## Chapter 1. Introduction.

The main idea of this diploma thesis is to bring together the e-learning concept and livestream in order to simulate the exact conditions of a classroom. This project wants to help students that follow a distance studies program, by allowing them to interact with their teacher. Also it will offer teachers who are travelling a lot the possibility to keep their curse even if they are on the other half of the world. An important thing is that usability of this platform may extend without modifications of the source code because even if the main purpose is to serve students and teachers, it can be used for any webinar or training session.

Beside the main feature, which is livestreaming, there are a set of extra features that make this platform more attractive. An important feature is allowing a questions and answers session, where students can put some questions which can be either private, only with the teacher, or in the chat room made with all students that have subscribed for the current curse. To this questions it will be teacher’s choice when to answer to that question, he can respond instantly, or when he thinks is properly to do it.

A feature that is useful for students is that curses are also persisted and they can be replayed any time the user wants. This feature may come in handy when the user wants to recapitulate things that have been discussed at the curse or they cannot be online when the curse is kept. In this case the chat feature will not be available.

In order to increase interactivity of the curse beside the chat, the teacher can initiate a quick quiz, which can have a time limit per question or per entire quiz. The students will not be able to make anything else with the application during a test. The test will be structured as a multiple choice and will be automatically corrected when the user will finish it or when the time expires. Only the teacher will be able to see the results of the test.

In order to keep track of the users that are present at a certain curse, the application will be able mark as present both users that are connected remote to the curse and those who have a smart phone with the mobile application for remote learning installed on it. This information can be used to send quizzes or in statistical purposes.

Another thing worth mentioning is that the platform will contain a Windows Phone 8.1 and a Windows 8.1 application, which will offer students a user-friendly experience, an application on the same platforms for the teacher and a server which will serve applications with requested data.

This chapter will also make a resume for every theme that will be covered by this thesis.

The entire implementation is based on .NET framework, so it is compulsory to talk in the first part after the introduction about this framework considering that all chapters that will follow this one will be strongly related to it. I will try to make a brief presentation of the main concepts that this framework facilitates.

The next chapter, named Streaming, will present the technologies and frameworks used in order to create the livestream environment and also how video data will be persisted. It will explain the smooth streaming concept, will offer an overview of the software development kit that enforces this concept and will discuss about encoding and types of encoding required by smooth streaming. Also this division will refer to Internet Information Services and Media Services extra feature needed in order to make livestream and on demand stream available.

In the chapter Universal Windows Applications we will discuss a little bit about the new concept that Microsoft has brought into developers world. I am referring to the fact that you can create an application that can run on phones, tables, laptops and personal computers. Also it will contain some explanations regarding Microsoft Media Platform's Player Framework and Smooth Streaming Client Software Development Kit. Near field communication will be another theme discussed because it is used to signal the presence of students at a certain curse.

The next chapter is about persistence, relational databases and Microsoft SQL Server 2012, which will be used as database management system. It will explain some terms as store procedure and discuss how SQL injection is avoided using their parameterized version.

Another chapter will refer to web services in order to make persisted data accessible. Here we will talk about Simple Object Access Protocol (SOAP), its benefits and downsides, also about how they are implemented in .NET framework.

If until now we have presented all the technologies used in order to create the application, in the next part a detailed overview of it will be made. Here we will talk about architecture, detailed feature explanation and implementation details. In this chapter will exist the use case diagram of the application and each use case will be explained. For each use case will exist a sequence diagram which will show the flow of data and the execution flow. Also will exist a class diagram which will help as observe what design patterns have been used and make an idea how applications are implemented. A database diagram will be added in order to easily understand how data is persisted. Main algorithms will also be presented in this area and will be explained.

The last chapter will contain the conclusions over this thesis.

There are a lot of e-learning platforms, which offer functionalities similar with Traveler Student. As a first example let’s look into Harvard Extension School, which is a project that offers videotaped, live web-conferences and hybrid courses. A big advantage is that people from all around the world can join this classes without travelling to the campus. A difference between their project and Traveler Student is the quiz feature, which provides a way to examine students and also to verify the attention to the current course.

Another online courses provider is Pluralsight. This is a web only platform that offers videotaped lectures for developers and IT admins. They have a very large number of courses and they cover almost all technologies and frameworks. The interesting part is that videos are based on examples and cover details with respect to the level that has been assigned to that course. A nice feature is that they mark the videos that you have seen, an idea that Traveler Student will inherit and improve it by saving the courses you have attended from the classroom.

## Chapter 2. .NET Framework and C# Language.

### 2.1 .NET Framework

.Net Framework is a technology that supports building and running applications of many types and XML Web Services. It can be split in two big parts: first one is the common language runtime and the second one is class library.

#### 2.1.1 Common Language runtime

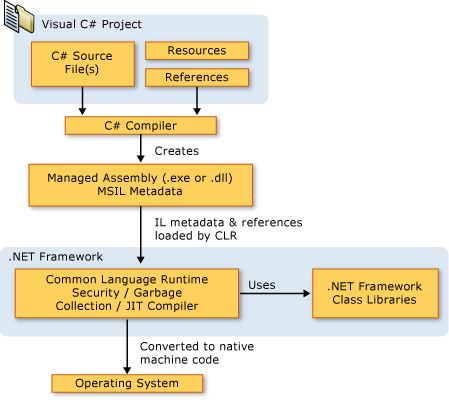
We can say that Common language runtime is the core component of the framework. It manages memory, thread execution, code execution, code safety verification, compilation and other services. It supports many languages, for example: C++, Visual Basic and F#, but we will discuss more about C#, given that all applications developed for this thesis are written in this language. In other words this component takes code which comes as Microsoft Intermediate Language (MSIL) metadata and transforms it into machine code that will be executed by the processor. The figure below shows the lifecycle of the source code.

Figure 1 [https://msdn.microsoft.com/en-us/library/z1zx9t92.aspx]

An important feature of the above mentioned component is memory management, also known as Garbage Collector. It is used in order to identify and free unused allocated memory. Garbage collection is a very expensive operation which, if is made at the wrong moment, can decrease drastically the performance of a program. So in order to avoid that there are three situations in which it is realized: when the system is low on physical memory, when the allocated memory surpasses a certain limit, which is adjusted as the process runs and when the GC.Collect method is called, situation which is totally unadvised.

#### 2.1.2 Class Library

.NET Framework includes, as you can see in the schema above, has a component, named Class Library, which includes a set of standard libraries. They are organized, hierarchical, in namespaces. For most of the Application Programming Interfaces included in it the root namespaces are System and Microsoft. This library contains a large number of useful functions such as graphic rendering, file reading and writing, XML manipulation, and many others. Also it can be divided into two parts: Framework Class Library and Base Class Library.

Base Class Library is the main core part of the framework. It includes base types for objects and important components which allow Common Language Runtime to operate properly. It includes namespaces as: System, System. Collections, System. Collections.Generic, System.Diagnostics, System.Globalization, System.IO, System.Security, System.Security.Permissions and System.Text, System.Threading.

Framework Class Library includes an expanded set of libraries as Windows Forms, ADO.NET, ASP.NET, Language Integrated Query (LINQ), Windows Presentation Foundation (WPF), Windows Communication Foundation (WCF) and Workflow Foundation (WF). It is very vast, being comparable as dimension with Java standard libraries.

### 2.2 C# Language

C# is a C derived object-oriented language, which enforces strong typing. It supports all object oriented concepts as encapsulation, inheritance and polymorphism, but has some extra features that make it easier to develop software. It is Microsoft programming language developed to compete with Sun’s Java language.

Even if it is a language that is like C++ and Java there are some interesting, innovative and handful language constructs. Delegates is one of them, which are encapsulated method signatures, allowing type-safe event notifications. Another one is the existence of so named Properties which are members with getter and setter either default defined or customized. Attributes are much alike Java annotations and are used to provide declarative metadata about types at runtime. A very interesting feature is Language-Integrated Query (LINQ) which allows you to make SQL alike queries, but over System.Collections data structures.

If until now we have spoken about differences it is important to underline that Java and C# are very much alike. An important similarity is that both are using interfaces for multiple inheritance, which is different from C++ way things work. Another similitude is the fact that any variable and method have to be encapsulated in a class, in other words there are no global variables and methods.

## Chapter 3. Streaming media.

Streaming media refers to the process of constantly receiving multimedia from a provider and presenting it to the end-user. The difference between streaming multimedia and downloading it is that when you stream it you can play it before the entire file has been stored on your local drive.

The first patented system for transmission and distribution of signal over electrical lines was designed by Major General George Owen Squier the early 1920s. His patents were acquired by North American Company in 1922, which created Wired Radio Inc., a company that delivered music, charging for it on the electric bill.

Even if the technology was discovered early, streaming media over computer networks was not possible until June 24, 1993, when the band “Severe Tire Damage” performed live on the Internet for the first time.

Live streaming is a division of multimedia streaming, which refers to delivering live, over the internet, the data received from a source of media (for example a video camera or a voice recorder). The process is a bit more complex than it is presented in the above definition, considering that the content is encoded and several protocols are applied over it before it is distributed to the client, which will unpack, decode and then present it.

