**TRƯỜNG ĐẠI HỌC SƯ PHẠM KỸ THUẬT TP. HỒ CHÍ MINH**

**KHOA ĐÀO TẠO CHẤT LƯỢNG CAO**

**NGÀNH CÔNG NGHỆ THÔNG TIN**

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_



**BÁO CÁO ĐỒ ÁN 1**

**PHẦN MỀM LUYỆN NGHE TIẾNG ANH**

|  |  |
| --- | --- |
| **GVHD:** | **ThS. ĐẶNG THỊ KIM GIAO** |
| **SVTH:** | **NGUYỄN MINH HIẾU (15110047)**  **LÊ MỸ THIÊN THANH (15110124)** |

**Tp. Hồ Chí Minh, tháng 12 năm 2017**

**MỤC LỤC**

[**1.** **GIỚI THIỆU** 1](#_Toc500975296)

[***1.1.*** ***Mục tiêu*** 1](#_Toc500975297)

[***1.2.*** ***Phạm vi*** 1](#_Toc500975298)

[***1.3.*** ***Mô tả ứng dụng*** 1](#_Toc500975299)

[**1.3.1.** **Thông tin ứng dụng** 1](#_Toc500975300)

[**1.3.2.** **Chức năng cơ bản** 1](#_Toc500975301)

[**1.3.3.** **Use case diagram** 2](#_Toc500975302)

[**1.3.4.** **Phân công nhiệm vụ** 3](#_Toc500975303)

[**2.** **CƠ SỞ DỮ LIỆU** 4](#_Toc500975304)

[***2.1.*** ***Tables*** 4](#_Toc500975305)

[**2.1.1.** **Lưu trữ thông tin** 4](#_Toc500975306)

[**2.1.2.** **Database Diagram** 4](#_Toc500975307)

[**3.** **CODE** 4](#_Toc500975308)

[**4.** **GIAO DIỆN** 25](#_Toc500975309)

[**5.** **ĐÁNH GIÁ** 30](#_Toc500975310)

[***5.1.*** ***Khảo sát người chơi*** 30](#_Toc500975311)

[***5.2.*** ***Quá trình thực hiện ứng dụng*** 30](#_Toc500975312)

[***5.3.*** ***Ứng dụng*** 31](#_Toc500975313)

[***5.4.*** ***Phát triển*** 31](#_Toc500975314)

# **GIỚI THIỆU**

## ***Mục tiêu***

Học tiếng Anh không đơn giản là học ngữ pháp, từ vựng mà bạn phải biết cách giao tiếp và nói chuyện với người nước ngoài. Để giao tiếp tốt thì ngoài bổ sung từ vựng, chuẩn ngữ pháp thì kỹ năng nghe nói cũng rất quan trọng. Tuy nhiên, rất nhiều người băn khoăn không biết tại sao mình học tiếng Anh đã lâu nhưng vẫn chưa có kết quả, thậm chí nghe thường xuyên mà vẫn không cải thiện được kỹ năng nghe nói. Hiện nay có rất nhiều phương pháp và phần mềm hỗ trợ giúp bạn cải thiện kỹ năng này. Nếu bạn vẫn băn khoăn về phương pháp học thì có thể tham khảo phần mềm luyện nghe tiếng anh sau.

## ***Phạm vi***

* Level dễ, hướng tới những bé < 10 tuổi để tiếp xúc với việc nghe chuẩn xác hơn.
* Level trung bình, hướng tới những bạn trung học và đại học muốn ôn và nắm vững lại kỹ năng nghe.
* Level khó, để nâng cao và rèn luyện kỹ năng.

## ***Mô tả ứng dụng***

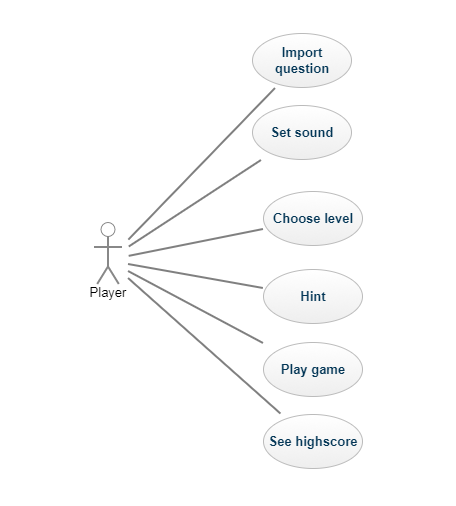
### **Thông tin ứng dụng**

* Tên ứng dụng: DA1\_DictationGame
* Ngôn ngữ lập trình: C#
* Hệ quản trị cơ sở dữ liệu: SQL Sever

### **Chức năng cơ bản**

* Phần mềm có chức năng chính là giúp người dùng phân biệt được từ ngữ theo phát âm của người bản xứ.
* Các chức năng phụ khác như so sánh điểm cao giữa các người chơi.
* Người sử dụng có thể tự thêm câu hỏi khi đã chơi hết tất cả các câu hỏi.

### **Use case diagram**



*Use case diagram mô tả hoạt động cơ bản của chương trình*

|  |  |  |
| --- | --- | --- |
| TT | Tên Actor | Ý nghĩa |
| 1 | Player | Người sử dụng chương trình |

*Bảng mô tả các actor trong diagram*

|  |  |  |
| --- | --- | --- |
| TT | Tên use case | Ý nghĩa |
| 1 | Import Question | Thêm các câu hỏi khi đã sử dụng hết các câu hỏi mặc định của chương trình. |
| 2 | Set sound | Bật tắt chế độ nhạc nền của chương trình. |
| 3 | Choose level | Chọn cấp độ khó của chương trình. |
| 4 | Hint | Gợi ý cho gói câu hỏi đang sử dụng. |
| 5 | Play game | Bắt đầu chơi game. |
| 6 | See highscore | So sánh điểm cao giữa các người sử dụng. |

*Bảng mô tả các use case trong diagram*

* + 1. **Phân công nhiệm vụ**

|  |  |  |
| --- | --- | --- |
| Sinh viên | Phần trăm đóng góp | Công việc |
| Nguyễn Minh Hiếu | 50% | * Thiết kế giao diện * Viết báo cáo * Xử lý code “Import Question”, “Choose Level” * Tạo cơ sở dữ liệu |
| Lê Mỹ Thiên Thanh | 50% | * Tạo cơ sở dữ liệu * Xử lý code “Play Game”, “Sound”, “HighScore” * Tổng hợp và chỉnh sửa bài |

# **CƠ SỞ DỮ LIỆU**

Cơ sở dữ liệu được thiết kế dựa trên hệ quản trị SQL Sever giúp cho việc quản lý dễ dàng hơn. Cơ sở dữ liệu cho hệ thống quản lý có tên “DA1\_DictationGame” với 2 bảng.

