

**BỘ CÔNG THƯƠNG**

**TRƯỜNG ĐẠI HỌC CÔNG NGHIỆP TP.HCM**

**Tìm Hiểu Công Nghệ WPF và viết ứng dụng học tiếng anh cho trẻ em**

# MỤC LỤC

[MỤC LỤC 1](#_Toc35268713)

[DANH MỤC CÁC HÌNH VẼ 3](#_Toc35268714)

[DANH MỤC CÁC BẢNG BIỂU 3](#_Toc35268715)

[LỜI MỞ ĐẦU 4](#_Toc35268716)

[CHƯƠNG 1 : GIỚI THIỆU 6](#_Toc35268717)

[1.1 Tổng quan 6](#_Toc35268718)

[1.2 Mục tiêu đề tài 6](#_Toc35268719)

[1.3 Phạm vi đề tài 6](#_Toc35268720)

[1.3.1 Tìm hiểu công nghệ WPF 6](#_Toc35268721)

[1.3.2 Hiện thực ứng dụng học tiếng anh cho trẻ em bằng WPF 6](#_Toc35268722)

[1.4 Mô tả yêu cầu chức năng 7](#_Toc35268723)

[CHƯƠNG 2 : CƠ SỞ LÝ THUYẾT 9](#_Toc35268724)

[2.1 Công nghệ WPF 9](#_Toc35268725)

[2.1.1 Giới thiệu về WPF 9](#_Toc35268726)

[2.1.2 Mục tiêu của WPF 9](#_Toc35268727)

[2.1.3 Ngôn ngữ XAML 9](#_Toc35268728)

[2.1.4 WPF cho Window và trình duyệt Web 11](#_Toc35268729)

[2.1.5 Các thành phần của WPF 12](#_Toc35268730)

[2.2 Cơ sở dữ liệu SQL Server 19](#_Toc35268731)

[2.2.1 Giới thiệu về SQL Server 19](#_Toc35268732)

[2.2.2 Lịch sử phát triển của SQL Server 19](#_Toc35268733)

[CHƯƠNG 3 : PHÂN TÍCH 21](#_Toc35268734)

[3.1 Phân tích yêu cầu bằng UML 21](#_Toc35268735)

[3.1.1 Usecase Tổng Quát 21](#_Toc35268736)

[3.1.2 Danh sách các actor 22](#_Toc35268737)

[3.1.3 Danh sách các Usecase 22](#_Toc35268738)

[3.2 Đặc tả các yêu cầu chức năng 23](#_Toc35268739)

[3.2.1 UseCase UC1: HocTuVungQuaVideo 23](#_Toc35268740)

[3.2.2 UseCase UC2: HocTuVungQuaTuDien 24](#_Toc35268741)

[3.2.3 UseCase UC3: ThuThapBauVat 25](#_Toc35268742)

[3.2.4 UseCase UC5: BaoVeBauVat 26](#_Toc35268743)

[3.2.5 UseCase UC6: LuyenTap 28](#_Toc35268744)

[3.2.6 UseCase UC6: XemBoSuuTap 29](#_Toc35268745)

[3.2.7 UseCase UC7: HuongDanSuDung 30](#_Toc35268746)

[CHƯƠNG 4 : THIẾT KẾ VÀ HIỆN THỰC 31](#_Toc35268747)

[4.1 Tiêu đề mục 31](#_Toc35268748)

[4.1.1 Tiêu đề tiểu mục 31](#_Toc35268749)

[4.1.2 Tiêu đề tiểu mục 31](#_Toc35268750)

[4.2 Tiêu đề mục 31](#_Toc35268751)

[4.2.1 Tiêu đề tiểu mục 31](#_Toc35268752)

[4.2.2 Tiêu đề tiểu mục 31](#_Toc35268753)

[CHƯƠNG 5 : KẾT LUẬN 32](#_Toc35268754)

[5.1 Kết quả đạt được 32](#_Toc35268755)

[5.2 Hạn chế của đồ án 32](#_Toc35268756)

[5.3 Hướng phát triển 32](#_Toc35268757)

[TÀI LIỆU THAM KHẢO 32](#_Toc35268758)

[PHỤ LỤC 33](#_Toc35268759)

# DANH MỤC CÁC HÌNH VẼ

[Hình 1‑1 Thao tác cập nhật mục lục **Error! Bookmark not defined.**](#_Toc262311533)

[Hình 1‑2 Cách chèn nhãn cho hình **Error! Bookmark not defined.**](#_Toc262311534)

[Hình 1‑3 Cách tạo một nhãn mới **Error! Bookmark not defined.**](#_Toc262311535)

[Hình 1‑4 Cách tham chiếu đến một nhãn **Error! Bookmark not defined.**](#_Toc262311536)

# DANH MỤC CÁC BẢNG BIỂU

[Bảng 1: Danh sách các Actor 22](#_Toc35268697)

[Bảng 2: Danh Sách Các Usecase chức năng 23](#_Toc35268698)

[Bảng 3: Đặc tả UseCase HocTuVungQuaVideo 24](#_Toc35268699)

[Bảng 4: Đặc tả UseCase HocTuVungQuaTuDien 25](#_Toc35268700)

[Bảng 5: Đặc tả UseCase ThuThapBauVat 26](#_Toc35268701)

[Bảng 6: Đặc tả UseCase BaoVeBauVat 28](#_Toc35268702)

[Bảng 7: Đặc tả UseCase LuyenTap 29](#_Toc35268703)

[Bảng 8: Đặc tả UseCasw XemBoSuuTap 29](#_Toc35268704)

[Bảng 9: Đặc tả UseCase HuongDanSuDung 30](#_Toc35268705)

# LỜI MỞ ĐẦU

1. Tổng quan tình hình nghiên cứu thuộc lĩnh vực của đề tài

Trong những năm gần đây với sự phát triển vượt trội của Công Nghệ Thông Tin, những ứng dụng của Công Nghệ Thông Tin vào khắp các lĩnh vực đời sống đã đóng góp to lớn cho sự phát triển của nhân loại về nhiều mặt. Việc đưa tin học vào các trường học, nhà máy, công ty, các đơn vị tổ chức, doanh nghiệp...và thậm chí đến từng gia đình thay cho việc quản lý bằng tay đã tiết kiệm được thời gian, công sức và tiền của đem lại hiệu quả cao giúp phát triển xã hội vững mạnh hơn, giàu đẹp hơn. Đặc biệt là việc phổ cập tin học trong nhà trường đang dần được triển khai đã giúp giới trẻ ngày nay không còn lạ lẫm với những con người, đất nước trên thế giới nữa. Tin học đã thật sự đưa đất nước ta đến gần với các nước tiên tiến trên thế giới.

