UDC: 378.046-021.68:001.5]:[004.63:004.774
Professional paper

E-THESIS REPOSITORY SYSTEM

Enes Bajrami[1], Eip Rufati[1], Florim Idrizi[1], Ermira Memeti[1], Lëndina Rushani[1]

[1] *Department of Informatics, Faculty of Natural Sciences and Mathematics, University of Tetova, North Macedonia*
**Corresponding author e-mail: e.bajrami3202009@unite.edu.mk*

Abstract

Digital services have played a significant role in changing processes, replacing manual with digital ones to provide fast and easy services to students and researchers. e-Thesis Repository System presents a web application built in C # programming language, ASP.NET web framework, HTML, CSS, JavaScript, Bootstrap and SQL for database creation. The main focus of the web application is the storage of three study cycles theses in a database, where students will have the opportunity to obtain study materials, and the administration will have easier access to statistics of students who have graduated. The web application allows any user to have access to the published theses. The Web Application has two types of user interfaces such as Administrator and Professor. The administrator has full access to the web application where he can create accounts, update and delete. Also, the administrator can set the thesis, update or delete it, while the professor can only set the thesis, update and change the personal data. For the development of the web application Microsoft Visual Studio was utilized, along with SQL to build the database, which represents an integrated application.

***Keywords*:** e-Thesis, system, web application, database, administrator, profesor

1. Introduction

Thesis Repository System characterizes an electronic deposit system which consists of information such as concepts and research results of researchers during their process of recognition as well as exploration and analysis. One of the most effective steps for higher education institutions to increase efficiency in scientific papers such as diploma, master and doctoral theses is the transition from print to electronic systems [1]. Some universities have already developed such systems. Many universities around the world have made applications for placing doctoral theses in their universities available to the campus community and the general public through an electronic system [2,3]. To keep up with technological innovations, these various institutions need to invest in the development of various applications which will be the catalyst for success and rapid access to the right information. Most Universities have such a system for storing theses in a secure database. Given that proposal idea for this web application, we know that University of Tetova has a large number of students and over time is making the replacement of manual processes with digital ones in order to provide students with fast and easy services. Anyway, it is inevitable that such a system will facilitate the work of students who would find it very easy to enrich themselves with information and paper works of graduates at the University of Tetova.

2. The purpose of the web application

The most challenging aspect of any system is precisely identifying goals. Clear set goals are a must, since they define the range of features that have to be implemented. There are a number of goals which our application is inalienable to meet and among the most important are: Facilitates the work of administrators who have a clear overview of how the graduation processes of the students are, since it facilitates their work, and through the system they are enriched with complete and accurate information. Undoubtedly, an application is successful if it manages to meet certain criteria, to be part of the market as fast as possible, simultaneously to be qualitative and affordable. The next thing the paper aims to achieve with this web application, is to meet these and many other criteria, in order fulfill all the expectations and requirements of the clients.

3. Web Application Requirements

The initial requirements for the implementation of the system, include a system development team and clearly identifying the technologies that will be used during the implementation of the same. The main limitations usually come when using the system. Considering that the system will be able to be utilized by people with different backgrounds and profiles, it has been essential fact that the system interface for users is as "user-friendly" as possible, as such a thing is always required. Then, we all want to develop and design successful web applications, but for that there are a considerable number of factors that define this success. For example, customers want to know different aspects of the products an offer is made, such as their price, design, product value, etc. As mentioned above, the system has an attractive and very useful design. Another requirement of the system is its own performance. According to research and studies conducted, it has been found that somewhere around 49% of customers expect a website to open in 2 seconds or less, while the rest decide to leave the website which takes more than 3 seconds to open. Therefore, it is very important for us that users do not show poor performance both when opening and navigating and using the web application. In general, system speed plays the biggest role for a successful web application and slow web applications are considered as failed. Some of the reasons that result in low performance may be: Poor coding, non-optimized database, very high quality and non-optimized files, low web server services, etc. During the development of the system, each of these aspects has been treated with special effort and with the most optimal coding, so that the performance of the system (dependent on these aspects and not on the host), is always at a satisfactory level. Another very welcome request is the delivery on time, considering and considering in advance any problems that may occur and cause delays. Let us not forget that the system we offer has integrated security, is reliable to use in understanding the entry of personal data, is very useful and efficient, offers great flexibility, is qualitative and what is most importantly, it is innovative at our university.