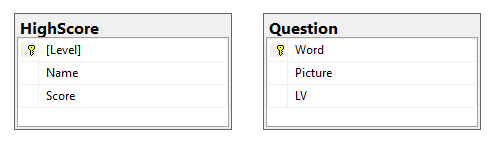
## ***Tables***

### **Lưu trữ thông tin**

|  |  |  |  |
| --- | --- | --- | --- |
| TT | | Tên bảng/Tên trường | Mục đích bảng/Mục đích trường |
| 1 | | Question | Lưu trữ gói câu hỏi. |
|  | 1.1 | Word | Câu hỏi. |
| 1.2 | Picture | Hình ảnh gợi ý cho câu hỏi. |
| 1.3 | LV | Độ khó của câu hỏi. |
|  | | | |
| 2 | | Hight Score | Lưu trữ điểm cao nhất của các cấp độ. |
|  | 2.1 | Level | Cấp độ. |
| 2.2 | Name | Tên người sử dụng cao nhất. |
| 2.3 | Score | Điểm cao nhất. |
|  | | | |

*Danh sách các bảng và thông tin cần lưu trữ và quản lý của ứng dụng*

### **Database Diagram**



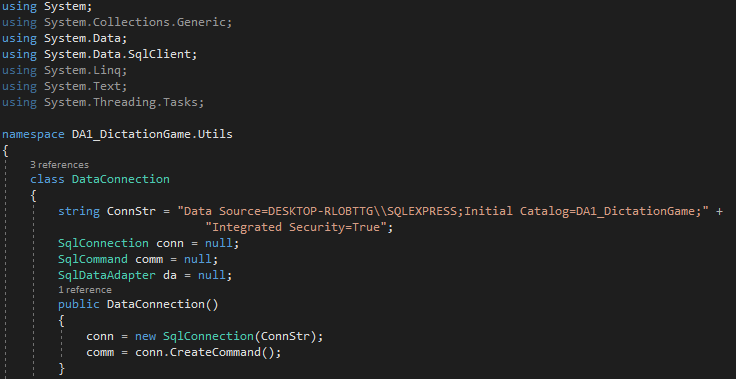
*Database Diagram được vẽ trong SQL Sever*

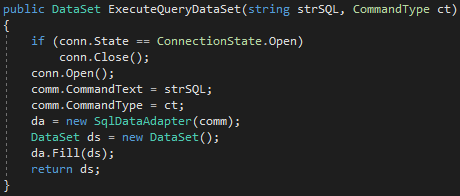
# **CODE**

Đồ án được viết trên nền tảng C#, theo mô hình ba lớp.

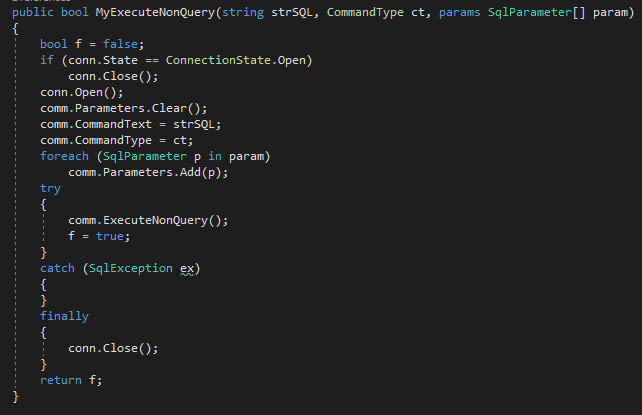


Ở folder Utils, C# sẽ lấy kết nối đến SQL

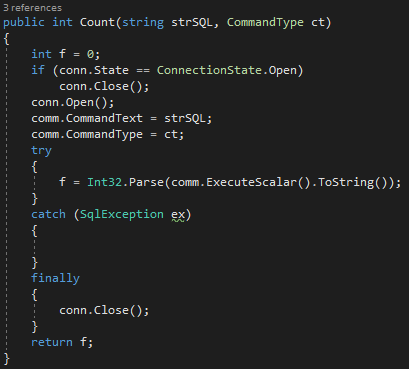




Phương thức ExecuteQueryDataSet(), để lấy dữ liệu từ cơ sở dữ liệu đổ về Dataset sử dụng trong C#.



Phương thức MyExecuteNonQuery(), để xử lý các câu truy vấn

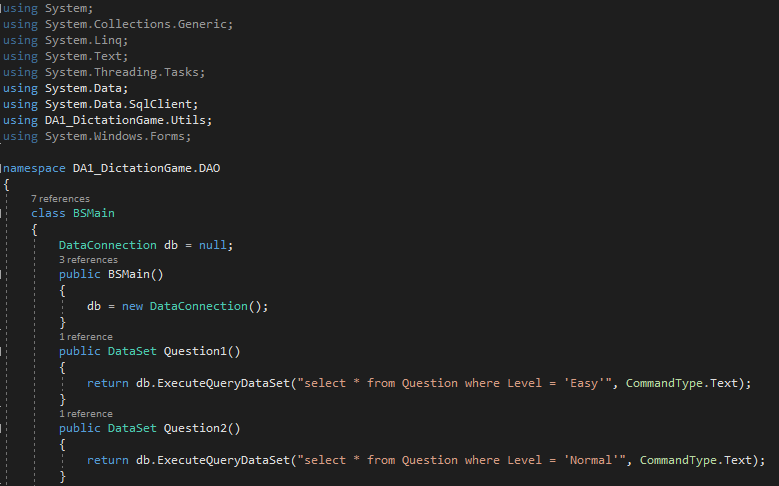


Phương thức Count(), để xử lý câu truy vấn nhưng có trả về dữ liệu kiểu int.

**Lớp DAO (Data Access Object)**

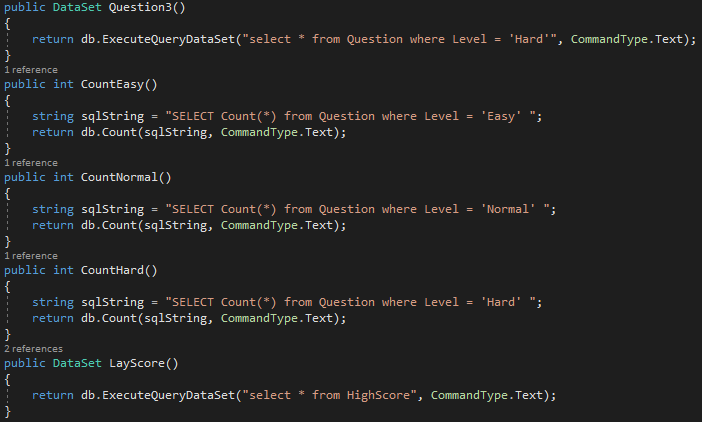


Ở lớp này, sẽ chứa những câu Query để xử lý dữ liệu theo yêu cầu.



Phương thức Question1() có nhiệm vụ trả về dữ liệu trong cơ sở dữ liệu từ bảng Question, sao cho có mức độ là “Easy”. Phương thức này sẽ lấy các câu hỏi mức độ dễ từ cơ sở dữ liệu.

Phương thức Question2() có nhiệm vụ trả về dữ liệu trong cơ sở dữ liệu từ bảng Question, sao cho có mức độ là “Normal”. Phương thức này sẽ lấy các câu hỏi mức độ trung bình từ cơ sở dữ liệu.



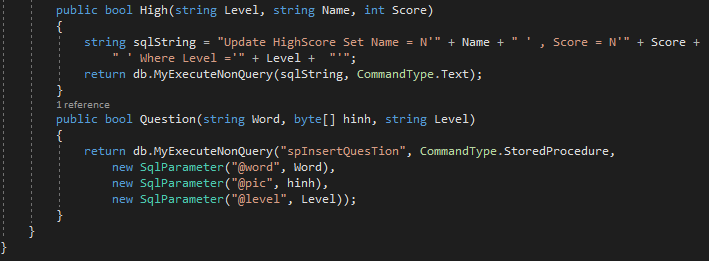
Phương thức Question3() có nhiệm vụ trả về dữ liệu trong cơ sở dữ liệu từ bảng Question, sao cho có mức độ là “Hard”. Phương thức này sẽ lấy các câu hỏi mức độ khó từ cơ sở dữ liệu.

Phương thức CountEasy() có nhiệm vụ đếm số câu hỏi có trong bảng Question sao cho mức độ câu hỏi là “Easy” và trả về giá trị int

Phương thức CountNormal() có nhiệm vụ đếm số câu hỏi có trong bảng Question sao cho mức độ câu hỏi là “Normal” và trả về giá trị int

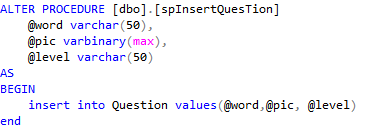
Phương thức CountHard() có nhiệm vụ đếm số câu hỏi có trong bảng Question sao cho mức độ câu hỏi là “Hard” và trả về giá trị int

Phương thức LayScore() có nhiệm vụ trả về dữ liệu trong cơ sở dữ liệu từ bảng HighScore.

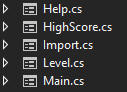


Phương thức High() có nhiệm vụ cập nhật dữ liệu trong bảng HighScore theo level. Phương thức này có chức năng cập nhật điểm số theo từng cấp độ.

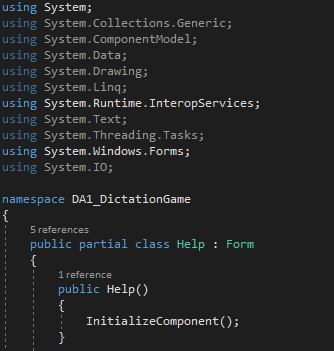
Phương thức Question() có nhiệm vụ thêm câu hỏi vào bảng Question. Phương thức này sử dụng 1 stored procedure và truyền tham số vào để thực thi. Cấu trúc của procedure này:

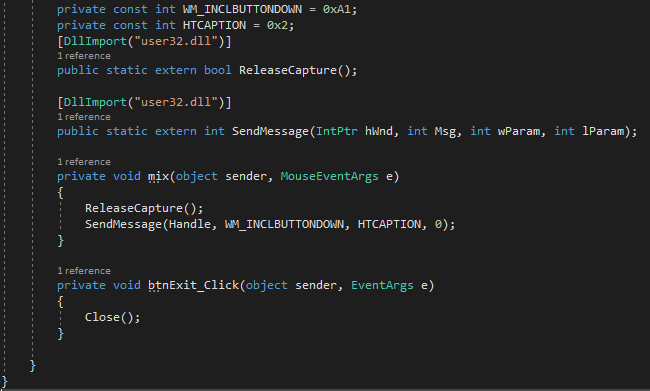


**Lớp GUI, BLL (Presentation Layers & Business Logic Layers)**



* **Class Help**

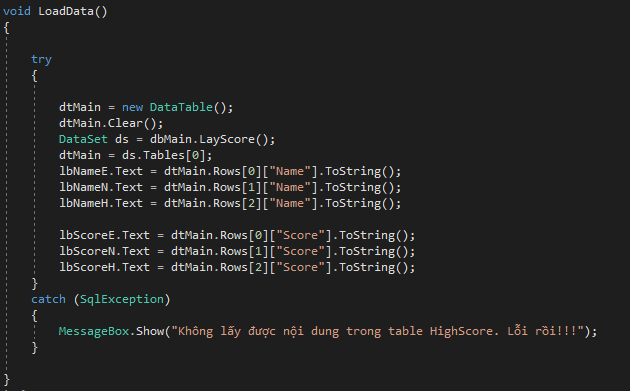




Trong class này, sử dụng code để di chuyển form không có border (nguồn google).

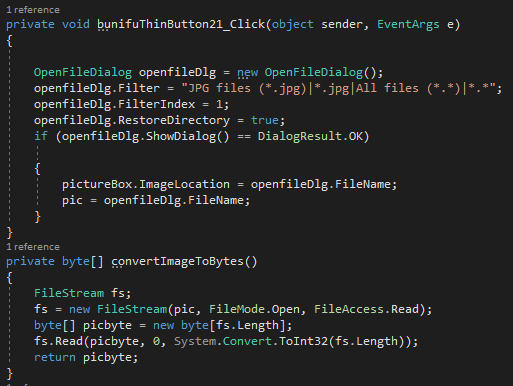
Và set chức năng đóng form cho nút thoát.

* **Class HighScore**



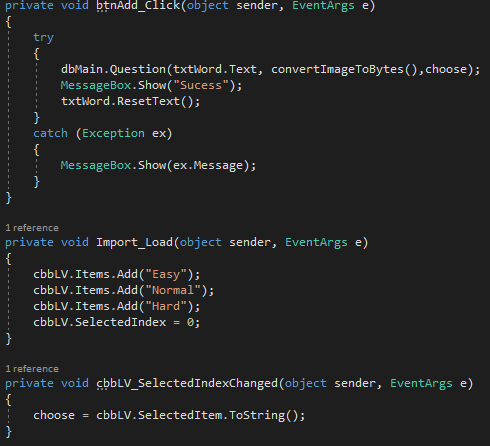
Trong class này, sẽ có những phương thức như class Help. Thêm vào đó, sẽ có thêm phương thức LoadData(). Đoạn code này có nhiệm vụ lấy điểm từ bảng điểm bằng cách gọi LayScore() từ lớp DAO, sau đó set các lable theo dữ liệu để show lên giao diện.

* **Class Import**



Ở class này, bắt sự kiện cho bunifuThinButton21. Khi click vào button này, sẽ mở vào file dialog, dùng để chọn hình sử dụng trong gợi ý

Phương thức convertImageToBytes() dùng để đổi hình ảnh sang mảng byte, dùng để lưu hình ảnh vào cơ sở dữ liệu.

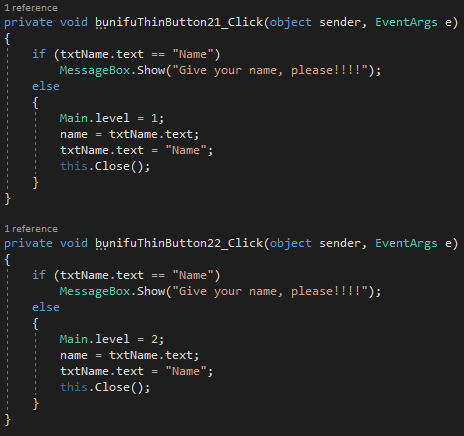


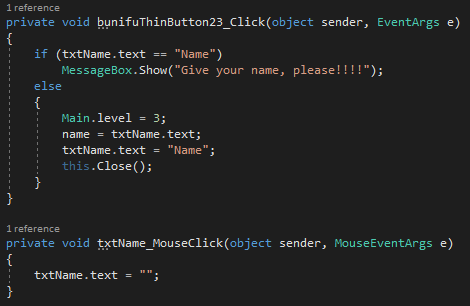
Bắt sự kiện cho btnAdd. Khi click vào button này, câu hỏi sẽ được thêm vào cơ sở dữ liệu, bằng cách gọi Question() ở lớp DAO, và truyền tham số thích hợp theo kiểu dữ liệu đã set trước. Thêm vào đó, dùng câu lệnh try catch để bắt lỗi cho công việc này.

Đầu tiên, khi form Import được load lên, sẽ set các biến như trong hình.

Bắt sự kiện cho combobox cbbLV. Khi dữ liệu ở combobox này được thay đổi, biến choose sẽ có giá trị của combobox này. Ý nghĩa của hành động này, là lấy giá trị mức độ để truyền vào việc thêm câu hỏi.

* **Class Level**

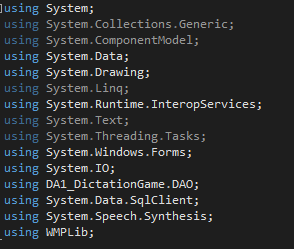




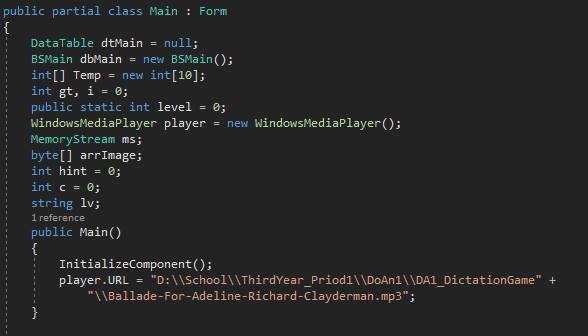
Ở class này, bắt sự kiện cho bunifuThinButton21, bunifuThinButton22, bunifuThinButton23. Khi click vào các button này, code sẽ kiểm tra việc nhập tên của người dùng. Nếu tên trùng với tên mặc định “Name”, ứng dụng sẽ yêu cầu người chơi nhập tên của mình. Việc nhập tên này sẽ giúp ứng dụng tự cập nhật điểm số của người chơi, nếu họ vượt qua kỉ lục trước đó. Sau khi đã nhập xong tên, giá trị biến level sẽ có giá trị, việc này có ý nghĩa đối với việc tải game theo mức độ vừa chọn, tương ứng với mức dễ là 1, bình thường là 2 và khó là 3. Biến name cũng sẽ lưu giá trị theo textbox vừa được nhập. Sau bước này, người dùng đã có thể chơi game.

Ngoài ra, ở class này, còn có bắt sự kiện cho textbox. Khi nhấn vào txtName, phần nội dung bên trong sẽ được xóa tất cả, điều này giúp tiện lợi hơn cho người chơi khi nhập tên.

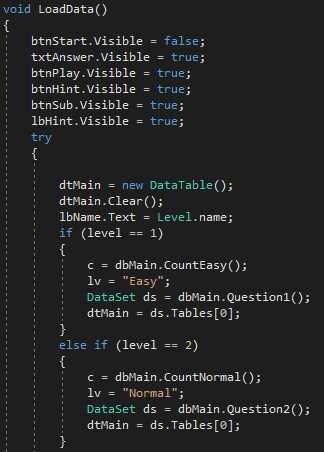
* **Class Main**

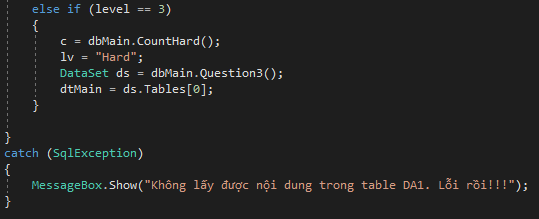


Trong class này, nhóm đã sử dụng thêm một số thư viện để hỗ trợ cho công cụ Speak Text và phần âm nhạc cho sinh động hơn.



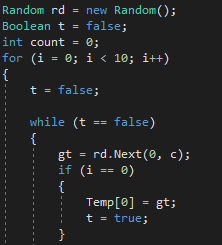
Đầu tiên là khai báo tất cả các biến cần để sử dụng. Trong đó Temp[] là mảng gồm 10 số nguyên được random ngẫu nhiên, mục đích của việc khai báo biến này là để chứa danh sách ngẫu nhiên của các câu hỏi để không bị trùng lặp giữa các lần chơi khác nhau; dtMain và dbMain là biến để lấy các câu hỏi từ cơ sở dữ liệu; player là giá trị dùng cho âm nhạc của ứng dụng; ms và arrImage[] là biến để xử lý ảnh sang dạng nhị phân. Trong phần đầu ứng dụng, sẽ set đường dẫn vào file âm thanh.

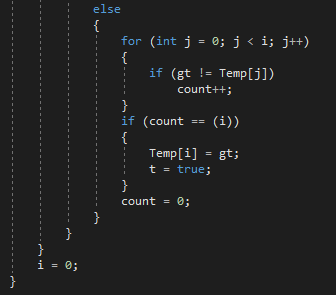




Đoạn code này có chức năng làm txtAnswer, btnPlay, btnHintt, btnSub, lbHint hiện lên và ẩn btnStart đi. Sau đó lbName sẽ có giá trị là biến name từ form Level, và chương trình sẽ được load theo cấp độ đã chọn tương ứng những con số 1,2,3.

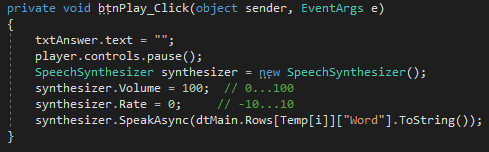
Biến c là biến lưu số câu hỏi theo cấp độ vừa chọn có trong database, bằng cách gọi phương thức CountEasy()/CountNormal()/CountHard() ở lớp DAO.



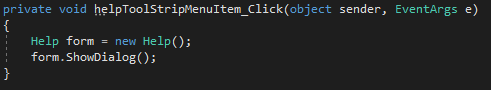


Đoạn code trên thực hiện việc random mảng Temp[] sao cho các giá trị trong mảng là không trùng nhau. Cơ chế hoạt động của đoạn code này như sau:

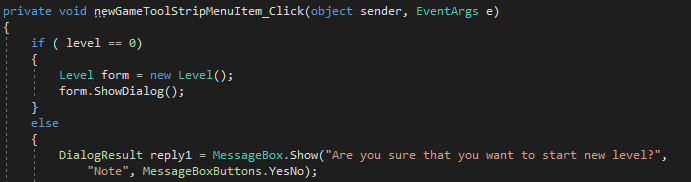
* Vòng lặp sẽ chạy 10 lần tương đương với 10 giá trị của mảng
* Gắn biến t= false và thực hiện vòng lặp while, gt bằng giá trị random ngẫu nhiên trong khoảng từ 0 tới c (với c là tổng số câu hỏi của level mà bạn đã chọn, c sẽ được truyền trước ở đoạn code trên).
* Nếu là giá trị đầu tiên của mảng, thì sẽ gắn ngay Temp[0]= gt và thoát khỏi vòng lặp while
* Tăng giá trị i và thực hiện vòng lặp for,
* Nếu không phải là giá trị đầu tiên của mảng thì trong vòng lặp while sẽ nhảy vào else, duyệt từ đầu mảng đến vị trí thứ i, và đếm xem có bao nhiêu giá trị trong mảng khác gt.
* Nếu count = i (nghĩa là toàn bộ giá trị trong mảng đều khác gt) thì gán Temp[i] = gt và thoát khỏi vòng lặp while, ngược lại thì sẽ lặp vòng while cho đến khi nào tìm được 1 giá trị không trùng với những phần tử trong mảng đứng trước nó.
* Sau khi thực hiện toàn bộ sẽ thu được mảng Temp[] 10 phần tử khác nhau từng đôi một.
* Sau cùng, set i=0