Đến thời điểm hiện tại, những ứng dụng của tin học không chỉ dừng ở lại đó mà nó dần được nâng cao hơn nữa, nhằm đáp ứng nhu cầu sử dụng tin học ngày càng cao của con người. Ngày nay, trong các ứng dụng hiện đại, giao diện người dùng trực quan chiếm vị trí hết sức quan trọng, nó không chỉ đóng vai trò**là cổng giao tiếp để người dùng thực hiện các chức năng của ứng dụng mà giao diện ngày càng được nâng tầm, dần trở thành công cụ hỗ trợ đắc lực, giúp người dùng dễ dàng thấy được sự liên hệ giữa các số liệu, có cái nhìn xuyên suốt, trực quan, sinh động với các thao tác thuận tiện, dễ dàng để nhanh chóng đưa ra quyết định quan trọng.**

**Ví dụ như trong trường hợp một ứng dụng học tiếng anh cho trẻ em, một giao diện trực quan, sinh động với nhiều hiệu ứng bắt mắt đi kèm những âm thanh, hình ảnh, video thú vị sẽ góp phần rất lớn vào việc thu hút trẻ em sử dụng ứng dụng để tiếp thu kiến thức mà không bị nhàm chán.**

Với công nghệ trước đây, một ứng dụng với các yêu cầu như vậy thật sự là thách thức đòi hỏi rất nhiều tài nguyên và công sức, và WPF ra đời chính là để xây dựng một nền tảng chung giúp cho việc xây dựng các ứng dụng trở nên trực quan và thuận tiện hơn bao giờ hết.

Cũng ví lí do đó mà chúng em đã chọn đề tài “Tìm hiểu về công nghệ WPF và xây dựng ứng dụng học tiếng anh cho trẻ em”.

2. Ý nghĩa khoa học và thực tiễn của đề tài

Có cơ hội tiếp cận, tìm hiểu một công nghệ mới để xây dựng một ứng dụng, hiểu rõ những điểm mạnh, điểm yếu của WPF và so sánh với Winform.

Củng cố kiến thức của bản thân về lập trình, về thiết kế ứng dụng và kiểm thử.

Thiêt kế và xây dựng một ứng dụng học tiếng anh cho trẻ em với tiêu chuẩn công nghiệp và có khả năng đưa vào áp dụng trong thực tế

# : GIỚI THIỆU

## Tổng quan

## Mục tiêu đề tài

Hiểu được cách thức lập trình với WPF.

Vận dụng kiến thức đã học về lập trình C#, kiến thức về thiết kế, kiểm thử, quản lý dự án để xây dựng ứng dụng WPF theo tiêu chuẩn công nghiệp.

## Phạm vi đề tài

### Tìm hiểu công nghệ WPF

Tìm hiểu về WPF: Giao diện XAML, các control cơ bản trong wpf, tích hợp âm thanh, đồ họa, media,…

Tìm hiểu và sử dụng thư viện hỗ trợ đồ họa Material Design.

### Hiện thực ứng dụng học tiếng anh cho trẻ em bằng WPF

Ứng dụng học tiếng anh cho trẻ em cung cấp phương pháp học tiếng anh thông qua video bài giảng và trả lời câu hỏi trắc nghiệm với giao diện hoạt hình thú vị cùng cốt truyện nhập vai đặc sắc giúp trẻ không nhàm chán khi sử dụng.

Người dùng sẽ nhập vai vào một nhà sưu tập báu vật trên khắp thế giới và mỗi báu vật được canh giữ bởi một Người bảo vệ báu vật. Để giành được báu vật, bạn cần phải vượt qua thử thách của họ.

Thử thách của những Người bảo vệ sẽ bao gồm các câu hỏi trắc nghiệm về từ vựng tiếng anh thuộc các chủ đề đặc sắc. Trước khi tham gia thử thách, bạn sẽ có thời gian để học từ vựng và chuẩn bị cho thử thách khó khăn sắp tới. Ôn tập thật kỹ và vượt qua thử thách để giành lấy báu vật sẽ là nhiệm vụ quan trọng nhất của bạn trong trò chơi này.

Tuy nhiên những Người bảo vệ báu vật không thể bỏ qua khi bạn cướp báu vật của họ, họ sẽ luôn chực sắn để đoạt lại thứ mà bạn đã giành lấy từ tay họ. Lúc này bạn sẽ phải tiếp tục đối mặt với thách thức của họ để bảo vệ bộ sưu tập của mình.

Chú thích:

* Mỗi chủ đề (bài học) sẽ tương đương với 1 báu vật, hoàn thành một chủ đề để thu thập báu vật.
* Việc học từ vựng sẽ thông qua video bài giảng hoạt hình, hoặc có thể ôn tập theo dạng từ điển bao gồm việc hiển thị hình ảnh, phiên âm, dịch nghĩa, âm thanh đọc của từng từ vựng.
* Thử thách thu thập báu vật: những câu hỏi từ vựng được hỏi theo thứ tự ngẫu nhiên, trả lời đúng tất cả các câu hỏi để hoàn thành thử thách, nếu trả lời sai sẽ chuyển sang câu tiếp theo và câu hỏi sai sẽ tiếp tục được hỏi sau đó cho đến khi hoàn thành thử thách.
* Thử thách bảo vệ báu vật (mỗi ngày có 1 thử thách): những câu hỏi từ vựng được hỏi theo thứ tự ngẫu nhiên, trả lời tất cả các câu hỏi lần để bảo vệ báu vật, nếu trả lời sai sẽ chuyển sang câu tiếp theo và câu hỏi sai sẽ tiếp tục được hỏi sau đó cho đến khi hoàn thành thử thách ( câu hỏi đã đúng 3 lần sẽ không được hỏi lại). Nếu có 1 câu hỏi bất kì sai quá 3 lần thì thử thách của bạn thất bại và báu vật sẽ bị cướp đi. Bạn có thể gặp lại Người bảo vệ báu vật đó để tham gia thử thách giành lại báu vật sau đó.

## Mô tả yêu cầu chức năng

* Yêu cầu chức năng:
  + Chức năng học từ vựng thông qua video.
  + Chức năng học từ vựng thông qua từ điển (hiển thị hình ảnh, phiên âm, dịch nghĩa, âm thanh đọc của từng từ vựng).
  + Chức năng Thử thách thu thập báu vật (trắc nghiệm từ vựng của một chủ để mới).
  + Chức năng Thử thách bảo vệ báu vật (trắc nghiệm từ vựng của chủ đề đã hoàn thành trước đó).
  + Chức năng luyện tập (trắc nghiệm từ vựng ngẫu nhiên trong danh sách các từ đã học).
  + Chức năng xem bộ sưu tập báu vật đã thu thập được.
  + Hẹn giờ nhắc nhở mỗi ngày.
  + Chức năng hướng dẫn sử dụng.
* Yêu cầu phi chức năng:
  + Âm thanh, hình ảnh phù hợp cho lứa tuổi từ 5-10.
  + Giao diện đẹp, nhiều hiệu ứng thú vị để thu hút trẻ em.
  + Dễ sử dụng: các thao tác học tập, chọn bài học, chuyển bài học… Phải đơn giản để bé có thể dễ dàng tự sử dụng ngay lần đầu tiên sử dụng sau khi được hướng dẫn.
  + Ứng dụng có thể cài đặt và chạy ổn định trên hệ điều hành win 8,10

# : CƠ SỞ LÝ THUYẾT

## Công nghệ WPF

### Giới thiệu về WPF

WPF, viết tắt của Windows Presentation Foundation, là hệ thống API mới hỗ trợ việc xây dựng giao diện đồ họa trên nên Windows. Được xem như thế hệ kế tiếp của WinForms, WPF tăng cường khả năng lập trình giao diện của lập trình viên bằng cách cung cấp các API cho phép tận dụng những lợi thế về đa phương tiện hiện đại. Là một bộ phận của .NET Framework 3.5, WPF sẵn có trong Windows Vista và Windows Server 2008. Đồng thời, WPF cũng có thể hoạt động trên nền Windows XP Pack 2 hoặc mới hơn, và cả trên Windows Server 2003.

