wants then get necessity data information directly from data manager.

**Resources:** Data manager needs absolutely more memory size. Since, data manager stores all data information to memory. On the other hand, data manager needs better processor to make query transaction over the database quickly. Therefore, data manager can give response so little time.

**Processing:** Data manager handle all processes by own algorithm. Data retrieval has own function which are well prepared. On the other hand, database works on by own and provides functions. "kaltura.com" is also works on its own and provides only functions.

**Interface/Exports:** Data manager provides and stores necessity data information.

---

### 7.3.1 DATA RETRIEVAL

---

**Classification:** Class

**Definition:** Data retrieval is class which extract necessity data information from database and "kaltura.com". Also this puts data information to database and "kaltura.com". In other words, this class is bridge for database and other packages.

**Responsibilities:** Data retrieval class separates desired data information to send database or "kaltura.com". Data retrieval gets order from other packages and converts appropriate information that is understood by database or "kaltura.com" easily. After that, if database or "kaltura.com" returns necessity information then data retrieval transfers data information to package.

**Constraints:** data retrieval needs connection to database. In order to use data retrieval, there should be opened connection. This connection can be time-out. Therefore, this data retrieval class time constraint can be woke up short time interval. This sub-package includes connection object as global variable.

**Composition:** Data retrieval is sub-package and it is one of the minor units of system.

**Uses/Interactions:** Data retrieval gets data information from graphic user interface package, user package and database.

**Resources:** Data retrieval class needs no more memory or processor. This class uses so much little memory and processor. Since, this class gets data information, converts appropriate data information and directly sends converted data information desired database. There is so little memory consumption because data information does not wait or is not buffered.

**Processing:** Data retrieval gets some orders from packages. Then it opens connection to database. After connection, gets necessity data information from package and sends data information to database.

**Interface/Exports:** Data retrieval class uses ADO.NET services and its function. Data retrieval class can connect database through using ADO.NET functions. Also data retrieval class gets data information from database through using ADO.NET functions.

---

### 7.3.2 DATABASE

---

**Classification:** Database is a subsystem.

**Definition:** Database is a place where data information is stored.

**Responsibilities:** Database stores desired information to memory using efficient storage planning. On the other hand, database provides data information to other package. Database hashes data information as reach so fast when data information is needed. Databases uses tree structure to store data information. Database includes database management system inside. Database management system interprets transactions which organize data information.

**Constraints:** Database has some constraints such as timing. Database management system which works in database, there is some time limit for queries. This query can be very complex and it takes long time collect data information.

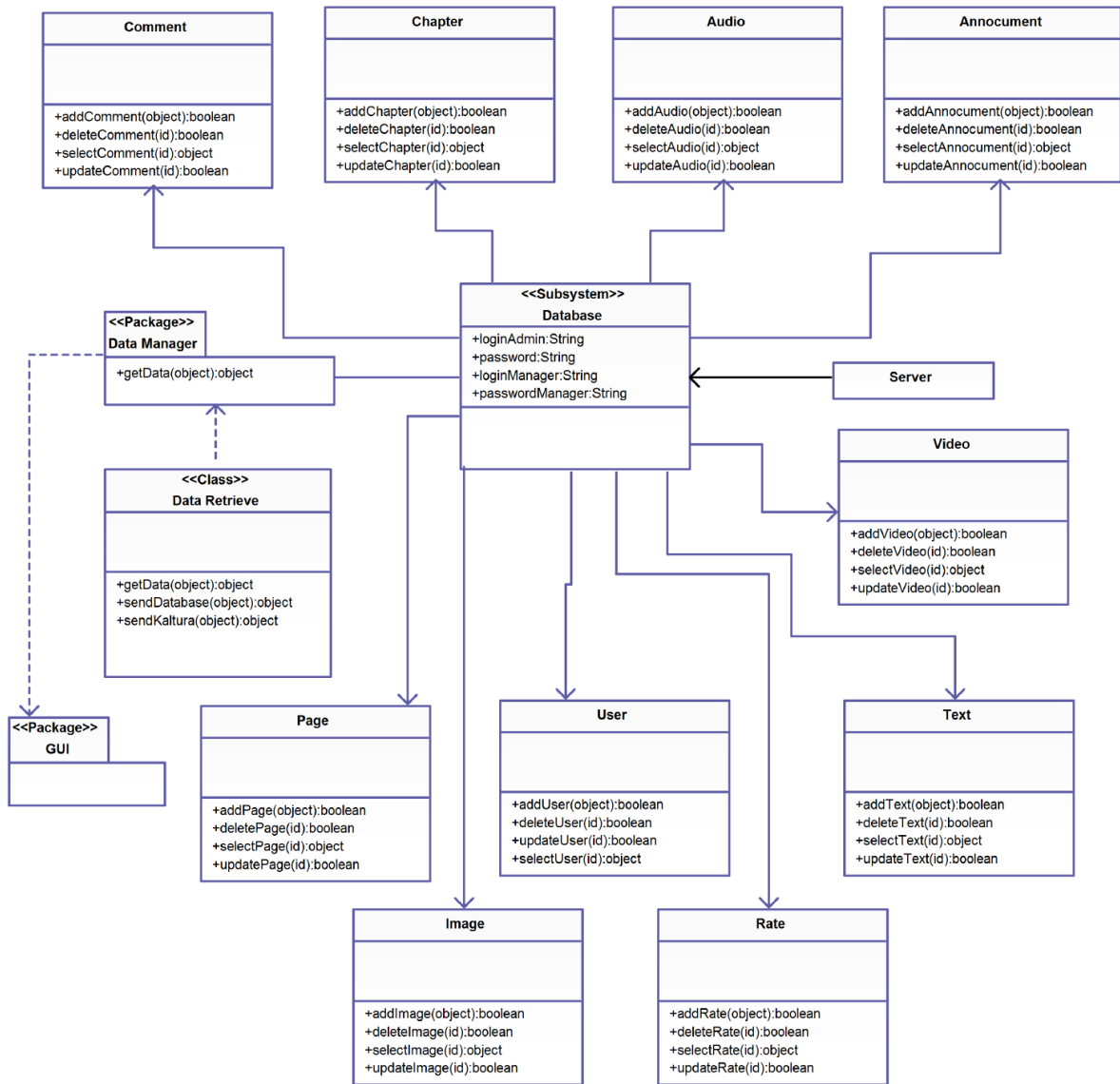
**Composition:** This subsystem is composed by data storage part and database management system. They are main systems in each.