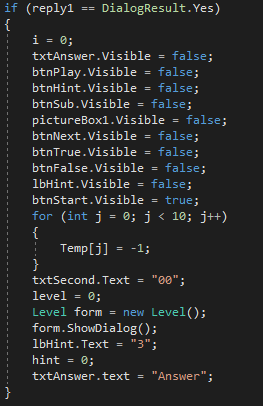
Bắt sự kiện cho btnPlay.Khi click vào button này, txtAnswer sẽ xóa hết chữ, nhạc sẽ tạm dừng và ứng dụng sẽ đọc từ lấy từ cơ sở dữ liệu lên (trong đó Temp[i] là giá trị đã random từ trước)



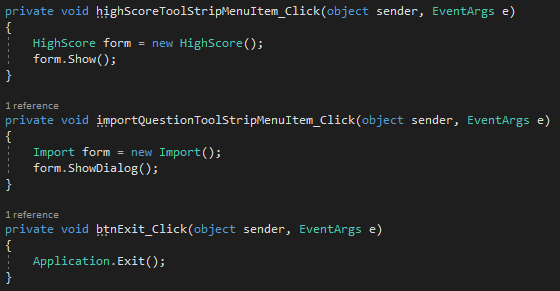
Khi click vào helpToolStripMenuItem, form Help sẽ hiện ra



Khi click vào newGameToolStripMenuItem, ứng dụng sẽ kiểm tra level có bằng 0 không (kiểm tra xem có đang chơi game không), nếu không đang chơi sẽ hiện form chọn level. Nếu đăng chơi game thì sẽ hiện lên thông báo để xác thực lại có muốn bắt đầu game mới hay không.



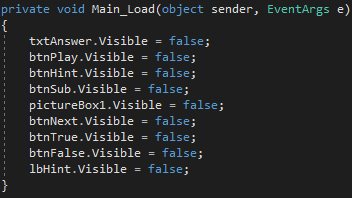
Nếu người dùng chọn có, set các textbox, label, button trên ẩn đi và set các giá trị về giá trị ban đầu như khi chưa bắt đầu game.



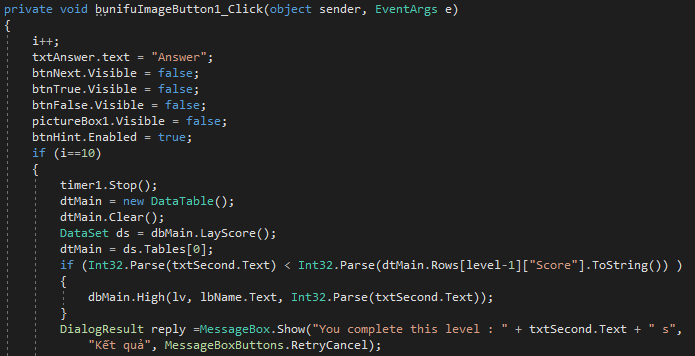
Bắt sự kiện cho highScoreToolStripMenuItem, khi click vào thì form HighScore sẽ hiện ra

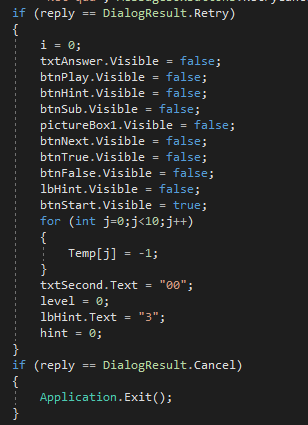
Bắt sự kiện cho importQuestionToolStripMenuItem, khi click vào thì form Import sẽ hiện ra

Bắt sự kiện cho btnExit, khi click vào thì sẽ đóng ứng dụng



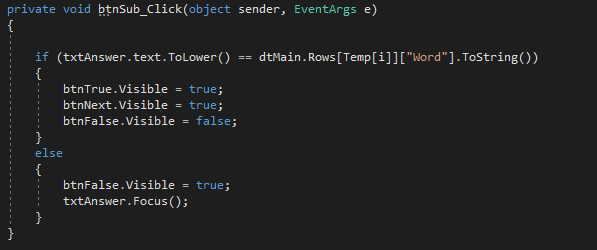
Vì form chính khi khởi động ứng dụng và form chơi game là một, nên khi load form thì sẽ ẩn đi những thành phần chưa cần sử dụng từ lúc đầu.



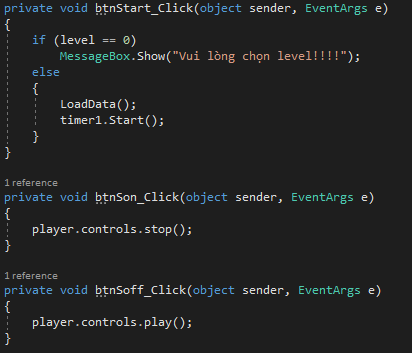


Bắt sự kiện cho bunifuImageButton1, khi click vào i sẽ tăng lên, điều này có ý nghĩa là chương trình sẽ qua câu hỏi kế tiếp. Trong trường hợp nếu i=10, thời gian người chơi chơi sẽ dừng lại và ứng dụng sẽ lấy điểm cao theo level mà người chơi đã chơi và bắt đầu so sánh bằng cách gọi phương thức LayScore() ở lớp DAO. Nếu thời gian của người chơi là ít hơn so với kỷ lục, kỷ lục sẽ được cập nhật.

Sau đó, chương trình sẽ thông báo người chơi có muốn tiếp tục chơi hay không, nếu người chơi muốn tiếp tục thì sẽ set lại các giá trị như ban đầu. Còn nếu không, ứng dụng sẽ thoát.