### Mục tiêu của WPF

WPF được xây dựng nhằm vào 3 mục tiêu cơ bản:

* Cung cấp một nền tảng thống nhất để xây dựng giao diện người dùng.
* Cho phép người lập trình và người thiết kế giao diện làm việc cùng nhau một cách dễ dàng.
* Cung cấp một công nghệ chung để xây dựng giao diện người dùng trên cả Windows và trình duyệt Web.

### Ngôn ngữ XAML

XAML, viết tắt của Extensible Application Markup Language, là ngôn ngữ đặc tả dựa trên XML được dùng để định nghĩa các đối tượng và thuộc tính của chúng, mối quan hệ cũng như sự tương tác. XAML đặc biệt được dùng trong các công nghệ của .NET Framework 3.0 (trong đó có WPF) như ngôn ngữ đặc tả giao diện người dùng (user interface - UI) nhằm mô tả cấu trúc và đặc tính của các phần tử UI, sự liên kết dữ liệu, các sự kiện và các đặc tính khác. XAML là mô hình có tính đột phá trong lĩnh vực tính toán trên Internet được chấp nhận rộng rãi trong nhiều hệ thống và bởi nhiều nhà cung cấp phần mềm.

XAML định ra một tập các phần tử XML như Button, TextBox, Label…, nhằm định nghĩa các đối tượng đồ họa tương ứng như nút bấm, hộp thoại, nhãn…, và nhờ đó cho phép mô tả chính xác diện mạo của giao diện người dùng. Các phần tử XAML cũng chứa các thuộc tính, cho phép thiết lập nhiều tính chất khác nhau của đối tượng đồ họa tương ứng. Ví dụ, đoạn mã sau sẽ tạo ra một nút bấm màu đỏ có nhan đề “No”.

*<Button Background="Red">No</Button>*

Mỗi phần tử XAML lại tương ứng với một lớp WPF, và mỗi thuộc tính của phần tử đó lại tương ứng với thuộc tính hay sự kiện của lớp này. Chẳng hạn, nút bấm màu đỏ trong ví dụ trên có thể tạo bằng C# code như sau:

*Button btn = new Button();*

*btn.Background = Brushes.Red;*

*btn.Content = "No";*

Nếu như mọi thứ có thể biểu diễn bằng XAML thì cũng có thể biểu diễn bằng đoạn mã, thì câu hỏi đặt ra là XAML có ý nghĩa gì? Câu trả lời là việc xây dựng các công cụ sinh và sử dụng các đặc tả bằng XML dễ dàng hơn nhiều so với xây dựng một công cụ tương tự làm việc với đoạn mã. Bởi vậy, XAML mở ra một cách thức tốt hơn để lập trình viên và người thiết kế làm việc với nhau.



Hình .Tương tác giữa nhà thiết kế và lập trình viên thông qua XAML

**XAML** đóng vai trò một ngôn ngữ chung giữa môi trường thiết kế giao diện và môi trường lập trình. Đối với người thiết kế, XAML không những cho phép người thiết kế dễ dàng mô tả, chỉnh sửa các đối tượng UI và các đặc tính của chúng, mà còn cho phép họ tương tác với các đối tượng này ở mức độ nhất định. Do vậy, tăng khả năng cảm quan của người thiết kế đối với giao diện. Ngoài ra, nó cũng giúp người thiết kế hạn chế những ý tưởng đồ họa không khả thi khi lập trình. Đối với người lập trình, nhờ một môi trường lập trình có khả năng tự động tái tạo giao diện đã thiết kế dựa trên file đặc tả bằng XAML do người thiết kế chuyển sang, người lập trình không cần tự mình tái tạo lại giao diện. Điều này giảm đi nhiều công sức và thời gian để phát triển giao diện, cũng như tránh những sai lệch giữa giao diện do người thiết kế và giao diện do người lập trình tái tạo. Vì cả môi trường thiết kế và lập trình đều có khả năng hiểu và sử dụng XAML, ứng dụng WPF có thể chuyển qua lại giữa hai môi trường phát triển để sửa đổi hay bổ sung giao diện một cách dễ dàng. Với tất cả những lợi điểm này, vị thế của người thiết kế trong việc xây dựng giao diện được nâng cao.

### WPF cho Window và trình duyệt Web

Trong thời đại bùng nổ của Internet, các ứng dụng Web ngày một phát triển. Việc trang bị giao diện người dùng với đầy đủ tính năng như một ứng dụng desktop sẽ thu hút nhiều người sử dụng;, và do đó góp phần làm tăng giá trị doanh nghiệp. Tuy nhiên, như đã nêu trong phần đầu, với những công nghệ truyền thống, để phát triển một giao diện đồ họa vừa hoạt động trên desktop vừa trên trình duyệt Web, đòi hỏi phải sử dụng những công nghệ hoàn toàn khác nhau, giống như việc xây dựng hai giao diện hoàn toàn độc lập. Điều này tạo ra chi phí không cần thiết để phát triển giao diện.

WPF là một giải pháp cho vấn đề này. Lập trình viên có thể tạo ra một ứng dụng trình duyệt *XAML* (XBAP) sử dụng WPF chạy trên Internet Explore. Trên thực tế, cùng đoạn code này có thể được dùng để sinh ứng dụng WPF chạy độc lập trên Windows.

Phần giao diện của ứng dụng dạng XBAP được trình duyệt chia thành các frame thay vì chạy trên các cửa sổ riêng, ngoài ra, các chức năng đều được bảo toàn. Cùng một đoạn mã được sử dụng chung cho cả hai trường hợp sẽ làm giảm khối lượng công việc cần thiết để phát triển hai dạng giao diện. Thêm vào đó, sử dụng cùng một đoạn mã cũng có nghĩa là sử dụng cùng kỹ năng của lập trình viên. Do đó, lập trình viên chỉ cần có học một kiến thức chung là có thể sử dụng trong cả hai trường hợp. Một lợi điểm nữa của việc dùng chung công nghệ cho cả giao diện Windows và giao diện Web là người xây dựng ứng dụng không nhất thiết phải quyết định trước loại giao diện nào được sử dụng. Miễn là máy client đáp ứng được những yêu cầu hệ thống để chạy XBAP, một ứng dụng có thể cung cấp cả giao diện Windows và giao diện Web, mà chỉ sử dụng phần lớn những đoạn mã giống nhau.

Mỗi ứng dụng XBAP được download khi cần từ một Web server, nên nó phải tuân theo những yêu cầu về an ninh khắt khe hơn đối với một ứng dụng Windows độc lập. Theo đó, XBAP chạy trong phạm vi sandbox an ninh do hệ thống an ninh truy nhập mã của. NET Framework cung cấp. XBAP chỉ chạy với các hệ thống Windows có cài đặt WPF và chỉ với Internet Explore phiên bản 6 và 7 trở lên.