**Uses/Interactions:** This subsystem interacts with other packages through data retrieval class. Database management system gets data information and order from data retrieval. Also database management system retrieves data information from database to data retrieval.

**Resources:** Database uses memory and processor. Memory usage is to store data information and processor is to process transaction from database management system.

**Processing:** Database manipulates data information and instantiation of new class object.

**Interface/Exports:** Database uses MSSQL database management service to manipulate data information.



**Fig39. Data Manager Class Diagram**

## 8 LIBRARIES AND TOOLS

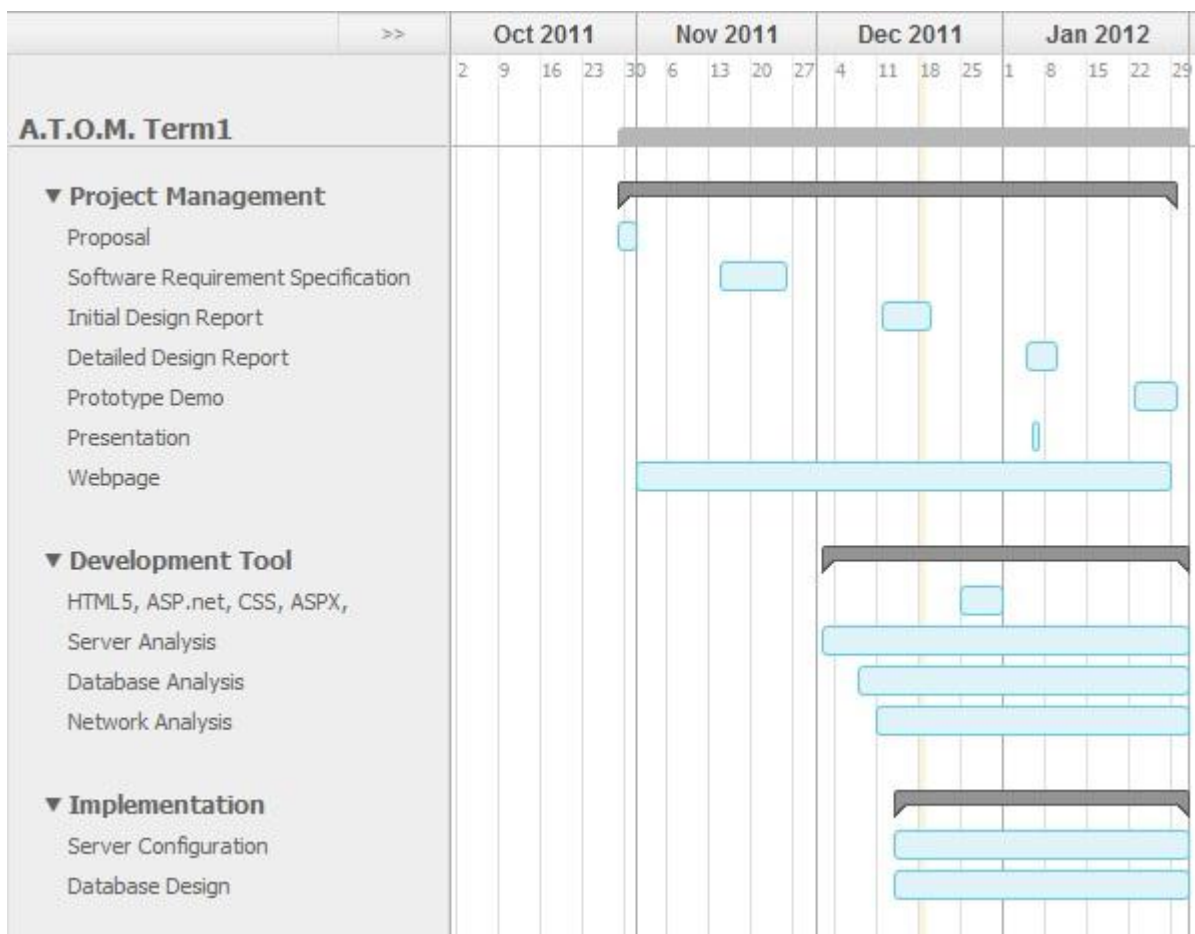
---

- **UML:** The Unified Modeling Language (UML) is a standard language for specifying, visualizing, constructing, and documenting the artifacts of software systems, as well as for business modeling and other non-software systems. The UML represents a collection of best engineering practices that have proven successful in the modeling of large and complex systems [3].
- **MSSQL:** Microsoft SQL is a software product whose primary function is to store and retrieve data as requested by other software application. In shortly; this software provides database to store necessity information [7].
- **MS VISUAL STUDIO:** This is an integrated development environment from Microsoft. It is used to develop web services. This software works on compatible with MSSQL. This software's code editor support ASP.NET. Also this includes database schema designer. Integrated debugger works both as a source-level debugger and a machine-level debugger is a plus [8].
- **ASP.NET:** This is web application framework which is supported from .NET family. This allows programmers to build dynamic web sites and web applications. This framework contains libraries to make connection and run queries on MSSQL database [9].
- **KALTURA:** This is web site. This provides to store videos and publish desired video as desired quality, size etc. This support also JAVASCRIPT language to send and receive information about video. This website allows getting any information about video.
- **HTML5:** This is a markup language for structuring and presenting content for World Wide Web. This provides some syntactical features such as "<video>", "<audio>","<object>". This is supported from all web browsers [10].