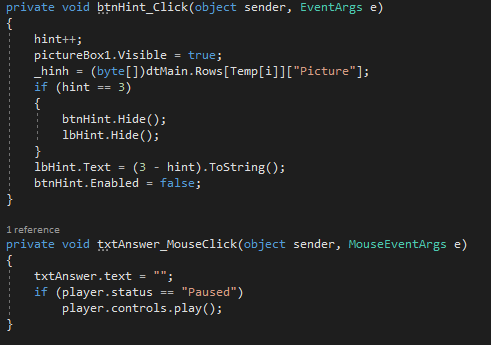
Bắt sự kiện cho btnSub, khi click vào sẽ so sánh các ký tự mà người chơi nhập với từ được tải từ cơ sở dữ liệu. Nếu trùng nhau, các button cần thiết sẽ hiện ra để qua câu tiếp theo. Nếu không, btnFalse sẽ hiện ra để cho người chơi biết mình đã sai và cần nhập câu trả lời lại.



Bắt sự kiện cho btnStart, khi click vào, nếu level = 0 (chưa chọn level) thì sẽ hiện ra thông báo cho người dùng cần chọn level trước khi chơi. Nếu level khác 0, dữ liệu sẽ được load lên bằng cách gọi LoadData() và thời gian sẽ bắt đầu được tính.

Bắt sự kiện cho btnSoff, khi click vào, nhạc sẽ phát.

Bắt sự kiện cho btnSon, khi click vào, nhạc sẽ dừng.

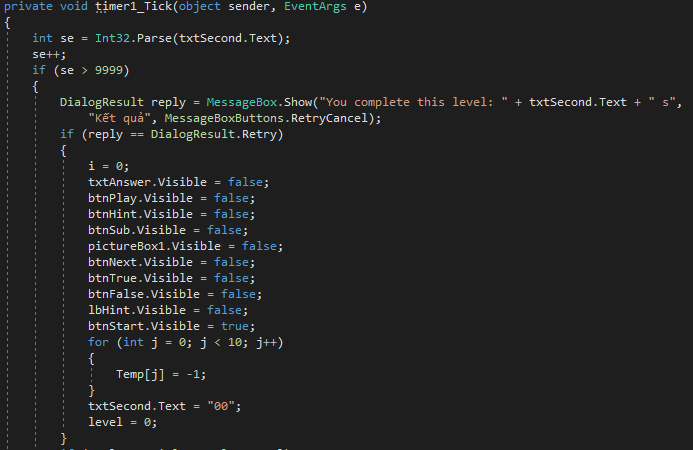


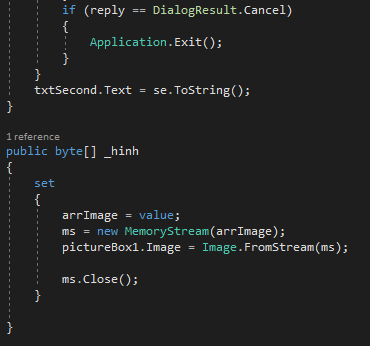
Bắt sự kiện cho btnHint, khi click vào, biến Hint sẽ cộng 1, picture box sẽ hiện ra và nhận giá trị lở cột Picture với dòng tương ứng từ cơ sở dữ liệu.

Kiểm tra nếu hint bằng 3 ( vì giới hạn của số lần gợi ý là 3), btnHint và lbHint sẽ ẩn đi

Sau mỗi lần click, thì lbHint sẽ cập nhật và giảm đi 1.

Bắt sự kiện cho txtAnswer, khi click vào, xóa chữ trong textbox và nếu nhạc đang ở trạng thái tạm dừng thì sẽ tiếp tục phát.

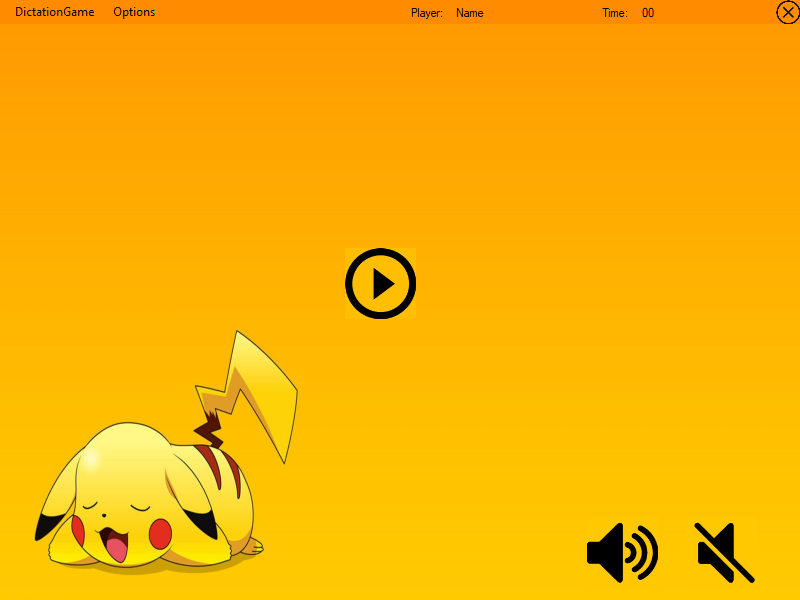




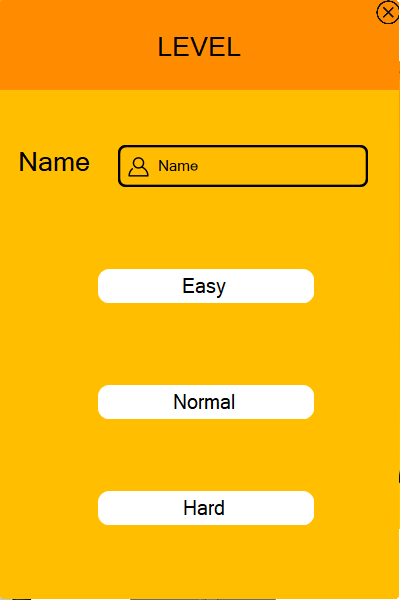
Phương thức timer1\_Tick dùng để tính thời gian người dùng chơi. Game sẽ dừng khi số giây vượt quá 9999. Và sau đó, người chơi cũng có 2 sự lựa chọn tiếp tục hoặc dừng lại như khi đã vượt qua 1 lượt game.