Non-functional System Requirements - When we talk about the non-functional requirements of our system, it is inalienable to divide them into 2 separate groups:

- Performance Requirement
  - Response Time – System Respond within 2 seconds
  - Capacity – System has ability to include unlimited numbers of users

- Interface – Appears after 2 seconds

- Safety Requirements – Security is one of the basic requirements of the system and any emergency can occur, such as the database may crash for a certain time, operating system failure, etc. Therefore, it is preferable to make a backup every time.

4. Use Case Diagram

A Use Case diagram is a visual description of the various use case interaction scenarios. The usefulness of use case diagrams is more based on the communication between the request team and the user group. A use case specification document should cover these areas:

- Actors - interact and participate in this use case
- Preconditions - must be met for the purpose of use case action
- Unconditionals - define the various states in which the system is expected to be after it is executed. The Use Case diagram lists the basic events that will occur when the system is executed. It includes all the primary actions that the system must perform.

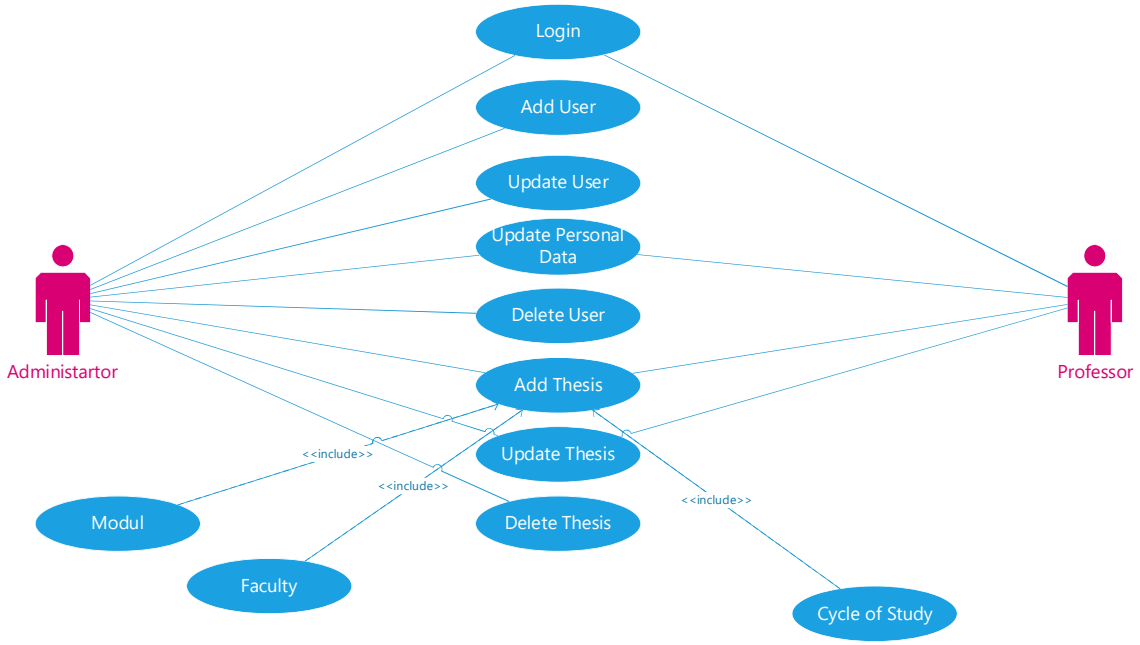

Diagram 1. Use Case Diagram

The diagram presented above shows the actors and processes performed by the system. As actors we have the administrator and the professor, and the processes presented within the ellipse shapes are the functions which can be performed depending on the roles provided. Activity Diagram: Activity diagrams signify a presentation in the form of diagrams, maintaining therefore a hierarchy of activities. These diagrams can be important for exploring the flow of activities in a Use Case, or between Use Cases. Activities are states of action that automatically switch to another state after the action has ended. The activity diagram contains several elements: The filled circles indicate the beginning of the diagram where the control flow begins, the

transitions represented by arrows show how we move from one activity to another, and the synchronized bars show how the activities occur in parallel timespan.