- **JAVASCRIPT:** This is a prototype-based scripting language that is dynamic, weakly typed and has first-class functions. It is a multi-paradigm language, supporting object-oriented, imperative, and functional programming styles. This is supported from almost all web languages. This is primarily used in the form of client-side in order to provide enhanced user interfaces and dynamic websites [11].
- **CSS:** Cascading Style Sheet is a style sheet language used to describe the presentation semantics (the look and formatting) of a document written in a markup language. It's most common application is to style web pages written in HTML and XHTML. CSS is designed primarily to enable the separation of document content (written in HTML or a similar markup language) from document presentation, including elements such as the layout, colors, and fonts. This separation can improve content accessibility, provide more flexibility and control in the specification of presentation characteristics, enable multiple pages to share formatting, and reduce complexity and repetition in the structural content (such as by allowing for web design without using table). CSS can also allow the same markup page to be presented in different styles for different rendering methods [6].

## 9 TIME PLANNING

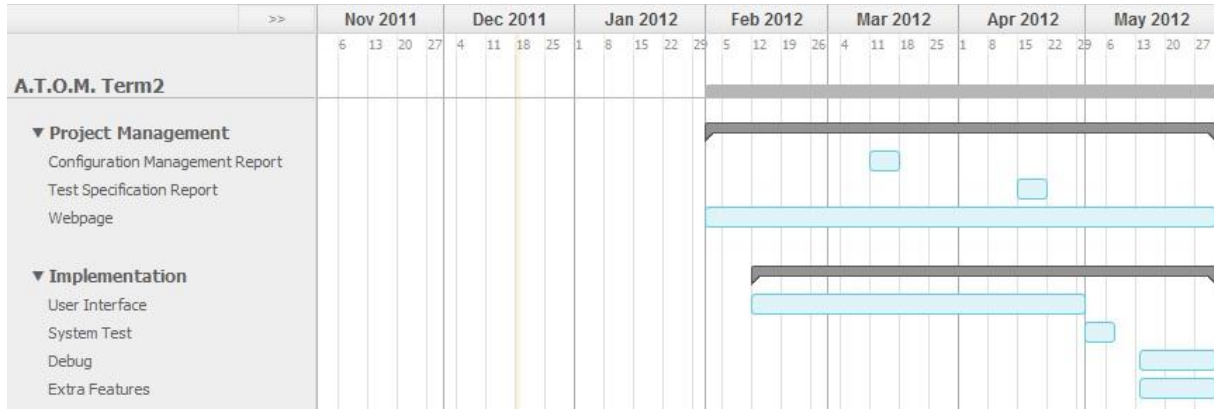
This section contains two Gantt Chart to illustrate approximate schedule of the project at this term and next term.

### 9.1 TERM 1 GANTT CHART



*Fig40. Term1 Gantt Chart*

## 9.2 TERM 2 GANTT CHART



**Fig41. Term2 Gantt Chart**

## 10 CONCLUSION

This detailed design report document gives detailed design information about the project, Mobile Content Generator, which is web-based software that simplifies forming e-learning material for instructors and enables learners to reach the material through different platforms. At the beginning of this document, basic introduction of entire design is provided. Afterwards, general description of the software and assumptions, dependencies, constraints, goals and guidelines are clarified. In the data design part, data domain organization of the system is presented. Decomposition of the system to its components has been presented in system architecture section. Next, user interfaces and detailed design of the project is explained. Finally, libraries and tool and time planning of the project are provided.