### Các thành phần của WPF

Giống như các thành phần khác của .NET Framework, WPF tổ chức các chức năng theo một nhóm namespace cùng trực thuộc namespace System.Windows. Bất kể chức năng nào được sử dụng, cấu trúc cơ bản của mọi ứng dụng WPF đều gần như nhau. Là ứng dụng Windows độc lập hay là một XBAP, một ứng dụng WPF điển hình bao giờ cũng gồm một tập các trang XAML và phần code tương ứng được viết bằng C# hoặc Visual Basic, còn gọi là các file code-behind. Tất cả các ứng dụng đều kế thừa từ lớp chuẩn Application của WPF. Lớp này cung cấp những dịch vụ chung cho mọi ứng dụng, chẳng hạn như các biến lưu trữ trạng thái của ứng dụng, các phương thức chuẩn để kích hoạt hay kết thúc ứng dụng.

Mặc dù WPF cung cấp một nền tảng thống nhất để tạo giao diện người dùng, những công nghệ mà WPF chứa đựng có thể phân chia thành những thành phần độc lập. Nhân của WPF là cơ chế tạo sinh đồ họa dựa trên vector và độc lập với độ phân giải nhằm tận dụng những lợi thế của phần cứng đồ họa hiện đại. WPF được mở rộng với các tập tính năng phát triển ứng dụng bao gồm XAML, các control, cơ chế móc nối dữ liệu, layout, đồ họa 2 chiều, ba chiều, hoạt họa, style, khuôn dạng mẫu, văn bản, media, text và in ấn. WPF nằm trong .NET Framework, nên ngoài ra, ứng dụng WPF có thể kết hợp các thành phần khác có trong thư viện lớp của .NET Framework.



Hình 2.Các thành phần cơ bản của WPF.

#### Layout và Control

Để sắp đặt các thành phần khác nhau trên giao diện, ứng dụng WPF sử dụng panel. Mỗi panel có thể chứa các thành phần con, bao gồm các control như nút bấm hay hộp thoại, hay bản thân những panel khác. Những loại panel khác nhau cho phép sắp xếp thành phần con theo những cách khác nhau. Ví dụ, DockPanel cho phép các thành phần con có thể được đặt dọc theo cạnh của panel đó, trong khi Grid cho phép sắp đặt các thành phần con của nó trên một lưới tọa độ.

Giống như bất kỳ một công nghệ giao diện người dùng nào, WPF cung cấp một số lượng lớn các control. Ngoài ra, người dùng có thể tùy ý định nghĩa các control theo ý mình. Các control chuẩn gồm Button, Label, TextBox, ListBox, Menu, Slider, hay phức tạp hơn có SpellCheck, PasswordBox… Các sự kiện do người dùng tạo ra, như di chuyển chuột hay ấn phím, có thể được các control nắm bắt và xử lý. Trong khi các control và các thành phần giao diện khác có thể được đặc tả đầy đủ bằng XAML, các sự kiện bắt buộc phải được xử lý bằng mã trình.

#### Style và Template

Giống như sử dụng Cascading Style Sheets (CSS) đối với HTML, việc định ra thuộc tính đồ họa cho các đối tượng giao diện một lần, rồi sau đó áp dụng lại cho các đối tượng khác cùng loại thường rất tiện lợi. WPF cũng cung cấp tính năng tương tự bằng việc sử dụng thành phần Style của XAML. Ví dụ, kiểu ButtonStyle có thể được định nghĩa như sau:

*<Style x:Key="ButtonStyle">*

*<Setter Property="Control.Background" Value="Red"/>*

*<Setter Property="Control.FontSize" Value="16"/>*

*</Style>*

Bất kỳ nút bấm nào sử dụng kiểu này sẽ có nền màu đỏ và sử dụng font chữ kích thước 16.

Ví dụ:

*<Button Style="{StaticResource ButtonStyle}">*

*Click Here*

</Button>

Một Style có thể được dẫn xuất từ một Style khác, thừa kế hoặc chồng lên những thuộc tính đã thiết lập. Mỗi style có thể định nghĩa các trigger cho phép tạo ra những hiệu ứng tương tác đặc biệt, chẳng hạn như khi lướt chuột qua nút bấm, nút bấm chuyển thành màu vàng. WPF cũng hỗ trợ sử dụng template. Mỗi template tương tự như một style, và ở hai dạng:

- Template cho dữ liệu: sử dụng thành phần DataTemplate của XAML để thiết lập một nhóm thuộc tính hiển thị của dữ liệu như màu sắc, phương thức căn lề...

- Template cho control: sử dụng thành phần ControlTemplate của XAML để định ra diện mạo của một control.

#### Text

Giao diện người dùng ít nhiều đều hiển thị chữ hay text. Đối với phần lớn mọi người, đọc text trên màn hình thường khó hơn đọc trên giấy in. Đó là do chất lượng hiển thị text trên màn hình kém hơn so với khi in ra giấy. WPF tập trung giải quyết vấn đề này, làm chất lượng text hiển thị trên màn hình tương đương trên giấy in. Cụ thể, WPF hỗ trợ các font chữ OpenType chuẩn, cho phép sử dụng các thư viện font đã có. WPF cũng hỗ trợ công nghệ font chữ mới ClearType, cho phép hiển thị các ký tự mịn hơn đối với mắt người, đặc biệt là trên màn hình tinh thể lỏng (LCD).

Để nâng cao hơn nữa chất lượng hiển thị text, WPF cho phép một số công nghệ khác như chữ ghép, theo đó một nhóm ký tự được thay thế bằng một ảnh đơn nhất, tạo tâm lý thoải mái hơn khi đọc đối với người dùng.

#### Văn bản

WPF hỗ trợ ba dạng văn bản: văn bản cố định (fixed), văn bản thích nghi (flow/adaptive) và văn bản XPS (XML Paper Specification). Kèm theo đó, WPF cũng cung cấp các dịch vụ để tạo, xem, quản lý, ghi chú, đóng gói và in ấn văn bản.

Văn bản cố định trông không đổi bất kể chúng được hiển thị trên màn hình hay in ra máy in. Trong WPF, những văn bản dạng này được định nghĩa bằng phần tử FixedDocument trong XAML và được hiển thị bằng control DocumentViewer.

Trong khi đó, văn bản thích nghi thường chỉ dùng để đọc trên màn hình, và có khả năng tự động thay đổi các thuộc tính hiển thị ảnh và text cho phù hợp với kích thước cửa số hay các yếu tố môi trường khác nhằm nâng cao chất lượng đọc cho người dùng. Văn bản thích nghi được định nghĩa bằng phần tử FlowDocument. Để hiển thị văn bản thích nghi, WPF sử dụng một số control khác nhau, chẳng hạn như FlowDocumentPageViewer, FlowDocumentScrollViewer, FlowDocumentReader…

Văn bản XPS xây dựng trên cơ sở văn bản bất động của WPF. XPS là một định dạng mở theo đặc tả XML, có khả năng sử dụng trên nhiều nền tảng khác nhau, được thiết kế nhằm tạo thuận lợi cho việc xây dựng, chia sẻ, in ấn và lưu trữ văn bản. Cũng như văn bản cố định, văn bản XPS được hiển thị bằng DocumentViewer.