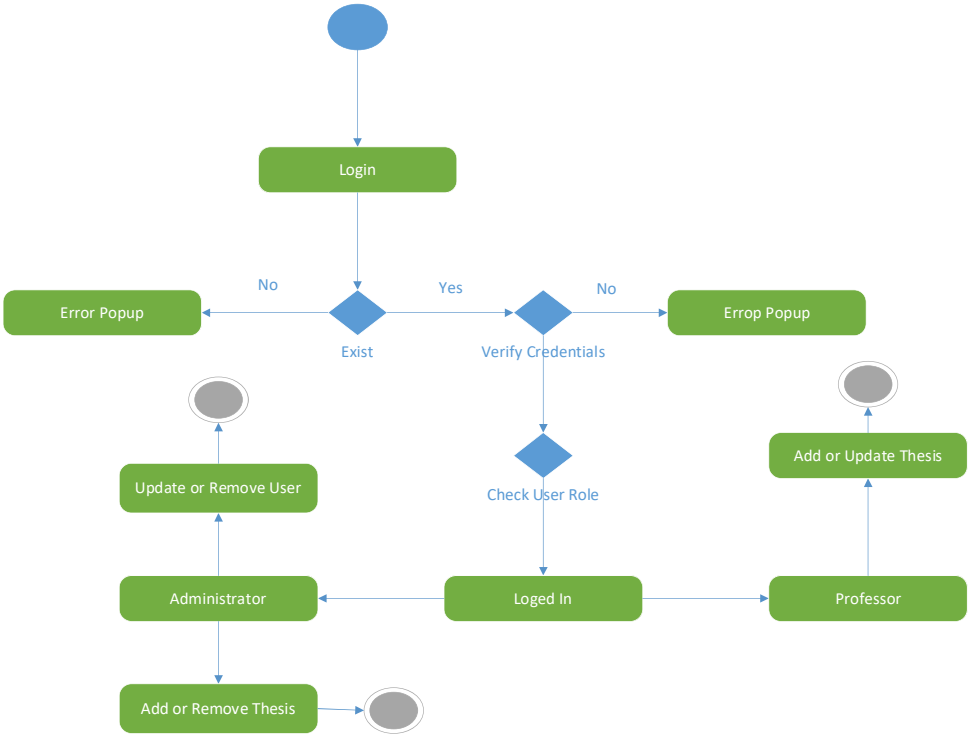

Diagram 2. Activity Diagram

In the Diagram of Activities, we have presented graphically how the processes flow, credentials are entered first, if they match the database, the role of the user where the connection to the web application is made is also checked. Depending on the user role, the displayed interface will change along with the operations they can perform.

State Diagrams: also called State Chart diagrams, are used to help developers better understand any complex or unusual functions and business flows of specialized areas of the system. In short terms, State diagrams record the dynamic behavior of an entire system, or a subsystem, or even a single object in a system. This is done with the help of behavioral elements. It is important to note that having a Status diagram for the system is not an obligation but it should only be determined based on the previously defined needs. The state diagram is represented by a rectangle with a rounded edge, and inside the diagram are written the necessary data for the appropriate software or web application in which the developers are working. From the terminology of the diagram in question, we can conclude that it defines the different states of the participants in the action or use of an application, in our concrete case the application for managing certain people information.

