BỘ GIÁO DỤC VÀ ĐÀO TẠO

TRƯỜNG ĐẠI HỌC BÁCH KHOA HÀ NỘI

---------------------------------------

PHÙNG NGỌC VỮNG

Nghiên cứu và xây dựng công cụ hỗ trợ học trực tuyến cho học sinh và sinh viên.

LUẬN VĂN THẠC SĨ KỸ THUẬT

KỸ THUẬT PHẦN MỀM

Hà Nội - 2017

BỘ GIÁO DỤC VÀ ĐÀO TẠO

TRƯỜNG ĐẠI HỌC BÁCH KHOA HÀ NỘI

---------------------------------------

PHÙNG NGỌC VỮNG

Nghiên cứu và xây dựng công cụ hỗ trợ học trực tuyến cho học sinh và sinh viên.

Chuyên ngành: Kỹ thuật phần mềm

LUẬN VĂN THẠC SĨ KỸ THUẬT

KỸ THUẬT PHẦN MỀM

NGƯỜI HƯỚNG DẪN KHOA HỌC:

TS. NGUYỄN THANH HÙNG

Hà Nội - 2017

# LỜI CAM ĐOAN

Tôi xin cam đoan những gì tôi viết dưới đây là hoàn toàn chính thống không sao chép, những kết quả đo đạc mô phỏng có trong luận văn chưa từng được công bố từ bất cứ tài liệu nào dưới mọi hình thức. Các thông tin sử dụng trong luận văn có nguồn gốc và được trích dẫn rõ ràng. Tôi xin hoàn toàn chịu trách nhiệm nếu có dấu hiệu sao chép kết quả từ các tài liệu khác.

***Hà Nội, ngày 19 tháng 9 năm 2017***

**TÁC GIẢ**

**PHÙNG NGỌC VỮNG**

# LỜI MỞ ĐẦU

|  |
| --- |
| Ngày nay, cùng với sự phát triển của công nghệ thông tin, việc học tập, dạy học từ xa, học trực tuyến đã không còn xa lạ. Trong tất cả các lĩnh vực giáo dục, việc công nghệ thông tin hóa đều trở nên hiện hữu, các bài giảng học trực tuyến được biên soạn, truyền tải ngày càng đa dạng, mang đến nhiều trải nghiệm hữu ích đến từng học viên. Làm thế nào để học trực tuyến hiệu quả, hình thức bài giảng phong phú, thu hút được nhiều đối tượng đang là bài toán chưa có đáp án tối ưu đối với người làm giáo dục nói chung và người thầy nói riêng. Trong khi đó, đối tượng là học viên luôn chú trọng đến bài giảng tốt, tính tương tác cao, chi phí giảng thấp. Vậy nên , một bài giảng trực tuyến tốt phải là kết quả của sự kết hợp giữa Công nghệ thông tin và người thầy, có thể đáp ứng cho nhiều nhóm đối khác nhau như học sinh, phụ huynh , người tham khảo …đồng thời học phí, chi phí lưu trữ phải thấp, tốc độ truyền tải phải cao. Đây cũng là định hướng của những doanh nghiệp khi muốn xây dựng hệ thống, ứng dụng hỗ trợ cho việc học trực tuyến.  Trong khuôn khổ đề tài luận văn của mình, tôi xin giới thiệu quá trình xây dựng và phát triển ứng dụng đa nền tảng hỗ trợ việc học trực tuyến cho học sinh , sinh viên với kiểu dữ liệu thống nhất quy chuẩn tối ưu hơn cho việc lưu trữ và truyền tải. Ứng dụng có những chức năng cơ bản như : Hỗ trợ đọc các bài giảng định dạng cơ bản như PDF, chữ viết tay; Sử dụng dữ liệu quy chuẩn để trình diễn bài giảng; hỗ trợ bài giảng thời gian thực; Có tính tương tác giữa bài giảng với học viên trong quá trình học; Quản lý lớp học, bài giảng, thông tin học viên…  Bố cục của luận văn bao gồm bốn chương như sau   * Chương 1 : Đặt vấn đề * