UNIVERSITY OF ECONOMICS AND LAW

**FACULTY OF INFORMATION SYSTEMS**

Ảnh có chứa biểu tượng, Nhãn hiệu, Phông chữ, Đồ họa

Mô tả được tạo tự động

**FINAL PROJECT REPORT**

**TEXT MINING**

**Topic:**

**FOSTERING ACADEMIC EXCELLENCE THROUGH UNIVERSITY STUDENT FEEDBACK: ENHANCING THE QUALITY OF SUBJECTS OF K20 IN FACULTY OF INFORMATION SYSTEMS**

Scientific fields: Data Analysis and Statistics

Specialization: Education Evaluation

**Ho Chi Minh City,** **August 9, 2023**

UNIVERSITY OF ECONOMICS AND LAW

**FACULTY OF INFORMATION SYSTEMS**

Ảnh có chứa biểu tượng, Nhãn hiệu, Phông chữ, Đồ họa

Mô tả được tạo tự động

**FINAL PROJECT REPORT**

**TEXT MINING**

**Topic:**

**FOSTERING ACADEMIC EXCELLENCE THROUGH UNIVERSITY STUDENT FEEDBACK: ENHANCING THE QUALITY OF SUBJECTS OF K20 IN FACULTY OF INFORMATION SYSTEMS**

**Team 2**

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| **NO.** | **Name** | **ID** | **Faculty** | **Role** | **Phone** | **Email** |
| 1 | Dao Phuong Anh | K204160660 | IS | Leader | 0867947072 | anhdp20416c @st.uel.edu.vn |
| 2 | Le Minh Tri | K204162004 | IS | Member | 0773213406 | trilm20416c @st.uel.edu.vn |
| 3 | Bui Thi Tri Tam | K204161999 | IS | Member | 0772198980 | tambtt20416c @st.uel.edu.vn |
| 4 | Vo Thi Phuong Anh | K204071522 | IS | Member | 0908533591 | anhvtp20407ca @st.uel.edu.vn |

Lecturer: **PhD. Tran Duy Thanh**

**Ho Chi Minh City, August 9, 2023**

# ACKNOWLEDGMENTS

In the research period, we got a lot of devoted help and advice from teachers. Besides, we also referenced from a number of authors. Thanks to your distribution, we can complete our final project to the fullest.

Despite our best effort, mistakes are inevitable. Therefore, we are glad to receive your judges and comments in order to improve our research. Those will be our enormous motivation to develop our project on the horizon.

# COMMITMENT

We commit that our final Text Mining project is unique due to the whole team’s research. There are still some documents we referenced from, we have listed and cited particularly in the report.

If all of the above are wrong, we will take all responsibilities from professors.

|  |  |
| --- | --- |
|  | Wednesday, 9th August 2023 |
|  | Commited group |
|  | Team 2 |

**CONTENTS**

[ACKNOWLEDGMENTS 3](#_Toc142485241)

[COMMITMENT 4](#_Toc142485242)

[**CONTENTS** 4](#_Toc142485243)

[CHAPTER 1. OVERVIEW 6](#_Toc142485244)

[1.1. Introduction 6](#_Toc142485245)

[1.2. Novelty 6](#_Toc142485246)

[1.3. Research Methodology 6](#_Toc142485247)

[1.4. Subject and Scope of the research 6](#_Toc142485248)

[1.4.1. Subjects of Research 6](#_Toc142485249)

[1.4.2. Scope of Research 6](#_Toc142485250)

[1.5. Research Questions 6](#_Toc142485251)

[CHAPTER 2. THEORETICAL BACKGROUND AND RELATED WORKS 6](#_Toc142485252)

[2.1. Related Works 6](#_Toc142485253)

[2.1.1. Sentiment Analysis and Text Mining 6](#_Toc142485254)

[2.1.2. Machine Learning in Education Feedback Analysis 6](#_Toc142485255)

[2.2. Sentiment Analysis 6](#_Toc142485256)

[2.3. Text Mining 6](#_Toc142485257)

[2.4. Related Tools 6](#_Toc142485258)

[2.4.1. Visual Studio 2022 6](#_Toc142485259)

[2.4.2. Microsoft SQL Server Management Studio 6](#_Toc142485260)

[2.4.3. SQL 6](#_Toc142485261)

[2.4.4. C# and Window Forms 6](#_Toc142485262)

[2.4.5. LINQ to SQL 6](#_Toc142485263)

[2.4.6. ML.NET 6](#_Toc142485264)

[CHAPTER 3. WINDOWS FORMS INTERFACE DESIGN 6](#_Toc142485265)

[3.1. frmHome 6](#_Toc142485266)

[3.2. frmSendFeedback 6](#_Toc142485267)

[3.3. frmLectures 6](#_Toc142485268)

[3.4. frmShowFeedback 6](#_Toc142485269)

[CHAPTER 4: DATABASE DESIGN AND APPLYING MACHINE LEARNING 6](#_Toc142485270)

[4.1 Database Design 6](#_Toc142485271)

[4.1.1. Database Table And Relationships 6](#_Toc142485272)

[4.1.2. Database Relationships 10](#_Toc142485273)

[4.2 Applying machine learning to the interface 11](#_Toc142485274)

[4.2.1 Preparing 11](#_Toc142485275)

[4.2.2 Student interacting 12](#_Toc142485276)

[4.2.3 Lecturers interating 13](#_Toc142485277)

[4.2.4 Train model processing 15](#_Toc142485278)

[CHAPTER 5. CONCLUSION & FUTURE WORKS 20](#_Toc142485279)

[5.1 Conclusion 20](#_Toc142485280)

[5.2 Future works 20](#_Toc142485281)

[REFERENCE 20](#_Toc142485282)

[APPENDIX 20](#_Toc142485283)

# CHAPTER 1. OVERVIEW

## Introduction

In the educational environment, students' perspectives on courses play a pivotal role in evaluating educational quality and enhancing the learning experience. Capturing students' feedback can yield invaluable insights into the strengths and limitations of a course, thereby facilitating the personalization of the educational experience and augmenting the course quality. Within the framework of the project " Fostering Academic Excellence through University student Feedback: Enhancing the Quality of Subjects of K20 in Faculty of Information Systems", the team gathers feedback after each course and analyzes the positive and negative factors encountered by students. Concurrently, the team will propose strategies to ameliorate the course quality from the institution's perspective.

## Novelty

In this project, our team will employ sentiment analysis techniques and course assessment based on student feedback. We will utilize the bi-dimensional opinion categorization method, known as Aspect-Based Sentiment Analysis (ABSA) using ML.NET package. This approach allows us to pinpoint both the positive and negative factors students allude to in their evaluations.

## Research Methodology

The primary objective of this project is to enhance students' overall satisfaction and educational experience. Our focus lies in augmenting goals in areas identified through sentiment analysis, formulating personalized educational services that cater to each student's needs, and enhancing learning efficacy as well as success throughout their educational journey.

## 1.4. Subject and Scope of the research

### 1.4.1. Subjects of Research

Our primary research subjects are students from the K20 in Faculty of Information Systems, University of Economics and Law. Their perspectives and feedback on courses serve as a vital data source for educational quality improvement.

### 1.4.2. Scope of Research

The research scope will predominantly center on collecting student feedback post each course. We do not constrain the collection period, ensuring a comprehensive perspective on course quality. The team will evaluate student feedback and consider how course quality improvements manifest from the institution's viewpoint.

## 1.5. Research Questions

The guiding question for our research is: "Solutions to enhance course quality based on student evaluation analysis: Proposing measures to ameliorate courses anchored in student feedback."

# CHAPTER 2. THEORETICAL BACKGROUND AND RELATED WORKS

## 2.1. Related Works

### 2.1.1. Sentiment Analysis and Text Mining

Thumbs up? Sentiment classification using machine learning techniques . In their seminal 2002 paper titled "Thumbs up? Sentiment classification using machine learning techniques," Pang, Lee, and Vaithyanathan explored the potential of machine learning in classifying movie reviews from the Internet Movie Database (IMDb) as positive or negative. Using various feature extraction techniques, such as the presence or absence of words and term frequency, they applied multiple machine learning algorithms, including Naive Bayes, Maximum Entropy, and Support Vector Machines. Their findings demonstrated that machine learning could be employed effectively for sentiment classification. Notably, unigrams (single words) served as especially potent features, especially when combined with classifiers like Naive Bayes or Maximum Entropy. The addition of bigrams (two-word combinations) offered further improvement, albeit less dramatically than unigrams. This pioneering research laid a foundation for numerous subsequent studies in sentiment analysis.

A survey of opinion mining and sentiment analysis . This paper provides a comprehensive review of the methodologies and advancements in the realm of opinion mining and sentiment analysis. Venturing deep into this domain, the authors meticulously unpack a myriad of techniques and tools used for extracting, analyzing, and understanding human sentiments and opinions from raw text. By offering a structured breakdown of both foundational approaches and novel innovations, they shed light on the challenges faced in sentiment analysis and discuss potential avenues for further research. Drawing from an extensive range of sources, Liu and Zhang present a holistic overview, making their work an invaluable resource for anyone seeking a nuanced understanding of sentiment analysis as it stood in the early 2010s.

### 2.1.2. Machine Learning in Education Feedback Analysis

MOOC performance prediction via clickstream data and social learning networks. . Brinton and Chiang delve into the potential of predicting student performance in Massive Open Online Courses (MOOCs). By harnessing data on student interactions coupled with insights from social learning networks, they sought to forecast how students would fare in these courses. Their methodology was centered on the meticulous examination of clickstream data—a detailed record of users' online interactions—and gauged possible metrics from social interactions. To achieve this, they employed sophisticated machine learning models, potentially integrating sequence prediction and network analysis techniques. Literature in the field corroborated their approach, suggesting that melding behavioral and social datasets could bolster prediction precision. However, this endeavor wasn't without its challenges. The vastness of MOOC data posed scalability issues. Moreover, the study had to grapple with concerns related to student privacy and the intricacies tied to interpreting clickstream data.

Sentiment Analysis of Students’ Feedback in MOOCs: A Systematic Literature Review . In the 2021 research paper, Dalipi, Zdravkova, and Ahlgren underscore the significance of discerning student sentiments in Massive Open Online Courses (MOOCs) feedback as a potent avenue for enhancing course quality. Their research methodology hinged on a systematic review of extant studies, which have incorporated sentiment analysis techniques on the feedback given by MOOC participants. Throughout the literature, a myriad of sentiment analysis techniques emerged, ranging from rudimentary lexicon-centric methods to intricate deep learning models. These diverse techniques have proven invaluable in extracting insights from the feedback, revealing pivotal indicators of student satisfaction, potential areas necessitating improvement, and the overall efficacy of the course. However, interpreting student feedback isn't devoid of challenges. The researchers pinpointed specific limitations, such as the inherent variability and subjectivity of student feedback. They also grappled with the intricacies of parsing sentiments across multiple languages and contending with feedback that might be laden with noise, sarcasm, or other ambiguities. Such insights demonstrate the complexities and potential of harnessing sentiment analysis for MOOC enhancement.

## 2.2. Sentiment Analysis

Sentiment analysis is a research field that aims to extract and analyze the subjective opinions, emotions, and attitudes of people from text. Sentiment analysis is a branch of natural language processing that uses computational methods to process, analyze, and synthesize natural language data. Sentiment analysis can be performed at different levels of granularity, such as document, sentence, aspect, or entity level. Sentiment analysis can also use different emotion models, such as categorical, dimensional, or appraisal models. Sentiment analysis has many applications in various domains, such as social media, e-commerce, customer service, and marketing.

The methodology of sentiment analysis involves several steps and techniques:

The first step is to collect and preprocess the text data that contains the sentiments and emotions of people. The text data can be obtained from various sources, such as online reviews, social media posts, blogs, news articles, or surveys. The preprocessing step can include tasks such as tokenization, normalization, stemming, lemmatization, stop-word removal, spelling correction, or noise removal.

The second step is to identify and extract the targets and sources of sentiment and emotion in the text. The targets are the entities or aspects that are being evaluated or commented on by the people. The sources are the people who express their opinions or emotions towards the targets. The identification and extraction of targets and sources can be done using techniques such as named entity recognition, part-of-speech tagging, dependency parsing, or rule-based methods.

The third step is to classify and quantify the sentiment and emotion in the text. The classification can be done using different approaches, such as lexicon-based methods, machine learning methods, or hybrid methods. Lexicon-based methods use a predefined list of words or phrases that have associated sentiment or emotion scores or polarities. Machine learning methods use supervised or unsupervised algorithms to learn from labeled or unlabeled data and generate sentiment or emotion labels or scores for the text. Hybrid methods combine both lexicon-based and machine learning methods to improve the accuracy and robustness of sentiment or emotion analysis. The quantification of sentiment and emotion can be done using different scales or models, such as binary (positive or negative), ternary (positive, negative, or neutral), ordinal (very positive, positive, neutral, negative, very negative), continuous (a numerical score between -1 and 1), categorical (a set of predefined emotion categories), dimensional (a set of dimensions such as valence, arousal, and dominance), or appraisal (a set of appraisal variables such as pleasantness, attention, certainty, etc.).

## 2.3. Text Mining

Text mining is the process of extracting meaningful information from large amounts of unstructured text data. It involves the use of advanced analytical techniques, such as Naïve Bayes, Support Vector Machines (SVM), and other deep learning algorithms, to identify patterns and relationships within the data. Text mining can be used for a variety of purposes, ranging from basic descriptions of text content through word counts to more sophisticated uses such as finding links between authors and evaluating the content of scripts (e.g., automated marking of essays).

Text mining is a relatively new field that has emerged as a result of the increasing availability of digital text data. With the advent of new technologies, text data are being used in new forms of communication, such as text messaging, social media activity, blogs and web searches. The increasing availability of published text, sophisticated technologies and growing interest in organizations in extracting information from text has led to replacing (or at least supplementing) the human effort with automatic systems.

Text mining owes its origin to a combination of various related fields – Data Mining (DM), Artificial Intelligence, Statistics, Database Management, Library Science and Linguistics. Its basic purpose is to process the unstructured information contained in text data in order to make text accessible to various DM statistical algorithms. This could help make text data as informative as standard structured data and allow us to investigate relationships and patterns which would otherwise be extremely difficult, if not impossible, to discover.

The process of text mining comprises several activities that enable you to deduce information from unstructured text data. Before you can apply different text mining techniques, you must start with text preprocessing, which is the practice of cleaning and transforming text data into a usable format. Text mining tools and natural language processing (NLP) techniques allow us to transform unstructured documents into a structured format to enable analysis and the generation of high-quality insights. This, in turn, improves the decision-making of organizations, leading to better business outcomes.

In conclusion, text mining is a powerful tool that can help organizations extract valuable insights from large amounts of unstructured text data. By applying advanced analytical techniques and leveraging the power of natural language processing, companies can gain a deeper understanding of their customers, markets and operations. As the amount of digital text data continues to grow, the importance of text mining is only set to increase.

## 2.4. Related Tools

### 2.4.1. Visual Studio 2022

According to Microsoft, Visual Studio is the best IDE to build rich, beautiful, cross platform applications for Windows, Mac, Linux, iOS, and Android. Build rich clients apps using a range of technologies such as; WinForms, WPF, WinUI, MAUI, or Xamarin.

In this project, we are using visual studio 2022 as an Integrated Development Environment (IDE). An IDE is a software application that provides a comprehensive environment for software development. It typically includes various tools, features, and capabilities to facilitate coding, debugging, testing, and deploying software applications.

In your context, we are using Visual Studio 2022 to develop and implement our machine learning project with ML.NET. The IDE will provide us with the necessary tools to write code, manage your project files, load and train ML models, and integrate those models into our application for prediction.

### 2.4.2. Microsoft SQL Server Management Studio

SQL Server Management Studio (SSMS) is an integrated environment for managing any SQL infrastructure. Use SSMS to access, configure, manage, administer, and develop all components of SQL Server, Azure SQL Database, Azure SQL Managed Instance, SQL Server on Azure VM, and Azure Synapse Analytics. SSMS provides a single comprehensive utility that combines a broad group of graphical tools with many rich script editors to provide access to SQL Server for developers and database administrators of all skill levels.

It's important to note that while SSMS can be helpful for certain aspects of data management, it might not directly interact with ML.NET code. Instead, it serves as a complementary tool for handling data that is used in machine learning project.

### 2.4.3. SQL

Structured Query Language, commonly known as SQL, is an essential component of our project's data management system. Serving as the standard language for relational database management, SQL empowers us to interact with stored data using various operations. These range from simple data retrieval tasks, like fetching user profiles, to more complex analytical functions, analyzing trends within our datasets. The utilization of SQL ensures that we can access, modify, and manage our data in an efficient, secure, and consistent manner. As our project scales, SQL's robustness and flexibility will enable seamless data transactions and optimizations. By integrating SQL into our system, we aim to ensure data integrity, facilitate real-time data access, and lay a strong foundation for future data-driven innovations.

### 2.4.4. C# and Window Forms

C# and Windows Forms are crucial components of our project, enabling us to develop robust and user-friendly applications. C# is a powerful and versatile programming language, while Windows Forms provides a graphical user interface (GUI) framework for creating desktop applications. Together, they form a potent combination that empowers us to build feature-rich software solutions with ease.

C# is a modern, object-oriented programming language developed by Microsoft. Its rich set of features, including strong type checking, garbage collection, and dynamic binding, make it ideal for building a wide range of applications. C# is highly versatile, allowing us to create everything from simple console applications to complex, enterprise-level systems. Its integration with the .NET framework provides access to a vast library of pre-built functions, accelerating development and fostering code reusability.

Windows Forms (WinForms): Windows Forms is a GUI toolkit within the .NET framework that simplifies the creation of graphical interfaces for Windows applications. By leveraging WinForms, we can design user-friendly interfaces with buttons, textboxes, menus, and other interactive elements. The drag-and-drop functionality and event-driven programming model of WinForms expedite the creation of visually appealing interfaces and streamline user interactions.

### 2.4.5. LINQ to SQL

LINQ to SQL is a vital component of our project, serving as a bridge between our C# code and the underlying relational database. LINQ, or Language-Integrated Query, is a powerful feature of C# that enables developers to query and manipulate data using a natural and intuitive syntax. LINQ to SQL specifically allows us to interact with the database using LINQ queries, making data access and manipulation seamless and efficient.

By using LINQ to SQL, we can benefit from several advantages. First, we can write more readable, maintainable, and testable code, as we do not need to write complex SQL queries or use data adapters or command objects. Second, we can leverage various features that LINQ to SQL provides, such as lazy loading, change tracking, concurrency control, and stored procedures. Third, we can easily integrate our data model with other .NET technologies, such as WPF, ASP.NET, or WCF.

However, we also face some challenges when using LINQ to SQL. One of them is that LINQ to SQL does not support all the SQL features or database types that we may need in our project. For example, LINQ to SQL does not support full-text search, XML data type, or Oracle database. Another challenge is that LINQ to SQL may have performance issues when dealing with large or complex data sets. For example, LINQ to SQL may generate inefficient SQL queries or consume more memory than necessary.

Therefore, we need to carefully evaluate the trade-offs between the benefits and the challenges of using LINQ to SQL in our project. We also need to use some best practices and tools to optimize our code and improve our performance.

### 2.4.6. ML.NET

ML.NET is a free, open-source, and cross-platform machine learning framework for the .NET developer platform.

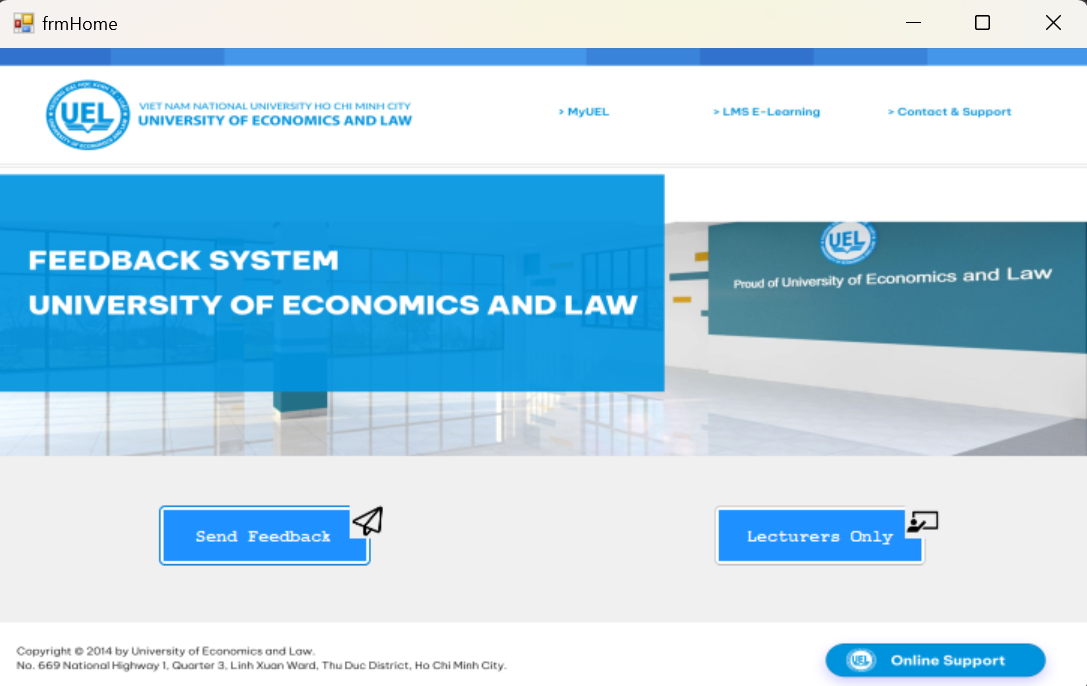
ML.NET allows you to train, build, and ship custom machine learning models using C# or F# for a variety of ML scenarios. ML.NET includes features like automated machine learning (AutoML) and tools like ML.NET CLI and ML.NET Model Builder, which make integrating machine learning into your applications even easier.

ML.NET is well-suited for a variety of applications, including business intelligence, predictive analytics, recommendation systems, fraud detection, sentiment analysis, and more. Its integration with the .NET ecosystem makes it a valuable tool for .NET developers looking to incorporate machine learning capabilities into their projects.

# CHAPTER 3. WINDOWS FORMS INTERFACE DESIGN

In the context of the rapidly evolving educational environment, collecting and managing feedback from students has become more crucial than ever. Recognizing the significance of this, our team has chosen the topic of Feedback Management to comprehensively understand how our school handles information. Below is the "frmHome" Form, an integral part of the school's feedback management system. This Form has been designed to closely align with the school's website, providing a user-friendly interface and functions ranging from submitting feedback to tracking evaluations for each course. Through our research, we have observed that this communication approach not only efficiently gathers feedback but also ensures that both students and instructors can interact smoothly and conveniently.

## 3.1. frmHome



This form includes functionalities to open windows related to student feedback management, including windows for students to submit feedback and for Lecturers to view and provide general evaluations for each course.

Button "Lecturers Only": When users click on this button, a new window will appear to display information about the feedback stored in the feedback database

Button "Send Feedback": When users click on this button, a new window will appear for students to submit feedback. This could be the place where students can input and send feedback about lectures, course quality, and satisfaction levels.

## 3.2. frmSendFeedback

Users will select the "Send Feedback" button in the "frmHome" form.

Ảnh có chứa văn bản, ảnh chụp màn hình, phần mềm, Trang web

Mô tả được tạo tự động

This form allows users to input information, select their class and corresponding subject, and then submit their feedback to the database.

Textbox "Student Code" and "Email":

- Students will input their information into these textboxes.

- It's important for students to provide accurate data, as after clicking the "Send" button, the form will verify whether a student with the provided student code and email exists in the database.

Combobox "Class": Activated when students click on the ComboBox to select their class. In this section:

- Check if the selected class at the faculty exists.

- Query data from the "Classes" table in the database to retrieve a list of available classes.

- Update the ComboBox "Class" with the list of available classes.

- When the user changes their selection in ComboBox "Class", clear the current data in ComboBox "Subject" to prepare for selecting a new subject.

Combobox Subject: Activated when students click on the ComboBox to select their subject. In this section:

- Check if the selected class from ComboBox "Class" exists.

- If it exists, query to retrieve a list of subjects related to the selected class.

- Update ComboBox "Subject" with the list of subjects corresponding to the selected class.

Button Send: Activated when students press the "Send" button:

- Retrieve student's information (student code, email) to verify their existence in the database.

- Check if Faculty, Class, and Subject match the student.

- Add the entered information into the database, particularly into the "Feedback" table.

- Clear the input content in ComboBoxes and textboxes, and display a success message.

This form facilitates the process of students providing feedback by ensuring the accuracy of their input and offering a structured way to select classes and subjects. It then stores their feedback into the database for further analysis and evaluation.

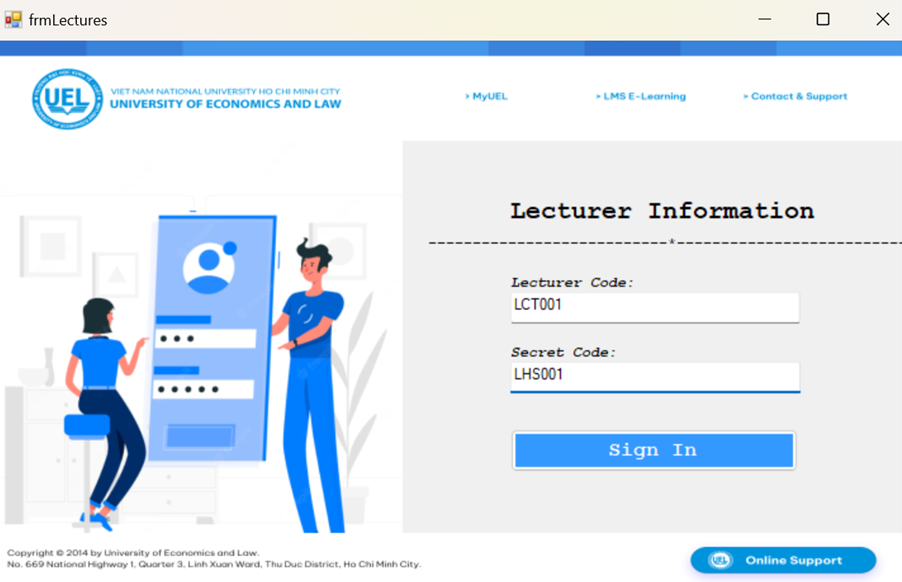
Ảnh có chứa văn bản, ảnh chụp màn hình, phần mềm, Trang web

Mô tả được tạo tự động

Button Home: Activated when users click on the "Home" button. This button's function is to close the current form and return to the "frmHome" form.

## 3.3. frmLectures

This indicates that the user intends to access the "frmLectures" form. This form likely provides access to lecturer-specific functions and feedback viewing capabilities, which are limited to lecturers only.

  
This form enables lecturers to log in and view feedback from students. If there are more than 3 consecutive login failures, the form will be closed, and the user will be redirected to "frmHome."

The form has been initialized with a "FeedbackDataContext" object to establish a connection with the database. A variable called "loginAttempts" has been initialized to keep track of the number of login failures.

Button Sign In: When a lecturer clicks the "Sign In" button to log in using their Lecturer Code and Secret Code:

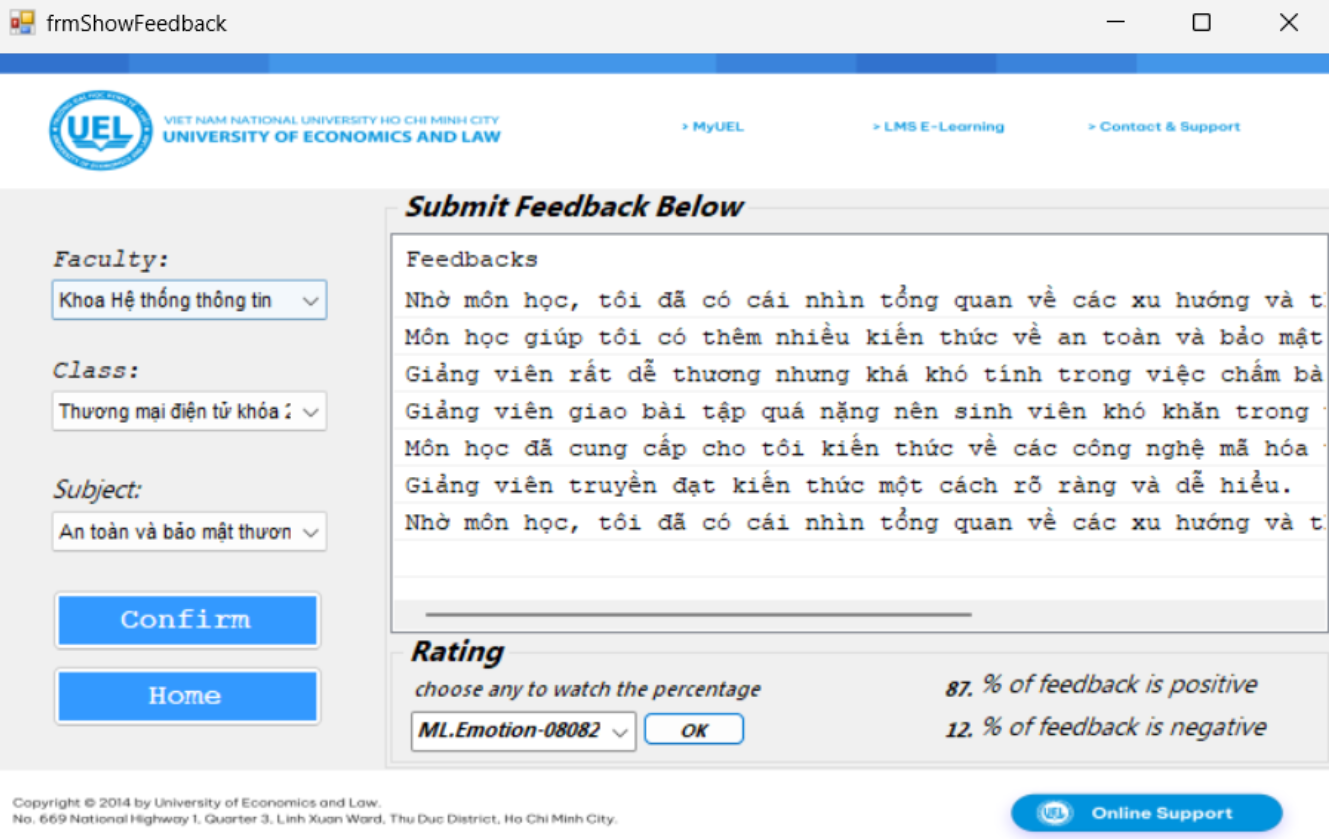
- Retrieve the Lecturer Code and Secret Code from the respective textboxes on the form.

- Check if the provided login information matches any entry in the "Lecturers" table.

- If the information is correct, open the "frmShowFeedback" form (described in the following section) to display student feedback.

- If the information is incorrect, increment the "loginAttempts" variable and check if the number of login attempts exceeds the limit (3 attempts). Display a message about exceeding the login attempt limit, close the "frmLectures" form, and open the "frmHome" form, allowing the user to return to the main screen.

## 3.4. frmShowFeedback



This form allows users to select Faculty, Class, and Subject, and then displays corresponding feedback from the database. Users can also return to "frmHome" using the "Home" button.

Combobox Class: Activated when users click on the ComboBox to select their class. The functionality here is similar to that in "frmSendFeedback."

- Check if the selected class at the faculty exists.

- Query data from the "Classes" table in the database to retrieve a list of available classes.

- Update the ComboBox "Class" with the list of available classes.

- When the user changes their selection in ComboBox "Class," clear the current data in ComboBox "Subject" to prepare for selecting a new subject.

Combobox Subject: Activated when users click on the ComboBox to select the subject. The functionality here is similar to that in "frmSendFeedback."

- Check if the selected class from ComboBox "Class" exists.

- If it exists, query to retrieve a list of subjects related to the selected class.

- Update ComboBox "Subject" with the list of subjects corresponding to the selected class.

Button Confirm: Activated when users click the "Send" button:

- Check if Faculty, Class, and Subject have been selected from the ComboBoxes.

- If selected, perform a query to retrieve corresponding feedback data from the database.

- Add the entered information into the database and display it in the ListView "Feedback" for the Lecturer to view.

- If the user switches to a different Class or Subject, clear the existing items in the ListView "Feedback" and add new items from the query data into the ListView "Feedback."

The series of forms in the feedback management system not only facilitate efficient feedback collection but also create a conducive interactive environment between students and lecturers.

# CHAPTER 4: DATABASE DESIGN AND APPLYING MACHINE LEARNING

## 4.1 Database Design

### 4.1.1. Database Table And Relationships

For this topic, we designed a database of 10 tables covering objects of faculty and student who are directly involved in the process of teaching, studying and giving feedback.

These 10 tables include tables: Student, Lecturer, Faculty, Subject, Class, Major, Class Register, Semester, Feedback, Course. The details of these tables are explained in the following subsections.

Disclaimer: This database is based on the actual context of Faculty of Information Systems, University of Economics and Law but does not consist of all the details, since this project has its scope and time limitation.

**4.1.1.1. Table Faculty**

|  |  |  |  |
| --- | --- | --- | --- |
| **Column** | **Data Type** | **Primary Key** | **Nullable** |
| FacultyCode | nchar(12) | x | No |
| FacultyName | nvarchar(100) |  | Yes |

This table describes the information of faculties, for example: Faculty of Information Systems will be stored as [FacultyCode: FIS – FacultyName: Khoa Hệ thống thông tin]. Besides there are some other faculties who are responsible for some subjects.

**4.1.1.2. Table Semester**

|  |  |  |  |
| --- | --- | --- | --- |
| **Column** | **Data Type** | **Primary Key** | **Nullable** |
| SemesterCode | nchar(12) | x | No |
| StartDate | date |  | Yes |
| EndDate | date |  |  |

This table describes the information of semesters of different schoolyears. As we will have a Course table, different classes may be the same subject but just differentiated by the semester it is hosted.

For example: Semester I of School Year 2022-2023 is stored as 221, with its start and end date.

**4.1.1.3. Table Subject**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Column** | **Data Type** | **Primary Key** | **Nullable** | **Foreign Key** |
| SubjectCode | nchar(12) | x | No | - |
| SubjectName | nvarchar(100) | - | Yes | - |
| FacultyCode | nchar(12) | - | Yes | x  (Faculty.FacultyCode) |
| Type | nvarchar(20) | - | Yes | - |
| Credits | int | - | Yes | - |

This table describes the information of teaching subjects. Each subject will have its code, name, the faculty of hosting, type (compulsory or optional) and the number of credits. The column FacultyCode is a foreign key (FK) refences to table Faculty.

For example: a row of this table is stored as: MI55 - Chuyển đổi số trong kinh doanh – FIS – BB - 3

**4.1.1.4. Table Major**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Column** | **Data Type** | **Primary Key** | **Nullable** | **Foreign Key** |
| MajorCode | nchar(12) | x | No | - |
| MajorName | nvarchar(100) | - | Yes | - |
| FacultyCode | nchar(12) | - | Yes | x  (Faculty.FacultyCode) |

This table describes the information of majors. A major information consists of its identifical code, name and the faculty code of the faculty it belongs to. Column FacultyCode is also a FK referencing table Faculty.

For example: a row of this table is stored as: 406 – Hệ thống thông tin quản lý - FIS

In this project, as the scope only covers Faculty of Information System (FIS) specialised subject and teaching quality, only 3 majors of FIS are stored.

**4.1.1.5. Table Class**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Column** | **Data Type** | **Primary Key** | **Nullable** | **Foreign Key** |
| ClassCode | nchar(12) | x | No | - |
| ClassName | nvarchar(100) | - | Yes | - |
| Program | nchar(10) | - | Yes | - |
| MajorCode | nchar(12) | - | Yes | x (Major.MajorCode) |

This table describes the information of student classes. The class information consists of class code, class name, program and the major the class belongs to.

For example: a row of this table is stored as: K20406 - Hệ thống thông tin quản lý khóa 20 - CQDT - 406

**4.1.1.6. Table Lecturer**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Column** | **Data Type** | **Primary Key** | **Nullable** | **Foreign Key** |
| LecturerCode | nchar(12) | x | No | - |
| LecturerName | nvarchar(100) | - | Yes | - |
| Email | nvarchar(100 | - | Yes | - |
| SecretCode | nchar(12) | - | Yes | - |
| FacultyCode | nchar(12) | - | Yes | x  (Faculty.FacultyCode) |
| Rank | varchar(10) | - | Yes | - |

This table describes the information of lecturers, who will later be responsible for teaching class and receiving feedback. The Lecturer table is storing the code, name, email, secretcode (used for logging into the feedback system, this column is used for system meaning only), their working faculty (the column act as a FK referencing the Faculty table) and their rank (Master, PhD...).

For example: a row of this table is stored as: LCT017 - Trần Duy Thanh - [thanhtd@uel.edu.vn](mailto:thanhtd@uel.edu.vn) - TDT001 - FIS - TS

**4.1.1.7. Table Course**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Column** | **Data Type** | **Primary Key** | **Nullable** | **Foreign Key** |
| CourseCode | nchar(12) | x | No | - |
| LecturerCode | nchar(12) | - | Yes | x  Lecturer.LecturerCode |
| SemesterCode | nchar(12) | - | Yes | x  (Semester.SemesterCode) |
| ClassCode | nchar(12) | - | Yes | x  Class.ClassCode |
| Capacity | int | - | Yes | - |

This table describes the information of teaching course. In this project, we define ‘Course’ is a class of Subject that been taught every semester. Therefore, the Course table consists of information about objects relevant to a course: its identifiable code, in-charge lecturer code, semeter code in which the course hosted, the belonging class code and its capacity.

For example: a row of this table is stored as: 222IS2901 - LCT008 - 222 - IS29 - K20411 - 80

**4.1.1.8. Table Student**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Column** | **Data Type** | **Primary Key** | **Nullable** | **Foreign Key** |
| StudentCode | nchar(12) | x | No | - |
| StundentName | nvarchar(100) | - | Yes | - |
| YearOfEntrance | int | - | Yes | - |
| Sex | bit | - | Yes | - |
| DOB | date | - | Yes | - |
| Email | nvarchar(70) | - | Yes | - |
| IsDelete | bit | - | Yes | - |
| ClassCode | nchar(12) | - | Yes | x  Class.ClassCode |

This table describes the information of students studying at the school/faculty. Each student will have their information such as Code; Name, YearOfEntrance, Sex, DOB, Email, Class Code of the class they belong to. Finally, the column IsDelete indicates if the student is still studying at the moment.

**4.1.1.9. Table Feedback**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Column** | **Data Type** | **Primary Key** | **Nullable** | **Foreign Key** |
| FeedbackCode | nchar(12) | x | No | - |
| StudentCode | nchar(12) | - | Yes | Student.StudentCode |
| CourseCode | nchar(12) | - | Yes | Course,CourseCode |
| Text | text | - | Yes | - |

This table describes the information of feedback submitted by students. The StudentCode indicates the student who wrote the feedback. CourseCode is the course they commented on and Text is the full feedback.

**4.1.1.10. Table Course Register**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Column** | **Data Type** | **Primary Key** | **Nullable** | **Foreign Key** |
| CourseCode | nchar(12) | x | No | - |
| StudentCode | nchar(12) | x | No | - |

This table is used to eliminate the relationship between Course and Student, where each student can register many courses and similarly, one course can have many students registered.

The relationships of this and others in the database is described in the section following up.

### 4.1.2. Database Relationships

The relationships are implied based on the foreign key constraints. In particular:

- Each Course is related to a Class, a Lecturer, a Semester, and a Subject. Meaning that each course only has one lecturer, belongs to one semester, one subject and one class only.

- Each Course Registration (CourseRegister) is related to a Course and a Student. We can see that each student can register many courses and similarly, one course can have many students registered, so the multiple-multiple relationship here is solved by creating CourseRegister table having 2 primary keys.

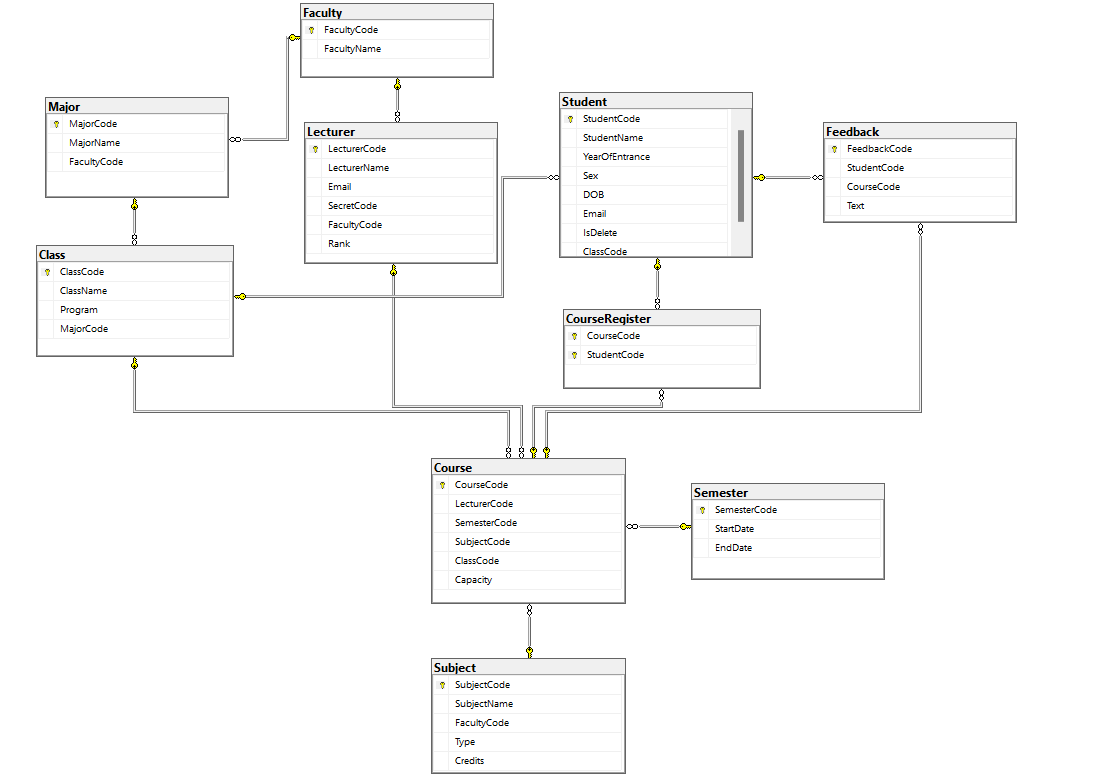
- Each Feedback is related to a Course and a Student, meaning that each feedback only about a course and submitted by a single student.

- Each Lecturer is belonged to a Faculty, whereas each faculty has more than one lecturer.

- Each Major is belonged to a Faculty, whereas each faculty has different majors.

- Each Student is belonged to a Class, while each class has many students. Double-majored students will still belong to their first class only.

- Each Subject is in charge by one faculty.



*Database Diagram*

These relationships help in maintaining data integrity and structuring the information in a way that reflects the academic domain. The database appears to be designed to handle aspects of academic administration, including course registration, class management, faculty-student interactions, and more.

## 4.2 Applying machine learning to the interface

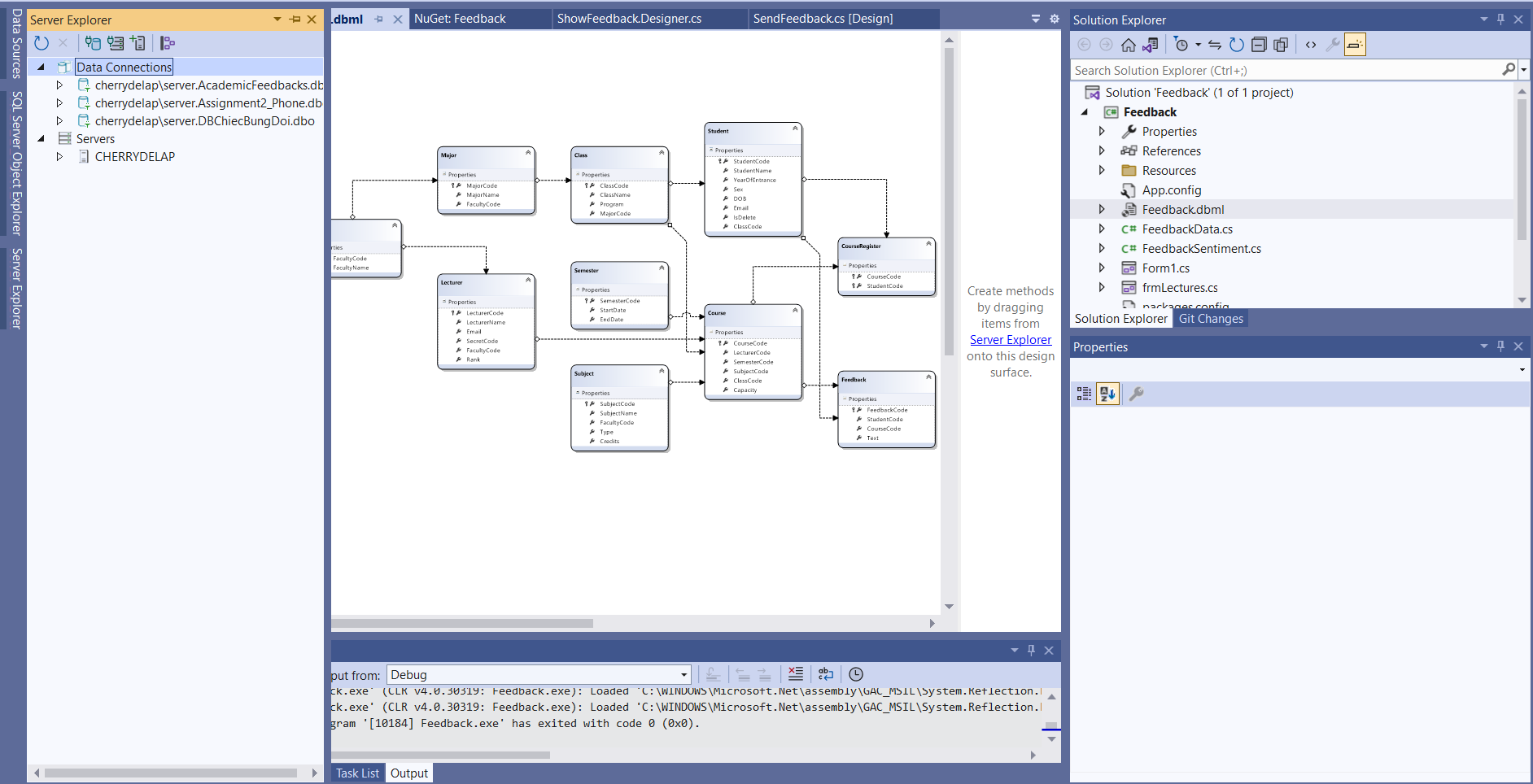
### 4.2.1 Preparing

Download Microsoft's ML.net package on Visual Studio 2022

A screenshot of a computer

Description automatically generated

Then connect check the connection of SQL server to the database, once the server is connected to the project, database will be shown as the photo below



Label all feedbacks text with 0 is negative point of view and 1 is positive point of view.