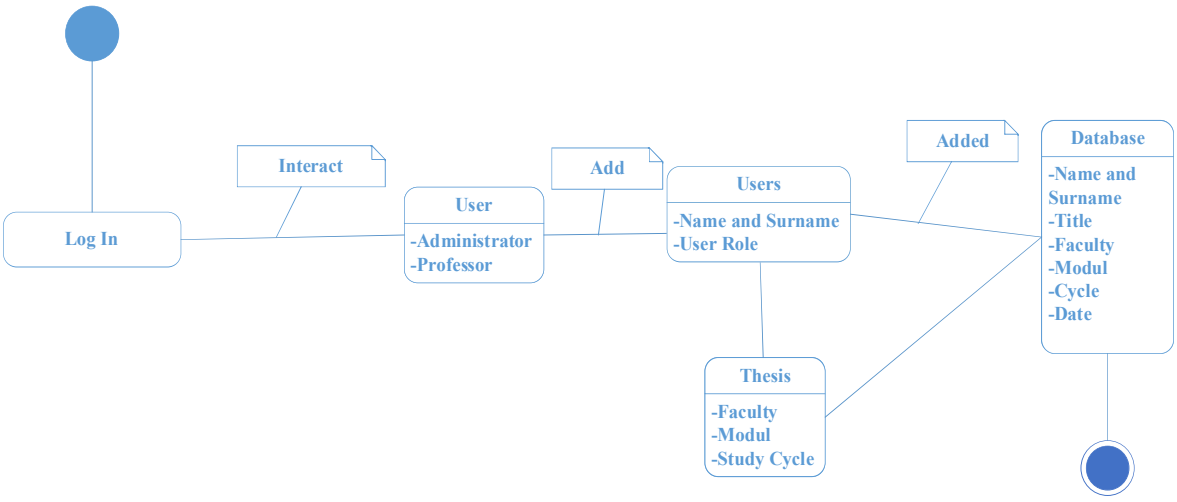

Diagram 3. State Diagram

Above we have presented the state diagram which presents the initial processes of the system, starting from the starting point which presents the connection to the system, then continues with user interaction where the system via SQL command detects the role of the user, then adds of a thesis according to the faculty, direction and study cycle, where the activity ends. All these one-on-one activities as we see each other are connected via arrows.

Class Diagram: The Class diagram describes the structure of the system by presenting the system classes, its attributes, operations (or methods), and the relationships between objects. The class diagram is the main object-oriented modeling block. Used for general conceptual modeling of application structure and for detailed modeling of model translation in programming code. Class diagrams can also be used to model the data. The classes in a class diagram represent both the key elements, the interactions in the application, and the classes to be programmed.

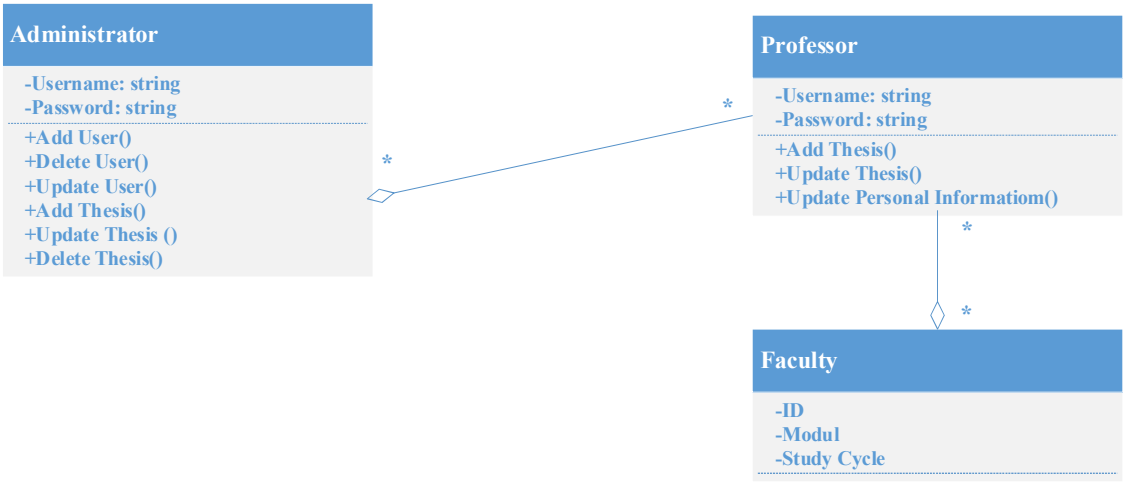

Diagram 4. Class Diagram

Above it is presented the class diagram, where the relation we have used is generalization. Each of the classes is made up of names, attributes, and operations. In a class diagram, we cannot have classes that will be placed around as single, since it is necessary to see the relationships between them. Therefore, with the generalization relation, the administrator and professor classes are related, where many professors can be administrators, in the other faculty class, we have generalized the same relation, since many professors can conduct lectures in several faculties.
Object Diagram: Object diagrams are derived from class diagrams, therefore the object diagrams depend on class diagrams. Overall, object diagrams represent an example of a class diagram. The basic concepts are similar for class diagrams and object diagrams. Object diagrams also represent the static view of a system, but this static view is a picture of the system at a particular moment. Object diagrams are used to give an array of objects and their relationships as an example.