A big problem that concerns streaming media is that it has to handle big amount of data. An improvement that would reduce the dimension of data would be to compress it. Here intervene the so named codecs. A codec, in our case, is a computer program, which is concerned with encoding-decoding and compressing-decompressing data. There are two types of codecs: video and audio. For each one there are multiple compression formats. Since in our platform will use H.264 video compression format and AAC audio coding standard we will discuss a bit about them.

“H.264 or MPEG-4 Part 10, Advanced Video Coding (MPEG-4 AVC) is a block-oriented motion-compensation-based video compression standard developed by the ITU-T Video Coding Experts Group (VCEG) together with the ISO/IEC JTC1 Moving Picture Experts Group (MPEG)”( http://en.wikipedia.org/wiki/H.264/MPEG-4\_AVC).

…………………………………………………………………………..

Advanced Audio Coding (AAC)

…………………………………………………………………………….

### 3.1 Internet Information Services

Internet information Services is a web server produced by Microsoft

#### 3.1.1 Media Services

## Chapter 4. Universal Windows Applications.

## Chapter 5.Microsoft SQL Server.

## Chapter 6. Web services.

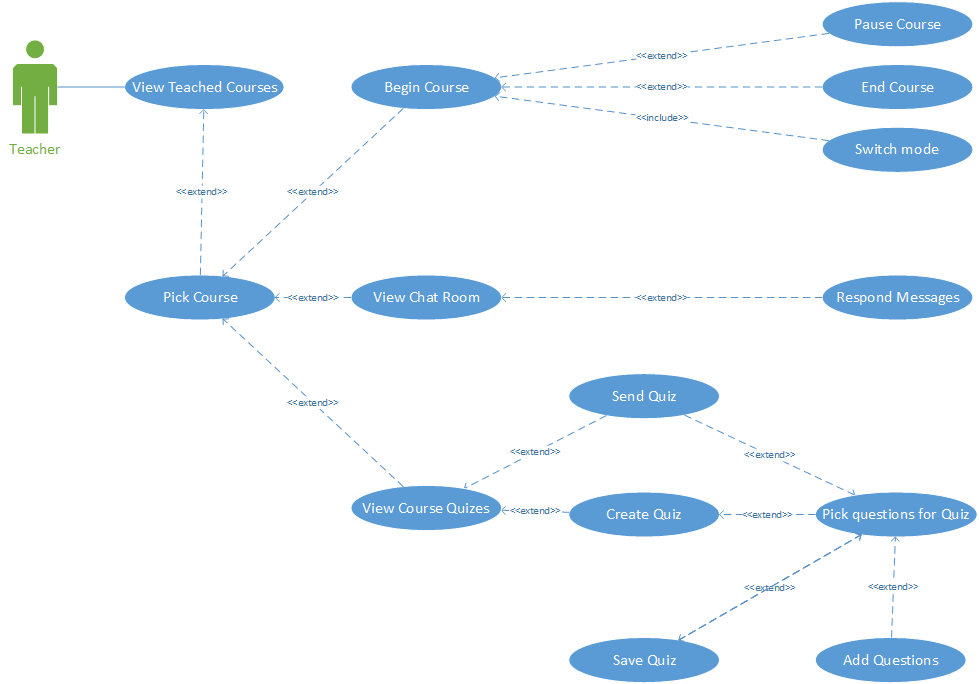
## Chapter 7. Traveler Student application.

As mentioned in the Introduction chapter, this chapter will discuss about TravelerStudent’s architecture details and use cases. A detailed presentation of how this platform can be used will be made first and then the parts about its core and code behind.

### 7.1 Use cases

This platform has three possible actors: Teacher, Student and Administrator. Teacher is the user that will present the course, that would interact with use cases like Quiz creation and course presentation. In order to make Use case diagrams more readable the LogIn use case has been excluded because this platform does not allow any anonymus operations. It should be included as the first use case for every actor.

#### 7.1.1 Teacher related use cases

The teacher is the user with the most use cases, so the diagram below offers an overview of all of them and the relations that they enforce.

##### View Taught Courses

After the user have logged in he will see all the courses that he is responsible for. This use case is necessary because a lecturer may preach more than one subject.

##### Pick Course

This actor should be able to choose the lecture that he wants to work with. There no other actions possible without selecting an item from the list. This action is enforced because there are no use cases that involve more than one course.

##### Begin Course

Represents the beginning of the broadcast, in other words the moment when the teacher starts the audio and video streaming from the course. In this moment all the students are enrolled for this subject will receive a notification that the course have begun.