#### Hình ảnh

Trong WPF, hình ảnh được hiển thị nhờ control **Image**, ví dụ:

<ImageWidth="200 Source="C:\DocumentsandSettings\AllUsers\Documents\MyPictures\Ava.jpg" />

Control Image có thể hiển thị hình ảnh lưu trữ dưới nhiều khuôn dạng khác nhau, bao gồm JPEG, BMP, TIFF, GIF và PNG. Nó cũng có thể hiển thị hình ảnh dạng Windows Media Photo mới được sử dụng trong Windows Vista. Bất kể ở khuôn dạng nào, WPF sử dụng Windows Imaging Component (WIC) để tạo ra hình ảnh. Cùng với các codec dùng cho các khuôn dạng ảnh kể trên, WIC cũng cung cấp một nền tảng chung để bổ sung codec khác.

#### Video và âm thanh

Khi tốc độ của các bộ xử lý và truyền thông mạng ngày một nâng cao, video trở thành một phần tương tác lớn của người dùng với phần mềm. Người dùng cũng sử dụng nhiều thời gian để nghe nhạc và các dạng âm thanh khác trên máy tính. Do đó, WPF cung cấp tính năng hỗ trợ cả hai dạng media này thông qua phần tử MediaElement. Control này có thể chơi các định dạng video WMV, MPEG và AVI, và nhiều định dạng âm thanh khác nhau. Việc lập trình để chạy một đoạn video trở nên khá đơn giản, như trong ví dụ sau:

<MediaElement Source="C:\DocumentsandSettings\AllUsers\Documents\My Videos\Ruby.wmv" />

#### Đồ họa hai chiều

Trong 20 năm gần đây, việc tạo ra đồ họa hai chiều trên Windows dựa trên Graphics Device Interface (GDI) và phiên bản sau của nó GDI+. Các ứng dụng Windows Forms phải sử dụng chức năng này thông qua một namespace khác hoàn toàn, bởi bản thân Windows Forms không tích hợp đồ họa 2 chiều. Đối với đồ họa 3 chiều thì càng tồi hơn, Windows Forms phải dựa trên công nghệ hoàn toàn biệt lập là Direct3D. Với WPF, vấn đề trở nên đơn giản hơn nhiều. Cả đồ họa 2 chiều và 3 chiều đều có thể được tạo ra trực tiếp trong XAML hoặc trong code sử dụng thư viện WPF tương ứng.

Đối với đồ họa 2 chiều, WPF định ra nhóm control của các khuôn hình (shapes) mà ứng dụng có thể sử dụng để tạo nên hình ảnh, gồm:

**\* Line**: vẽ đường thẳng qua 2 điểm.

**\* Elllipse**: vẽ ellipse.

**\* Rectangle**: vẽ chữ nhật.

**\* Polygon**: vẽ đa giác.

**\* Polyline**: vẽ đa giác mở.

**\* Path**: vẽ hình theo một đường bất kỳ.

Mỗi khuôn hình đều có các thuộc tính phong phú cho phép hiển thị với nhiều tính chất khác nhau: màu nền, màu biên… Một đặc điểm quan trọng trong WPF là: vì mọi thứ đều được xây dựng trên một nền chung, việc kết hợp các đặc tính và đối tượng khác nhau, chẳng hạn, lồng một ảnh vào một hình chữ nhật, trở nên đơn giản. Điểm thú vị nữa là các đối tượng hình học này còn có thể thu nhận các sự kiện từ phía người dùng như một control, chẳng hạn sự kiện nhắp chuột.

Ngoài ra, WPF cũng cung cấp một nhóm chức năng hình học khác, gọi là geometries, để làm việc với đồ họa hai chiều, như LineGeometry, RectangleGeometry, EllipseGeometry, và PathGeometry. Dạng hình học này có nhiều thuộc tính và chức năng tương tự như các khuôn hình đã nêu trên. Điểm khác biệt quan trọng nhất là các geometries không được dùng để hiển thị, chúng được dùng chủ yếu để tính toán hình học, ví dụ như để định ra các vùng miền, theo dõi vị trí bấm chuột...

Thêm vào đó, WPF cung cấp lớp Transform cho phép thực hiện các biến đổi hình học như xoay, dịch chuyển, co giãn đối tượng đồ họa; hoặc cho phép thực hiện các hiệu ứng hoạt họa theo thời gian thông qua các lớp Animation và Timing.

#### Đồ họa ba chiều

WPF hỗ trợ đồ họa 3 chiều bằng việc gói các lời gọi API của Direct3D, và do vậy, việc sử dụng chúng trở nên thống nhất và đơn giản hơn đáng kể. Để hiển thị đồ họa ba chiều, ứng dụng WPF sử dụng control Viewport3D. Để tạo ra các cảnh ba chiều, lập trình viên mô tả một hay nhiều mô hình, sau đó, phân định cách thức các mô hình này được chiếu sáng hay hiển thị. Như thường lệ, điều này được thực hiện bằng XAML, bằng code hay trộn cả hai. Để mô tả mô hình, WPF cung cấp lớp GeometryModel3D để tạo ra hình dạng của mô hình. Khi mô hình đã được định hình, diện mạo bên ngoài của nó có thể được điều khiển bằng việc phủ lên các vật liệu (material). Chẳng hạn, lớp SpecularMaterial cho phép tạo bóng trên bề mặt mô hình.

Bất kể được làm từ vật liệu gì, một mô hình có thể được chiếu sáng theo nhiều cách. Lớp DirectionalLight cho phép ánh sáng tới từ một hướng xác định, trong khi lớp AmbientLight tạo ra ánh sáng đồng đều trên mọi vật trong cảnh. Cuối cùng, để định ra cách nhìn cảnh, lập trình viên phải định ra một *camera*. Ví dụ, PerspectiveCamera cho phép phân định khoảng cách từ vị trí nhìn tới vật thể và kiểu nhìn phối cảnh (tuân theo luật gần xa).

Xây dựng cảnh ba chiều trực tiếp bằng XAML hay mã trình đều không đơn giản. Do đó, chỉ nên dùng ứng dụng WPF để hiển thị cảnh ba chiều, việc xây dựng cảnh nên được thực hiện bằng những công cụ đồ họa chuyên biệt.

#### Móc nối dữ liệu

Phần lớn các ứng dụng được tạo ra đều cung cấp cho người dùng phương tiện để xem và sửa đổi dữ liệu. Trong các ứng dụng WPF, việc lưu trữ và truy xuất dữ liệu đã được thực hiện bởi các công nghệ như Microsoft SQL Server và ADO.NET. Sau khi dữ liệu được truy xuất và tải vào các đối tượng quản lý dữ liệu trên ứng dụng, phần việc khó khăn của ứng dụng WPF mới bắt đầu. Về cơ bản, có hai công việc phải thực hiện:

1) Sao chép dữ liệu từ các đối tượng quản lý dữ liệu vào các control trên giao diện, qua đó, dữ liệu có thể được hiển thị hay sửa đổi.

2) Đảm bảo rằng những thay đổi trên dữ liệu từ các control được cập nhật trở lại các đối tượng quản lý dữ liệu.