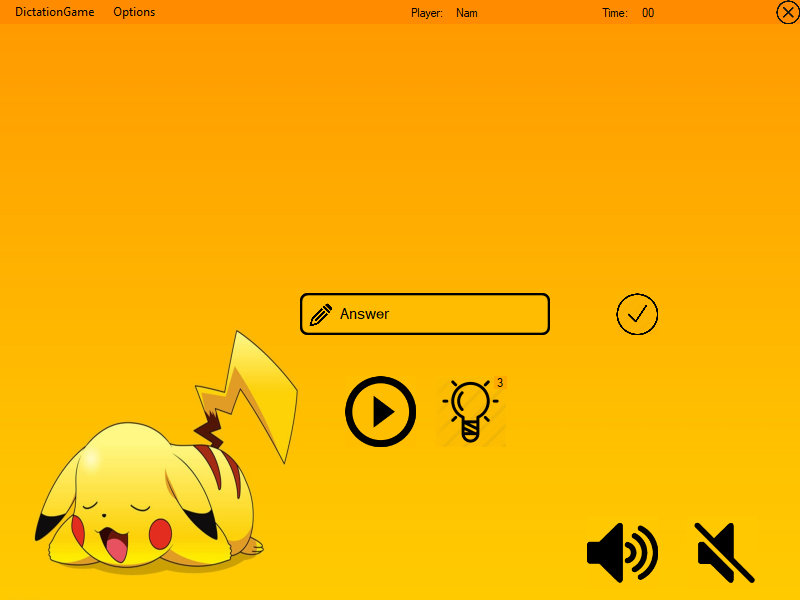
Phương thức \_hinh có chức năng lấy giá trị hình và đưa vào pictureBox1, nó được sử dụng cho chức năng hint.

# **GIAO DIỆN**

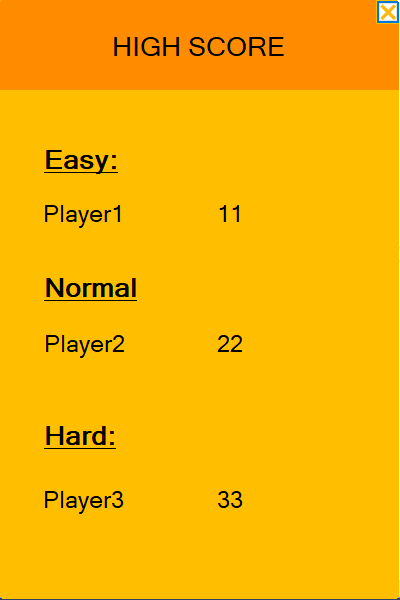


*Giao diện khi mở chương trình*.

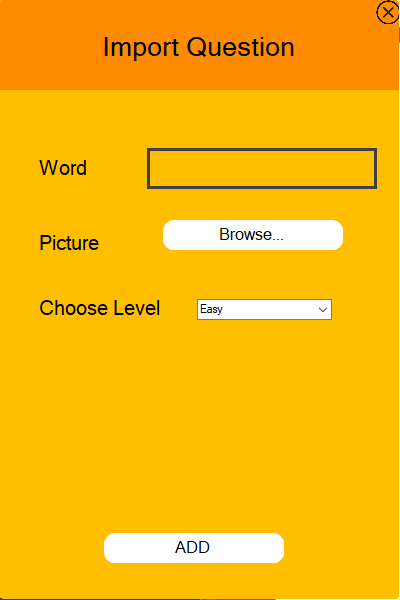


*Giao diện chọn cấp độ khó và điền tên người sử dụng. *

*Giao diện khi chơi game.*

**

*Giao diện xem điểm cao.*

**

*Giao diện thêm câu hỏi.*

# **ĐÁNH GIÁ**

* 1. ***Khảo sát người chơi***

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Độ tuổi | Số lượng | Giao diện | Tính năng | Độ hài lòng  (Điểm trung bình) |
| 10 | 2 | * Dễ thương và bắt mắt * Dễ sử dụng * Tuy nhiên, thỉnh thoảng vẫn chưa ổn định | * Giúp bé cảm thấy thoải mái và thích thú ở mức độ dễ * Vì có gợi ý nên vẫn khiến bé ỷ lại vào sự trợ giúp | * 9/10 |
| 20-22 | 6 | * Giao diện đơn giản, dễ sử dụng | * Rèn luyện kỹ năng nghe của người chơi * Sự chênh lệch giữa những mức độ khá lớn khiến người chơi không nắm bắt kịp | * 8/10 |
| 25-30 | 2 | * Giao diện đơn giản * Nhưng chưa thực sự bắt mắt | * Giúp cải thiện khả năng nghe cũng như trao dồi thêm từ vựng * Rất thích chinh phục cấp độ cao nhất | * 8/10 |

## ***Quá trình thực hiện ứng dụng***

**Thuận lợi:**

* Nhiều công nghệ hỗ trợ cho việc phát triển ứng dụng
* Nhiều thư viện (API) tiện ích

- Nguồn tài liệu phong phú

* Bản thân sinh viên có kiến thức cơ bản về lập trình C#
* Thời gian thiết kế và phát triển ứng dụng nhiều.

**Khó khăn:**

* Ít kinh nghiệm trong việc thiết kế giao diện lẫn cơ sở dữ liệu
* Chưa tận dụng, tối ưu chức năng các thư viện, công nghệ.

## ***Ứng dụng***

**Ưu điểm:**

* Có những chức năng cơ bản

**Nhược điểm:**

* Còn lỗi
* Cơ sở dữ liệu chưa ràng buộc chặt chẽ
* Load dữ liệu còn chậm.

## ***Phát triển***

* Thêm mới các bảng
* Ràng buộc chặt chẽ hơn
* Mở rộng chức năng
* Cải tiến giao diện
* Phát triển ứng dụng trên mobile.