##### Pause course

The lecturer should have the possibility to pause the transmission without ending the course.

##### End Course

This one gives the actor the possibility to stop the streaming and mark the end of the lesson. The broadcast will be stopped and student will be notified that current lecture has ended.

##### Switch mode

The teacher is able to stream images from the class room and from the desktop of the computer he uses. This use case allows him to change the source of the images.

##### View Chat Room

TravelerStudent offers a chat room for every course so the teacher is able to read those messages.

##### Respond Messages

The chat room is available when the transmission is off so the teacher must have a way to respond questions that are posted there. Here intervenes this use case which allow him to provide the information that the students requested. In order to keep an academic air no private messages are allowed.

##### View Course Quizzes

A professor is able to give students tests with this platform. This quizzes have been predefined by him and this use case offers the opportunity to view all of them.

##### Send Quiz

The actor can send the test to all users that have attended this course. This use case can be reached in two ways. One is by picking a test from the list and the other is by creating a new one and then sending it.

##### View All Questions

This functionality offers the possibility of viewing all questions available for the current course.

##### Add Questions

Adding a question assumes to enter the problem statement and all the choices that will be available. Considering that only multiple choice questions are allowed, the user also need to provide the correct answer. It is needed for the automatic correction of the tests. Another information requested is the average time that the teacher considers proper in order to answer to the requirement.

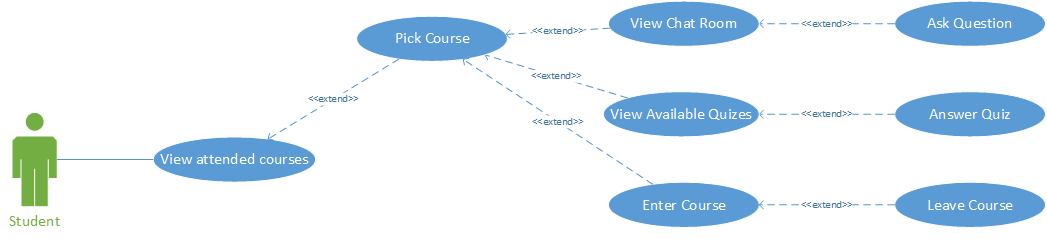
##### Pick Questions for Quiz

From the list with all the questions the lecturer has previously added he can add the ones that he considers fitted in. He can also pick the way time is counted: per question or per quiz as a sum of time limits from all questions.

##### Save Quiz

If the teacher intends to reuse the quiz he has just created, then he can save it.

#### 7.1.2 Student related use cases

 Another actor, the user which is going to use the application in order to watch and learn, does not have so many use cases. In a few words he is able only to watch, chat and answer quizzes.

##### View Attended Courses

A student should be able to see the list of courses in which he is enrolled, because it is easier to identify the class he wants to attend if he sees only those he joined.

##### Pick Course

Given the fact that a scholar can attend more than one courses all functionalities regarding one lecture will became available only when he has selected it from the list.

##### View Chat Room

This use case represents opening the chat room for a certain lecture.

##### Ask Question

This actor is also able to write in the chat room.

##### View Available Quizzes

If there are quizzes that the current logged in actor have not solved he will be able to see them here. The other functionality offered is that the user can see the results for previous tests at currently selected course here.

##### Answer Quiz

After the student selected the quiz he wants to take the test will begin. If he will leave the test or the time will expire the test will be corrected at current state.

##### Enter Course

The learner will begin watching the live images from the classroom.

##### Leave course

The student will stop watching live the course.

#### 7.1.3 Administrator related use cases

The third actor is the administrator of the system. He is the user that controls the way courses teachers and students are related.

##### Add Teacher

Adding a new teacher is a feature that becomes useful when new lecturers are employed or decide to join the cause.

##### Add Student

Considering that there are no scenarios which do not require log in any student must be have an account.

##### View Courses

The administrator is the only actor that is allowed to see all the existing courses.

##### Add Course

Allows the administrator to add a new subject to the collection.

##### Pick Course

The administrator is the user that is able to make changes in the structure of a course. He can do that for a course at a time so this use case which represents the choosing action.

1. Reset Course

This functionality allows the actor to delete all information related to the chosen course. It will unsubscribe students from the current lecture, will remove quiz results, quizzes, questions, recorded lessons, messages from chatroom.

##### Set teacher

It represents the action of assigning the teacher that will lecture to the current course.

##### View All Teachers

In order to choose the professor that will edify the students the administrator will see the complete list of teachers.

##### Subscribe Student

A student will receive notifications and will be able to attend a lecture only if the administrator has added him to the list of learners.