Để đơn giản hóa quá trình phát triển ứng dụng, WPF cung cấp một cơ chế móc nối dữ liệu để thực hiện tự động những bước này. Phần nhân của cơ chế móc nối dữ liệu là lớp Binding mà nhiệm vụ của nó là liên kết control trên giao diện (đích) với đối tượng quản lý dữ liệu (nguồn). Mối quan hệ này được minh họa trong hình 4 dưới đây:



Hình .Quan hệ giữa đối tượng dữ liệu và đối tượng phụ thuộc.

Việc hỗ trợ móc nối dữ liệu được xây dựng ngay từ nhân của WPF. Tất cả các đối tượng đồ họa trong WPF đều kế thừa từ DependencyObject, chúng là các đối tượng phụ thuộc. Chức năng mà lớp cơ sở này hỗ trợ cho phép thực hiện hiệu ứng hoạt họa, tạo kiểu mẫu (styling) và móc nối dữ liệu. Các đối tượng này đều mang một thuộc tính đặc biệt gọi là DependencyProperty, thuộc tính phụ thuộc. Phần lớn các thuộc tính hay dùng như Text, Content, Width, Height, vân vân đều là các thuộc tính phụ thuộc. Tất cả các thuộc tính phụ thuộc đều có thể tạo hiệu ứng hoạt họa, tạo kiểu và kết nối dữ liệu.

Cơ chế móc nối dữ liệu trong WPF còn cung cấp thêm những tính năng như xác thực tính hợp lệ, sắp xếp, lọc và phân nhóm dữ liệu. Thêm vào đó, tính năng móc nối dữ liệu cũng hỗ trợ sử dụng khuôn mẫu dữ liệu (data template) để tạo ra các đối tượng giao diện tùy biến có kết nối dữ liệu, khi các control chuẩn không phù hợp. Móc nối dữ liệu và khuôn dạng dữ liệu có thể được coi là tính năng mạnh nhất của WPF.

## Cơ sở dữ liệu SQL Server

### Giới thiệu về SQL Server

SQL là viết tắt của Structure Query Language, nó là một công cụ quản lý dữ liệu được sử dụng phổ biến ở nhiều lĩnh vực. là một hệ thống quản lý cơ sở dữ liệu (Relational Database Management System (RDBMS) ) sử dụng Transact-SQL để trao đổi dữ liệu giữa Client computer và SQL Server computer. Một RDBMS bao gồm databases, database engine và các ứng dụng dùng để quản lý dữ liệu và các bộ phận khác nhau trong RDBMS.Hầu hết các ngôn ngữ bậc cao đều có trình hỗ trợ SQL như Visual BASic,Oracle,Visual C...

SQL Server được tối ưu để có thể chạy trên môi trường cơ sở dữ liệu rất lớn (Very Large Database Environment) lên đến Tera-Byte và có thể phục vụ cùng lúc cho hàng ngàn user. SQL Server có thể kết hợp "ăn ý" với các server khác như Microsoft Internet Information Server (IIS), E-Commerce Server, Proxy Server....

### Lịch sử phát triển của SQL Server

SQL được phát triển từ ngôn ngữ SEQUEL2 bởi IBM theo mô hình Codd tại trung tâm nghiên cứu của IBM ở California ,vào những năm 70 cho hệ thống QTCSDL lớn.

Ðầu tiên SQL được sử dụng trong các ngôn ngữ quản lý CSDL và chạy trên các máy đơn lẻ. Song do sự phát triển nhanh chóng của nhu cầu xây dựng những CSDL lớn theo mô hình khách chủ( trong mô hình này toàn bộ CSDL được tập trung trên máy chủ (Server)). Mọi thao tác xử lý dữ liệu được thực hiện trên máy chủ bằng các lệnh SQL máy trạm chỉ dùng để cập nhập hoặc lấy thông tin từ máy chủ). Ngày nay trong các ngôn ngữ lập trình bậc cao đều có sự trợ giúp của SQL. Nhất là trong lĩnh vực phát triển của Internet ngôn ngữ SQL càng đóng vai trò quan trọng hơn. Nó được sử dụng để nhanh chóng tạo các trang Web động..

SQL đã được viện tiêu chuẩn quốc gia Mỹ (ANSI)và tổ chức tiêu chuẩn quốc tế (ISO) chấp nhận như một ngôn ngữ chuẩn cho CSDL quan hệ .Nhưng cho đến nay chuẩn này chưa đưa ra đủ 100%.Nên các SQL nhúng trong các ngôn ngữ lập trình khác nhau đã được bổ xung mở rộng cho SQL chuẩn cho phù hợp với các ứng dụng của mình.Do vậy có sự khác nhau rõ ràng giưã các SQL.

# : PHÂN TÍCH

## Phân tích yêu cầu bằng UML

### Usecase Tổng Quát



### Danh sách các actor

|  |  |  |
| --- | --- | --- |
| **Tác nhân** | **Mô tả tác nhân** | **Ghi chú** |
| Người dùng | Người sử dụng ứng dụng để học tiếng anh |  |

Bảng : Danh sách các Actor

### Danh sách các Usecase

|  |  |  |  |
| --- | --- | --- | --- |
| **ID** | **Tên Use case** | **Mô tả ngắn gọn Use case** | **Chức năng** |
| UC1 | HocTuVungQuaVideo | Usecase này thực hiện việc phát video bài học của một chủ đề từ vựng. | Phát video bài giảng của một chủ đề từ vựng. |
| UC2 | HocTuVungQuaTuDien | Usecase thực hiện việc hiển thị lần lượt thông tin từng từ vựng của một chủ đề từ vựng. | Hiển thị thông tin từng từ vựng bao gồm từ tiếng anh, phiên âm, dịch nghĩa, hình ảnh, âm thanh đọc. |
| UC3 | ThuThapBauVat | Usecase hiển thị lần lượt từng câu hỏi trắc nghiệm về từ vựng của một chủ đề từ vựng. | Những câu hỏi từ vựng được hỏi theo thứ tự ngẫu nhiên, trả lời đúng tất cả các câu hỏi để thu thập được báu vật của chủ đề đó, nếu trả lời sai sẽ chuyển sang câu tiếp theo và câu hỏi sai sẽ tiếp tục được hỏi sau đó cho đến khi đúng tất cả các câu hỏi. |
| UC4 | BaoVeBauVat | Usecase hiển thị lần lượt từng câu hỏi trắc nghiệm về từ vựng của một chủ đề từ vựng đã hoàn thành trước đó. ( mỗi khi qua ngày mới, lần đầu tiên khởi động ứng dụng thì usecase này sẽ được gọi 1 lần) | Những câu hỏi từ vựng được hỏi theo thứ tự ngẫu nhiên, trả lời tất cả các câu hỏi lần để bảo vệ báu vật, nếu trả lời sai sẽ chuyển sang câu tiếp theo và câu hỏi sai sẽ tiếp tục được hỏi sau đó cho đến khi trả lời đúng tất cả các câu hỏi ( câu hỏi đã đúng 3 lần sẽ không được hỏi lại). Nếu có 1 câu hỏi bất kì sai quá 3 lần thì thử thách của bạn thất bại và báu vật sẽ bị mất. |
| UC5 | LuyenTap | Usecase này thực hiện việc hiển thị lần lượt từng câu hỏi trắc nghiệm trong danh sách các từ vựng thuộc các chủ đề đã hoàn thành. | Những câu hỏi trắc nghiệm các từ vựng đã học sẽ được hỏi liên tục cho người sử dụng trả lời cho đến khi trả lời sai sẽ dừng lại và ghi nhận số câu đúng. |
| UC6 | XemBoSuuTap | Usecase này hiển thị danh sách các báu vật đã thu thập được. ( tương đương với các chủ đề đã hoàn thành của người dùng). | Hiển thị danh sách các báu vật đã thu thập. |
| UC7 | HuongDanSuDung | Usecase này hướng dẫn quy tắc trò chơi và cách sử dụng ứng dụng | Hướng dẫn cách sử dụng và các quy tắc trò chơi. |