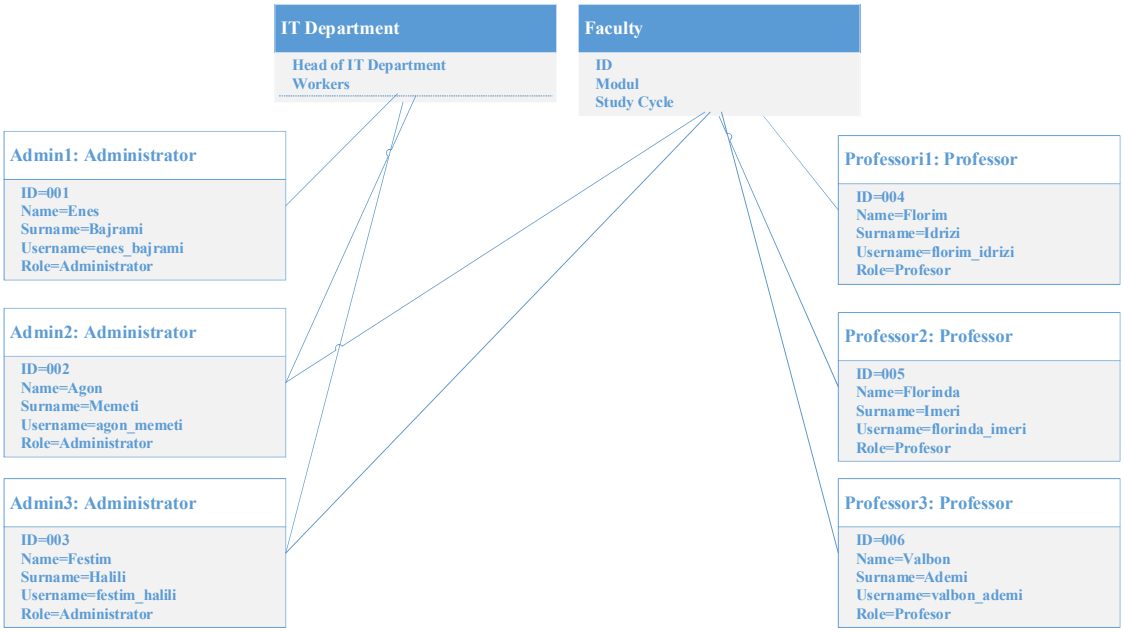

Diagram *3*. Object Diagram

Component Diagram: The component diagrams present the code module. They are the physical analogues of class diagrams. The main point that distinguishes component diagrams from other diagrams is that component diagrams represent the implementation perspective of the system build. Therefore, the components in a such diagram are reflected by grouping different designs of system elements, such as system classes. Firstly the component must be replaceable, and secondly it must provide interfaces to enable other elements to interact and provide the services provided by the component.

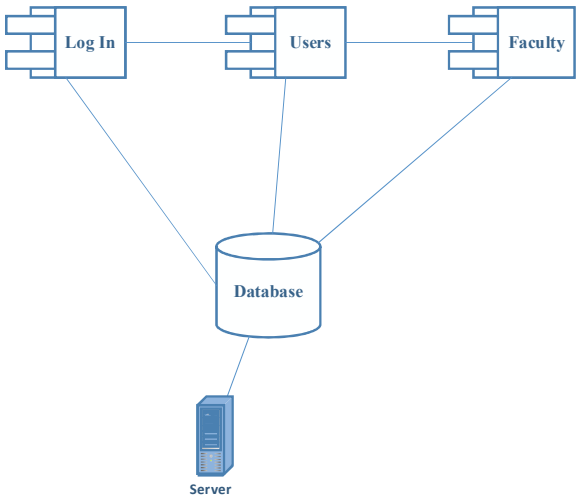

Diagram 4. Component Diagram

The diagram above shows how components such as administrator, professor, faculty, study program, and study cycle interact with each other and also with the database, whereas the database communicates with the web server.
Deployment Diagram: Such kind of diagrams are used to visualize the topology of the physical components of a system, where the software components are located. Distribution diagrams are used to describe the static layout of a system. Placement diagrams consist of nodes and their relationships.