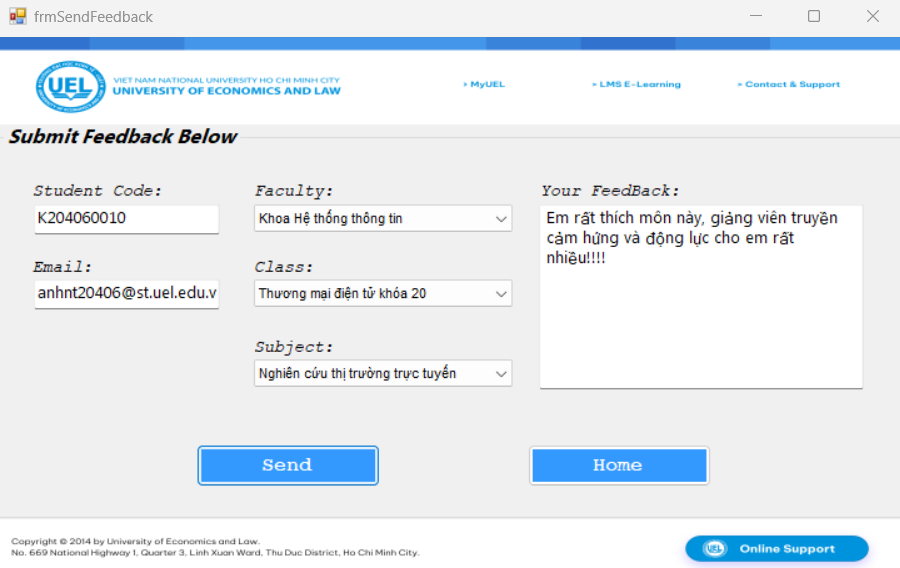
A screenshot of a computer

Description automatically generated

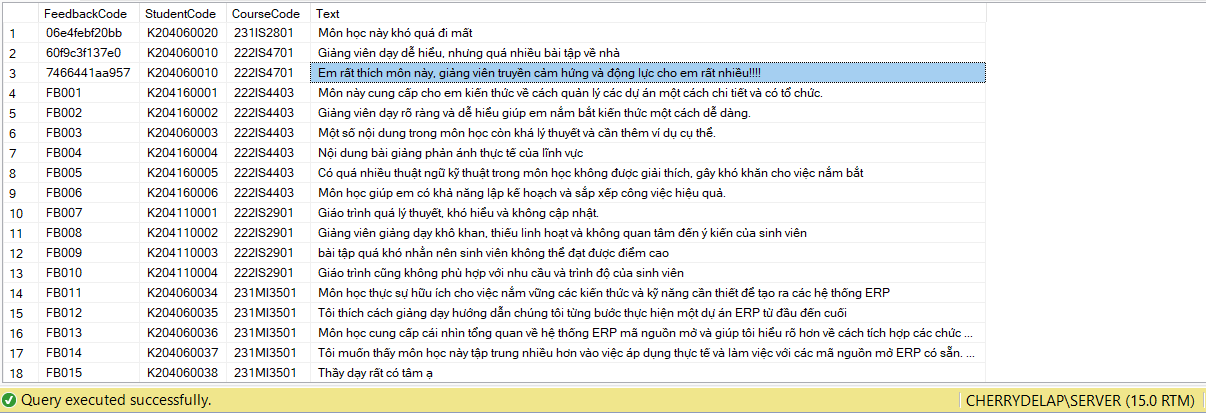
### 4.2.2 Student interacting

By clicking “Send Feedbacks” button, our application required students to fill the information to the blank space. Students information that they fill need to be matched with that of information in the database. Once they click “Send” button, the feedback text will be updated immediately in the database. Our group will export these feedback texts as text file in order to label the positive or negative point of view.

This is an example of filling information of student



The feedback text is updated to the database

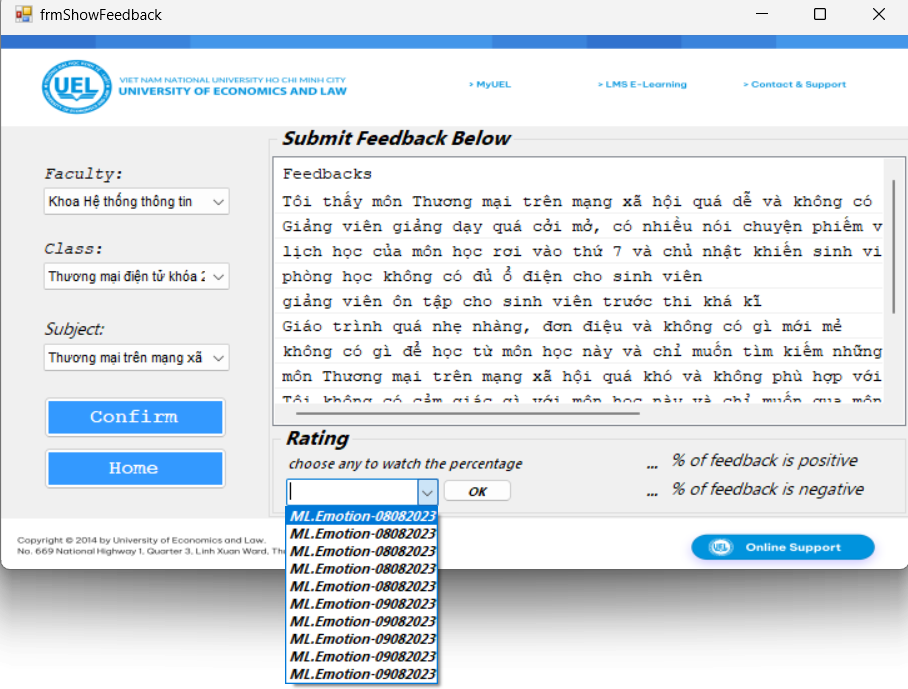


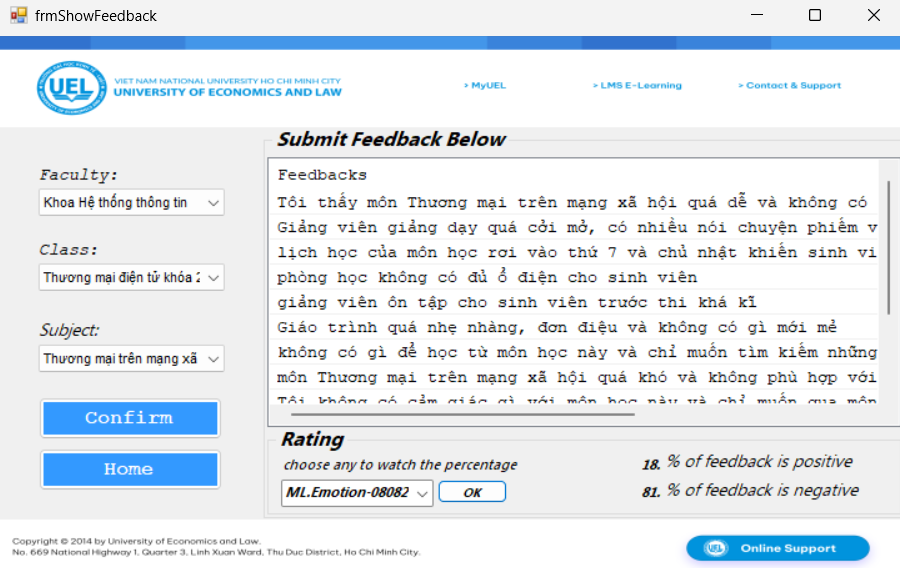
### 4.2.3 Lecturers interating

By clicking the “Lecturers only” button, the lecturer need to enter their lecturer code and secret code that matching with the information in the database to login.



After signing in the system, lecturers are available to view full feedback text from student, however, they are unable to know who create that feedbacks. The lecturers can also view the percentage of negative or positive from the feedbacks.





### 4.2.4 Train model processing

**Preparing**

A screenshot of a computer program

Description automatically generated

Using the necessary library, in this project context, we are using ML.NET. Inside the class, several class members are declared and initialized:

**FeedbackDataContext dbContext**: An instance of a data context class, possibly related to Entity Framework or LINQ to SQL, used for database operations.

**mlContext**: An instance of the MLContext class from the ML.NET library. It's used to create a context for machine learning operations.

**dataView**: An IDataView that is used to hold and manipulate the data for machine learning. It's later used for model training and evaluation.

**splitDataView:** It seems to hold the split data for training and testing sets after a train-test split operation.

**model:** This variable is expected to hold the trained ML.NET model.

**folder:** A string that indicates the name of the folder where the trained model will be saved or loaded from.

**BackgroundWorker worker:** An instance of BackgroundWorker class, which is likely used to perform background tasks without blocking the main UI thread.

**modelSchema:** It's an object that might be used to hold information about the schema of the trained model.

Using Constructor and Form Initialization

Overall, this code sets up the initial environment for a Windows Forms application that uses the ML.NET library for machine learning. It initializes the MLContext, prepares variables for data manipulation, sets up a BackgroundWorker for background tasks, and more. The specific functionality of this form would likely include loading data, training and evaluating a machine learning model, and potentially displaying results to the user.

A close-up of a computer screen

Description automatically generated

Firstly, a new instance of the BackgroundWorker class named worker is created.

Then, an event handler named Worker\_DoWork is attached to the DoWork event of the BackgroundWorker. This event handler contains the code that will be executed in the background.

Overall, this code sets up a background worker to perform an asynchronous task without freezing the UI, and it specifies an event handler that will execute the task.

**Load data**



This code snippet represents the Worker\_DoWork event handler, which is executed asynchronously by a BackgroundWorker when its RunWorkerAsync method is called. The purpose of this event handler is to load data from a text file, perform data preprocessing, and train a binary classification model using ML.NET.

The LoadFromTextFile method is used to load data from the specified file into an IDataView. The FeedbackData class is used to specify the structure of the data.

**Train-Test Split**

Split data to 8:2 with 80% for training and 20% for testing.

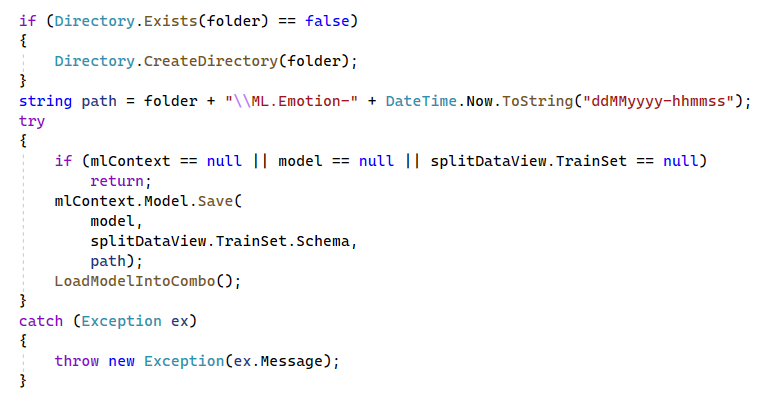
**Creating an Estimator and Training the Model**

An estimator pipeline is constructed using ML.NET's transformation and trainer components.