Bảng : Danh Sách Các Usecase chức năng

## Đặc tả các yêu cầu chức năng

### UseCase UC1: HocTuVungQuaVideo

#### Mô tả:

|  |  |  |
| --- | --- | --- |
| UseCase: HocTuVungQuaVideo | | |
| Mục đích | Xem video bài giảng của một chủ đề từ vựng | |
| Mô tả | Người dùng chọn chủ đề để xem video của chủ đề đó | |
| Tác Nhân | Người dùng | |
| Điều kiện trước | Chủ đề đã chọn phải có video trong csdl | |
| Điều kiện sau | Video được phát lên giao diện của ứng dụng | |
| Luồng sự kiện chính | Người dùng | Hệ thống |
| 1. Người dùng chọn chủ đề từ vựng.  3. Người dùng chọn nút bắt đầu. | 2. Hiển thị form thông tin chủ đề, báu vật của chủ đề, nút bắt đầu, nút hủy.  4. Phát video của chủ đề đó lên giao diện. |
| Luồng sự kiện phụ | 3.1 Người dùng chọn nút hủy để quay lại giao diện chọn chủ đề. | |

Bảng : Đặc tả UseCase HocTuVungQuaVideo

#### Biểu đồ

### UseCase UC2: HocTuVungQuaTuDien

#### Mô tả:

|  |  |  |
| --- | --- | --- |
| UseCase: HocTuVungQuaTuDien | | |
| Mục đích | Xem từ điển các từ vựng của chủ đề đã chọn | |
| Mô tả | Người dùng chọn chủ đề để xem video của chủ đề đó, sau khi xem xong video hoặc chọn nút bỏ qua khi video đang phát thì thực hiện usecase này. | |
| Tác Nhân | Người dùng | |
| Điều kiện trước | Đã chọn chủ đề để sử dụng usecase UC1  Đã xem xong video hoặc chọn nút bỏ qua khi đang xem video | |
| Điều kiện sau | Thông tin từng từ vựng của chủ đề đã chọn được hiển thị lên giao diện kèm âm thanh và hình ảnh của từ vựng đó | |
| Luồng sự kiện chính | Người dùng | Hệ thống |
| 2. Chọn nút Next để xem từ tiếp theo  4. Chọn nút bỏ qua để kết thúc usecase. | 1. Hiển thị giao diện xem từ điển của chủ đề đã chọn với thông tin của từ đầu tiên ( có âm thanh và hình ảnh đi kèm).  3. Chuyển sang từ tiếp theo |
| Luồng sự kiện phụ | 2.1 Chọn nút Previous để xem từ trước đó. | 3.1 Chuyển sang từ trước đó |
| Ghi chú | 1. Nếu đang là từ cuối cùng thì nút next bị ẩn đi.  2. Nếu đang là từ đầu tiên thì nút previous bị ẩn đi. | |

Bảng : Đặc tả UseCase HocTuVungQuaTuDien

#### Biểu đồ

### UseCase UC3: ThuThapBauVat

#### Mô tả:

|  |  |  |
| --- | --- | --- |
| UseCase: ThuThapBauVat | | |
| Mục đích | Hiển thị từng câu hỏi từ vựng thuộc 1 chủ đề cho người dùng trả lời, hoàn thành UC này để thu thập báu vật của chủ đề đó | |
| Mô tả | Hệ thống sẽ hiển thị từng câu hỏi (bao gồm hình ảnh và âm thanh) cho người dùng chọn câu trả lời | |
| Tác Nhân | Người dùng | |
| Điều kiện trước | Chủ đề đã chọn phải có dữ liệu trong csdl ( danh sách từ vựng của chủ đề đó).  Đã hoàn thành UC1 và UC2. | |
| Điều kiện sau | Hiển thị giao diện thu thập báu vật  Sau khi trả lời đúng tất cả câu hỏi thì thông báo chúc mừng (kèm âm nhạc và hình ảnh của báu vật của chủ đề đó) | |
| Luồng sự kiện chính | Người dùng | Hệ thống |
| 2. Chọn 1 trong 4 đáp án để trả lời câu hỏi | 1. Hiển thị giao diện thu thập báu vật với câu hỏi đầu tiên trong danh sách.  3. Kiểm tra kết quả  3.1 Nếu đáp án đúng thì phát chuông thông báo chính xác và chuyển sang câu tiếp theo.  3.2 Nếu đáp án sai thì phát chuông thông báo không chính xác, chuyển câu hỏi về cuối danh sách câu hỏi và chuyển sang câu tiếp theo.  3.3 Nếu đáp án đúng và tất cả câu hỏi trong danh sách câu hỏi đã đúng hết thì hiển thị thông báo chúc mừng hoàn thành thử thách (kèm âm nhạc và hình ảnh của báu vật của chủ đề đó) kết thúc usecase. |
| Luồng sự kiện phụ |  | |
| Ghi chú | 1. giao diện thu thập báu vật là giao diện trả lời câu hỏi trắc nghiệm gồm 1 câu hỏi hình ảnh và 4 câu trả lời tiếng anh, chỉ có 1 từ trong 4 từ là nghĩa đúng của hình ảnh.  2. các câu hỏi trong danh sách câu hỏi tương đương với các từ vựng trong chủ đề đó. | |