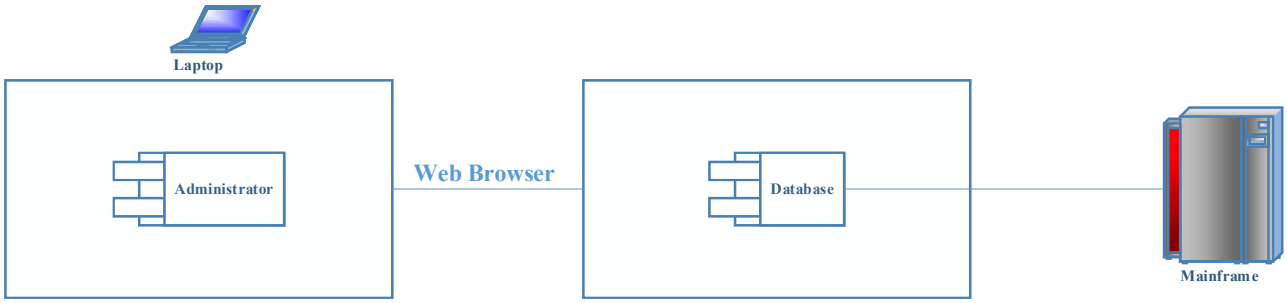

Diagram 7. Deployment Diagram

In the diagram 7 shown above we have shown how the web application is visually located on a server.

Interaction diagram: Interaction diagrams model the behavior of use cases by describing how groups of objects interact to complete the task. Two types of Interaction diagrams are: sequential diagrams and collaboration diagrams.

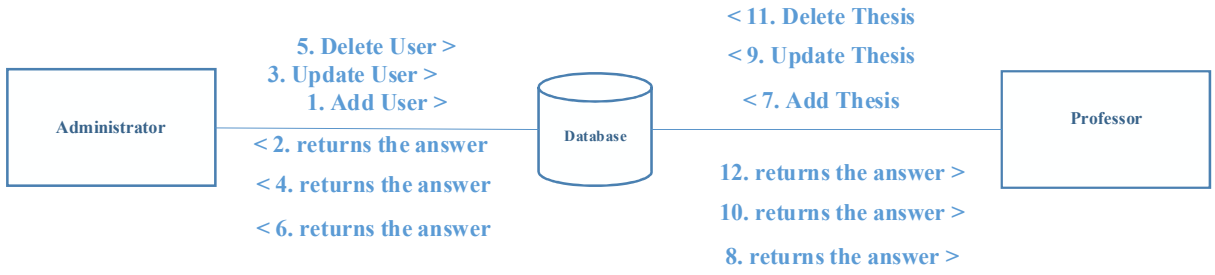

Diagram 8. Interaction Diagram

In the Interaction Diagram we have shown how a group of objects interacts with the database, the interaction can be both, a single user and many users at the same time.

5. Software project management – CPM

A project represents a series of activities directed to the fulfillment of a desired goal. CPM can be used to improve the management as well as the operations of a system. In the CPM network, the whole project consists of a number of known operations, called activities. Activities are usually operations that we need time to accomplish. Intersections that occur between activities are known as events. CPM networks are often referred to as activity-oriented diagrams, in which activities

are represented by an arrow, and the sequence in which the activities are performed is represented by the arrow sequence. CPM networks are usually used for types of repetitive projects, or for those projects that have a fairly accurate estimate of the completion time for which each activity will be ready.

CPM helps us to:

• How long will the project take in order to finish

• Which activities are "critical" which means that they must be done on time, otherwise the whole project will be delayed.