The FeaturizeText transformation is used to convert the text data into numerical features.

The SdcaLogisticRegression trainer is used for binary classification. It's a linear classifier optimized for sparse data.

The Fit method trains the model using the training set.



We are checking whether the directory existed or not, if not, create a new one. Then save the model as:

- This section verifies that essential components (mlContext, model, and splitDataView.TrainSet) are not null.

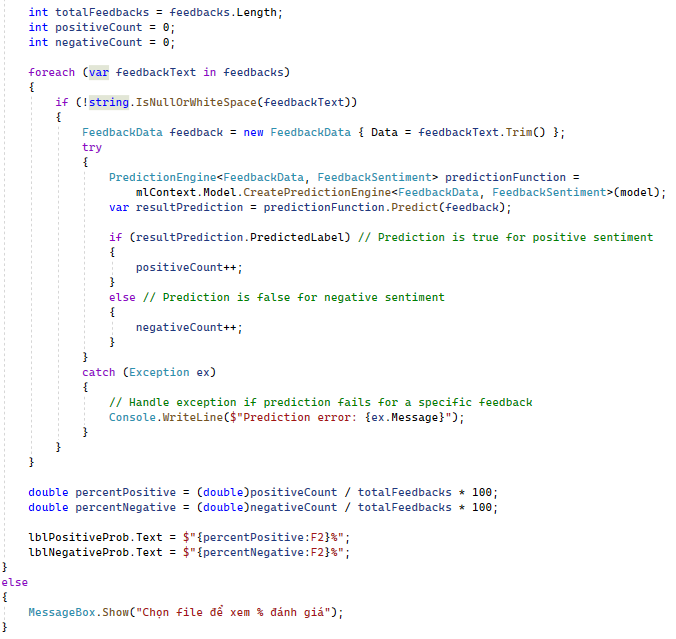
- The mlContext.Model.Save method is used to save the trained model to the specified path.

- The model parameter is the trained model to be saved.

- splitDataView.TrainSet.Schema provides the schema of the training data used to create the model.

- The LoadModelIntoCombo function is called after saving the model to update a combo box with the saved model's path.

**Using model to show the percentage of positive and negative sentiments**



This code snippet is designed to read feedbacks from a text file, classify their sentiment using a trained ML.NET model, and then calculate and display the percentage of positive and negative sentiments among the feedbacks.

First, variables are initialized to keep track of the total number of feedbacks, the count of positive sentiments, and the count of negative sentiments.

Then, a FeedbackData object is created with the feedback text to prepare it for prediction. The code attempts to create a prediction engine using the loaded ML.NET model and then predicts the sentiment of the feedback. Based on the prediction result, the counts of positive and negative sentiments are updated.

After iterating through all feedbacks, the code calculates the percentages of positive and negative sentiments. The calculated percentages are then displayed in UI labels (lblPositiveProb and lblNegativeProb).

# CHAPTER 5. CONCLUSION & FUTURE WORKS

## 5.1 Conclusion

In the realm of modern education, the voice of the student holds an undeniable significance. This study embarked on a journey to harness the power of student feedback and channel it towards the betterment of academic subjects. Through the lens of sentiment analysis and cutting-edge technology, we aimed to bridge the gap between students' experiences and subject enhancement.

The findings of this study underscore the valuable insights that student feedback can provide. It has become increasingly evident that the quality of education is not solely determined by curriculum design or pedagogical methods; rather, it is an intricate interplay between these elements and the lived experiences of students. By embracing their feedback, institutions open the door to a more responsive and relevant education system.

## 5.2 Future works

While this study has provided a significant stride in understanding student sentiment and its impact on subject enhancement, there lies an uncharted frontier: creating a predictive model that not only analyzes feedback sentiment but also suggests actionable solutions for educators.

Building upon the sentiment analysis foundation, the next frontier involves crafting a sophisticated model that can not only discern sentiments but also suggest tailored solutions based on the content of the feedback. This AI-powered model could identify recurring issues, patterns, and underlying causes, offering proactive strategies for instructors to enhance their teaching methods and subject materials.

Then, implementing advanced NLU techniques can enable the model to grasp the subtleties of language, context, and intent. By understanding the nuances within feedback, the model can provide more accurate and contextually relevant solutions.

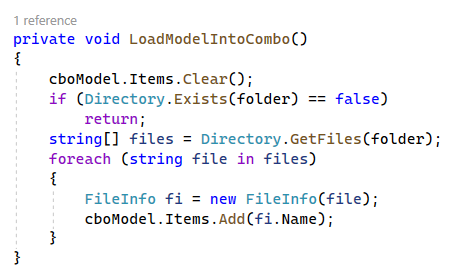
Leveraging collaborative filtering techniques used in recommendation systems, the model could identify solutions applied successfully in similar scenarios. This would enable instructors to learn from their peers' experiences and adapt best practices.

In summary, the envisioned solution suggestion model holds immense promise in revolutionizing how institutions leverage feedback for subject enhancement. By not just identifying issues but also providing proactive solutions, this model can empower educators to cultivate a learning environment that is responsive, innovative, and ultimately focused on nurturing the academic excellence of each student.

# REFERENCE

|  |  |
| --- | --- |
| [1] | Pang, B., Lee, L., & Vaithyanathan, S., "Thumbs up? Sentiment classification using machine learning techniques," 2002. |
| [2] | Liu, B., & Zhang, L., "A survey of opinion mining and sentiment analysis," *Mining text data,* pp. 415-463, 2021. |
| [3] | Brinton, C. G., & Chiang, M., "MOOC performance prediction via clickstream data and social learning networks," *In 2015 IEEE conference on computer communications,* pp. 2299-2307, 2015. |
| [4] | Dalipi, F., Zdravkova, K., & Ahlgren, F., "Sentiment analysis of students’ feedback in MOOCs: A systematic literature review," Frontiers in artificial intelligence, 2021. |
| [5] | Mata, P., Mata, M. N., & Martins, J. N., "Sentiment analysis: A literature review.".*1st International Conference on Management, Technology and Tourism: Social Value Creation–ICOMTT2020.* |
| [6] | "Sentiment Analysis Mining Opinions, Sentiments, and Emotions," 2020. [Online]. Available: https://www.cambridge.org/core/books/sentiment-analysis/introduction/563742A639EEE9F5AB3F29CB2387E41C. [Accessed 8 8 2023]. |
| [7] | Michael W. Berry, Jacob Kogan, Text Mining: Applications and Theory, Wiley, 2010. |
| [8] | "IBM," [Online]. Available: https://www.ibm.com/topics/text-mining. [Accessed 8 8 2023]. |
| [9] | "Microsoft," [Online]. Available: https://learn.microsoft.com/en-us/dotnet/framework/data/adonet/sql/linq/. [Accessed 8 8 2023]. |
| [10] | "Microsoft," [Online]. Available: https://learn.microsoft.com/en-us/visualstudio/ide/create-csharp-winform-visual-studio?view=vs-2022. [Accessed 8 8 2023]. |
| [11] | "Microsoft," [Online]. Available: https://dotnet.microsoft.com/en-us/learn/ml-dotnet/what-is-mldotnet. [Accessed 8 8 2023]. |
| [12] | "Microsoft," [Online]. Available: https://visualstudio.microsoft.com/vs/#:~:text=Visual%20Studio%20is%20the%20best,WinUI%2C%20MAUI%2C%20or%20Xamarin.. [Accessed 8 8 2023]. |
| [13] | "Microsoft," [Online]. Available: https://learn.microsoft.com/en-us/sql/ssms/sql-server-management-studio-ssms?view=sql-server-ver16. [Accessed 8 8 2023]. |

# APPENDIX



The LoadModelIntoCombo() function is responsible for checking a specific directory for machine learning model files, extracting their names, and populating these names into a combo box. This function is useful for providing users with a convenient way to select a pre-trained machine learning model for further usage or evaluation within the application.

**Video demo link:**

<https://drive.google.com/file/d/1aZo3MgE1pgaiwpNNt7-viXQKKLDfOZgA/view?usp=sharing>