##### View All Students

An administrator is able to see the complete list of students for choosing which one will attend the current course.

1. View Attending Students

Because he is able to unsubscribe a student at a time the administrator can see the list of users that are following the selected course.

##### Remove student

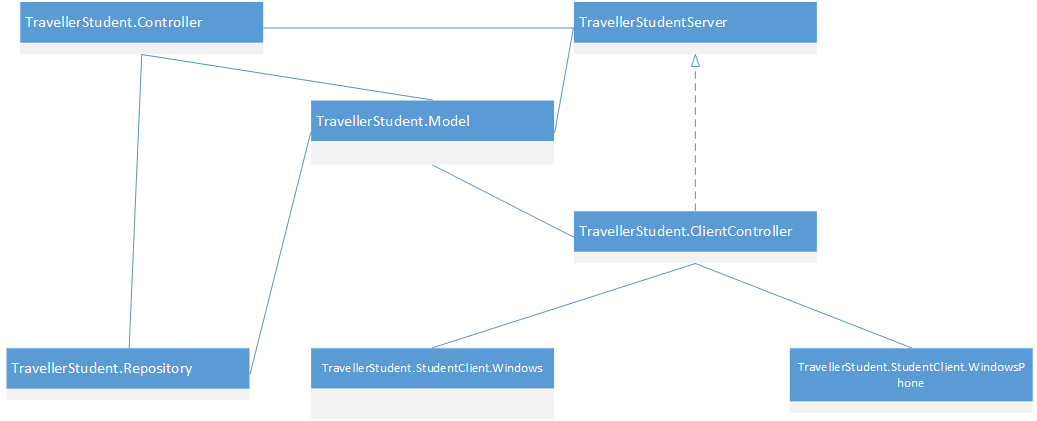
This feature allows the actor to remove a single user from list of attendees.

### 7.2 Architecture

The platform offers a client/server architecture. The client/server architectural style describes distributed systems that involve a separate client and server system, and a connecting network. It can be defined as the relationship between a client and one or more servers, where the client initiates one or more requests (perhaps using a graphical UI), waits for replies, and processes the replies on receipt. The server typically authorizes the user and then carries out the processing required to generate the result. The server may send responses using a range of protocols and data formats to communicate information to the client.

Another design pattern that has been implemented by this platform is model-view-controller. “The Model-View-Controller (MVC) pattern separates the modeling of the domain, the presentation, and the actions based on user input into three separate classes” [2, https://msdn.microsoft.com]:

* Model: “The model manages the behavior and data of the application domain, responds to requests for information about its state (usually from the view), and responds to instructions to change state (usually from the controller)” [1, Burbeck].
* View: The view manages the display of information.
* Controller: “The controller interprets the mouse and keyboard inputs from the user, informing the model and/or the view to change as appropriate” [2, <https://msdn.microsoft.com>].

The solution is organized in multiple projects, each of them representing an important component of the system. The diagram below shows the relationships between them.

The figure above also draws the schema of a layered architecture where each component serves a single role. “Layered architecture focuses on the grouping of related functionality within an application into distinct layers that are stacked vertically on top of each other. Functionality within each layer is related by a common role or responsibility. Communication between layers is explicit and loosely coupled. Layering your application appropriately helps to support a strong separation of concerns that, in turn, supports flexibility and maintainability.” [2, <https://msdn.microsoft.com>]

Further will be discussed each layer and will be explained its role in the system.

#### 7.2.1 Model Layer

It is named “TravellerStudent.Model” and its concern is to represent data. It is referenced by all the other layers because they work with the notions defined in this one. It also contains the base objects on top of which the other layers are built.

#### 7.2.2 Repository Layer

This level is concerned with interactions with the database. The only thing that it does is to read, insert and update the objects that are mapped in the Model and persist them.

#### 7.2.3 Business Logic Layer

Its role is to process data received from the user and from the repository layer.

#### 7.2.4 Services Layer

This is the entry point for the server. Any client will be able to make requests in order to retrieve information, or to send data to be persisted.

#### 7.2.5 Client Controller

This is a wrapper over the Services Layer, which will have the role of communicating with the server and process data received from it.

#### 7.2.6 User Interface

This time there are several projects that have that role, considering that there are two applications: one for Windows Phone 8.1 and one for Windows 8.1.

## Chapter 8.Conclusions.

## Chapter 9.Bibliografy

1. Burbeck, Steve. "Application Programming in Smalltalk-80: How to use Model-View-Controller (MVC)."University of Illinois in Urbana-Champaign (UIUC) Smalltalk Archive
2. https://msdn.microsoft.com