Bảng : Đặc tả UseCase ThuThapBauVat

#### Biểu đồ

### UseCase UC5: BaoVeBauVat

#### Mô tả:

|  |  |  |
| --- | --- | --- |
| UseCase: BaoVeBauVat | | |
| Mục đích | Cho người dùng ôn tập từ vựng đã học trước đó.  Hiển thị từng câu hỏi từ vựng thuộc 1 chủ đề cho người dùng trả lời | |
| Mô tả | Hệ thống sẽ hiển thị từng câu hỏi (bao gồm hình ảnh và âm thanh) cho người dùng chọn câu trả lời | |
| Tác Nhân | Người dùng | |
| Điều kiện trước | Người dùng phải hoàn thành ít nhất 1 chủ đề | |
| Điều kiện sau | Hiển thị giao diện bảo vệ báu vật  Sau khi trả lời đúng tất cả câu hỏi thì thông báo chúc mừng (kèm âm nhạc và hình ảnh của báu vật của chủ đề đó)  Nếu thất bại thì hiển thị thông báo báu vật đã bị đánh cắp. | |
| Luồng sự kiện chính | Người dùng | Hệ thống |
| 2. Chọn 1 trong 4 đáp án để trả lời câu hỏi | 1. Hiển thị giao diện bảo vệ báu vật với câu hỏi đầu tiên trong danh sách.  3. Kiểm tra kết quả  3.1 Nếu đáp án đúng thì phát chuông thông báo chính xác và chuyển sang câu tiếp theo.  3.2 Nếu đáp án sai thì phát chuông thông báo không chính xác, tăng số lần trả lời sai của câu hỏi này lên 1, chuyển câu hỏi về cuối danh sách câu hỏi và chuyển sang câu tiếp theo.  3.3 Nếu đáp án đúng và tất cả câu hỏi trong danh sách câu hỏi đã đúng hết thì hiển thị thông báo chúc mừng hoàn thành thử thách (kèm âm nhạc và hình ảnh của báu vật của chủ đề đó) kết thúc usecase. |
| Luồng sự kiện phụ |  | 3.2.1 Nếu câu hỏi đó có số lần trả lời sai là 3 thì thông báo báu vật đã bị đánh cắp và kết thúc usecase. |
| Ghi chú | 1. giao diện bảo vệ báu vật là giao diện trả lời câu hỏi trắc nghiệm gồm 1 câu hỏi hình ảnh và 4 câu trả lời tiếng anh, chỉ có 1 từ trong 4 từ là nghĩa đúng của hình ảnh.  2. các câu hỏi trong danh sách câu hỏi tương đương với các từ vựng trong chủ đề đó.  3. Usecase này bắt buộc thực hiện hằng ngày, được gọi vào lần đầu tiên khởi chạy ứng dụng trong ngày với điều kiện người dùng đã hoàn thành ít nhất 1 chủ đề. | |

Bảng : Đặc tả UseCase BaoVeBauVat

#### Biểu đồ

### UseCase UC6: LuyenTap

|  |  |  |
| --- | --- | --- |
| UseCase: LuyenTap | | |
| Mục đích | Cho người dùng ôn tập từ vựng đã học trước đó.  Hiển thị từng câu hỏi từ vựng thuộc tất cả các chủ đề người dùng dã hoàn thành trước đó | |
| Mô tả | Hệ thống sẽ hiển thị từng câu hỏi (bao gồm hình ảnh và âm thanh) cho người dùng chọn câu trả lời | |
| Tác Nhân | Người dùng | |
| Điều kiện trước | Người dùng phải hoàn thành ít nhất 1 chủ đề | |
| Điều kiện sau | Hiển thị giao diện bảo vệ báu vật  Sau khi trả lời đúng tất cả câu hỏi thì thông báo chúc mừng (kèm âm nhạc và hình ảnh của báu vật của chủ đề đó)  Nếu trả lời sai thì hiển thị số câu đúng.  Số câu đúng cao nhất được ghi nhận vào csdl. | |
| Luồng sự kiện chính | Người dùng | Hệ thống |
| 2. Chọn 1 trong 4 đáp án để trả lời câu hỏi | 1. Hiển thị giao luyện tập với câu hỏi đầu tiên trong danh sách.  3. Kiểm tra kết quả  3.1 Nếu đáp án đúng thì phát chuông thông báo chính xác và chuyển sang câu tiếp theo.  3.2 Nếu đáp án sai thì phát chuông thông báo không chính xác, ghi nhận số câu đúng và thông báo số câu đúng.  3.3 Nếu đáp án đúng và tất cả câu hỏi trong danh sách câu hỏi đã đúng hết thì hiển thị thông báo chúc mừng , ghi nhận số câu đúng và kết thúc usecase. |
| Luồng sự kiện phụ |  |  |
| Ghi chú | 1. giao diện luyện tập là giao diện trả lời câu hỏi trắc nghiệm gồm 1 câu hỏi hình ảnh và 4 câu trả lời tiếng anh, chỉ có 1 từ trong 4 từ là nghĩa đúng của hình ảnh.  2. các câu hỏi trong danh sách câu hỏi tương đương với các từ vựng trong tất cả các chủ đề đã hoàn thành trước đó. | |

Bảng : Đặc tả UseCase LuyenTap

#### Biểu đồ

### UseCase UC6: XemBoSuuTap

#### Mô tả:

|  |  |  |
| --- | --- | --- |
| UseCase: XemBoSuuTap | | |
| Mục đích | Xem danh sách các báu vật thuộc các chủ đề đã hoàn thành | |
| Mô tả | Hiển thị danh sách các báu vật đã thu thập được | |
| Tác Nhân | Người dùng | |
| Điều kiện trước |  | |
| Điều kiện sau | Hiển thị danh sách báu vật | |
| Luồng sự kiện chính | Người dùng | Hệ thống |
| 1. Chọn chức năng xem bộ sưu tập | 2. hiển thị danh sách bộ sưu tập. |
| Luồng sự kiện phụ |  | |

Bảng : Đặc tả UseCasw XemBoSuuTap

#### Biểu đồ

### UseCase UC7: HuongDanSuDung

#### Mô tả

|  |  |  |
| --- | --- | --- |
| UseCase: HuongDanSuDung | | |
| Mục đích | Hướng dẫn người dùng sử dụng ứng dụng | |
| Mô tả | Hiển thị hướng dẫn sử dụng của ứng dụng | |
| Tác Nhân | Người dùng | |
| Điều kiện trước |  | |
| Điều kiện sau | Sau khi xem xong hướng dẫn sử dụng thì quay lại giao diện trang chủ | |
| Luồng sự kiện chính | Người dùng | Hệ thống |
| 1. Chọn chức năng xem hướng dẫn | 2. hiển thị hướng dẫn sử dụng. |
| Luồng sự kiện phụ |  | |

Bảng : Đặc tả UseCase HuongDanSuDung

#### Biểu đồ

# : THIẾT KẾ VÀ HIỆN THỰC

Chương này sinh viên trình bày quá trình thiết kế, thử nghiệm và hiện thực hệ thống hoặc bài toán.

## Tiêu đề mục

### Tiêu đề tiểu mục

Nội dung tiểu mục

### Tiêu đề tiểu mục

Nội dung tiểu mục

## Tiêu đề mục

### Tiêu đề tiểu mục

Nội dung tiểu mục

### Tiêu đề tiểu mục

Nội dung tiểu mục

# : KẾT LUẬN

## Kết quả đạt được

Trình bày các chức năng đã hoàn thành trong đồ án hoặc kết quả nghiên cứu.

## Hạn chế của đồ án

Trình bày những hạn chế, công việc chưa hoàn tất trong đồ án

## Hướng phát triển

Trình bày các định hướng phát triển cho hệ thống hoặc hướng nghiên cứu trong tương lai.

# 

# TÀI LIỆU THAM KHẢO

Các tài liệu Tiếng Việt

1. Họ và Tên tác giả (Thứ tự theo Tên). Tên sách. Tên nhà xuất bản, năm xuất bản.

Các tài liệu Tiếng Anh

1. Amy Apon. Lecture for Cluster and Grid Computing. University of Arkansas, 2004.

Các tài liệu từ Internet

1. Website: [www.cengageasia.com](http://www.cengageasia.com)
2. ...

# PHỤ LỤC