Below we have divided the necessary activities for the web application and for each activity is determined the time needed to complete, and through this data the construction of CPM is reached:

Table 1: Project Activities

Activity		Predecessor Activity	Duration (day)	Optimist Time	Pessimist Time	Expected Value
A	Analysis of requirements	/	1	0.5	2	1.08
B	Presentation of the plan	/	2	1	3	2
C	Presentation of UML diagrams	B	2	1	3	2
E	Data Entry	A	2	1	3	2
F	Creating Files	E	2	1	3	2
G	Application update	F	2	1	3	2
H	Note the specifics of the application	A	2	1	3	2
I	Application construction	H,A	15	7	10	12.83
J	System testing	D,F,I	2	0.5	3	1.93
K	Web Server Deployment	I	1	1	1.5	1.08
L	Training of professors	K	10	8	14	10.33
M	Software submission	L,J,G	1	0.5	1	0.91
			44	24.5	52.5	42.16

Critical Path: A-H-I-J-M (1+2+15+2+1) = 21 day
Optimistic Time: 24.5

Pessimistic Time: 52.5

Expected Value: 42.16

6. Database Design and Modeling

Database modeling is an important step in designing the database, serving as a bridge between the real world and what is inside the computer.
Database design focuses on how the database structure will be utilized to store and manage end-user data. Data modeling, which is presented as the first step in data design, refers to the process

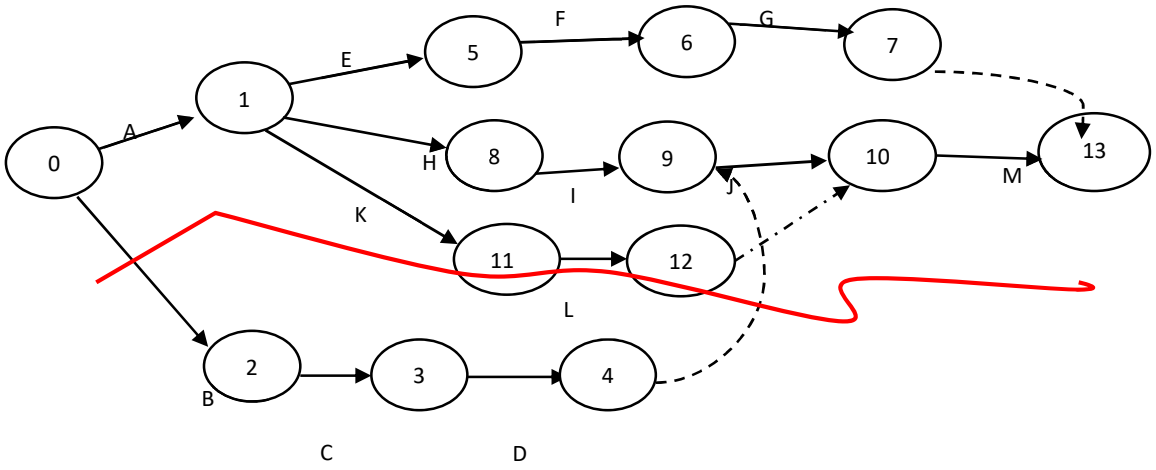

Diagram 5. CPM Diagram

of creating a specific data model for a given problem. The data model is a simple representation, often times presented graphically for complex data. The database of our web application is entitled e-Thesis and includes 7 tables which we have displayed below:

E-R Diagram - An ER-diagram represents the data model. Standard symbols that are familiar to everyone are used to design the database using the ER-diagram. Most ER diagrams use squares, connecting lines, and diamonds. Squares represent entities, connecting lines represent relationships, and diamonds are used to indicate relationship characteristics. Our application is built with ASP.NET and we have created the database in SQL, because in our web application we also insert files we have used LINQ to SQL. LINQ in SQL is a component of the .NET Framework version 3.5 that provides an infrastructure for managing relational data as objects. When the

application is executed, LINQ in SQL translates into SQL the queries integrated into the language in the object model and sends them to the database for execution.

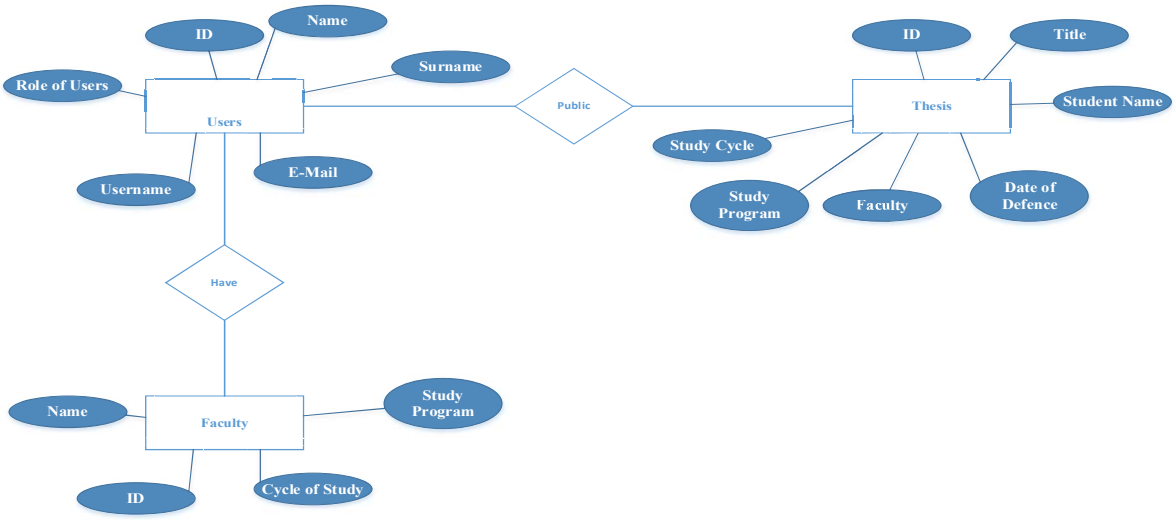

Diagram 6. E-R diagram

7. Application development

The web application presented in this paper is built with Visual Studio 2019 software and C# programming language, whereas the utilized web framework is ASP.NET. It is important to note that the interface design of this web application is "user friendly" since during the work we have taken into account the fact that it can be understood by any user, i.e., by users with different backgrounds. When the application launches, the interface shown the following figure is displayed, consisting of the main menu, where all the faculties are listed.

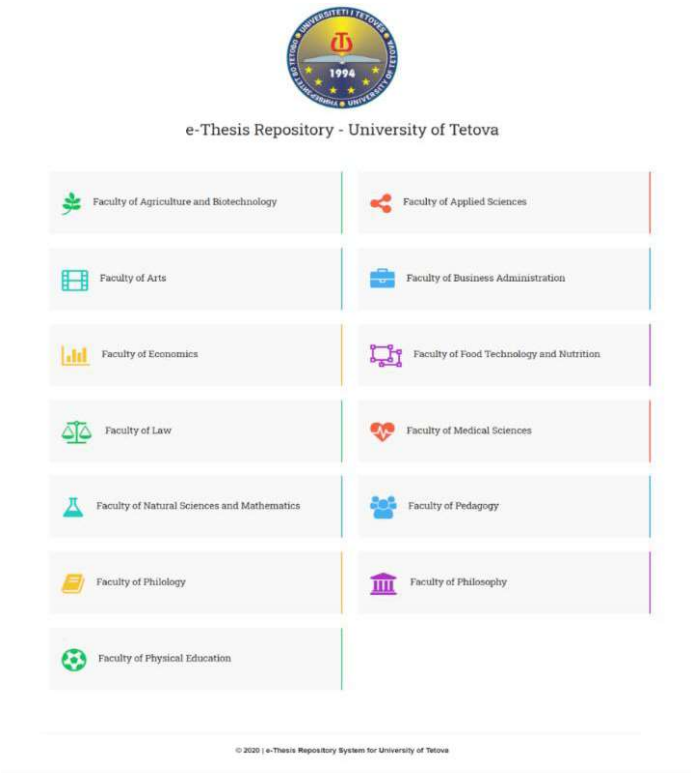

Figure 1: Home Page

6. Conclusions

The system is successful if it manages to meet all the requirements of the users, in this case students and professors, as well as being able to access and receive the right information at any time.
The system presented above has integrated security, is reliable to be used in the sense of entering personal data, is very useful and efficient, offers great flexibility, is qualitative and what is most important, it presents an innovative approach at our university.

References

[1]. McCutcheon, S. Basic, fuller, fullest: Treatment options for electronic theses and dissertations. *Library Collections, Acquisitions, & Technical Services, 35* (2-3), 64–68 (2011).

[2]. Levy F., Pyles R., Szarejko C.,& Wyatt L, Developing an electronic repository for undergraduate theses. *Honors in Practice – Online Archive.* Paper 160, (2012).

[3]. Sheeja, N.K. (2011). The development of an Indian electronic theses and dissertations repository: An overview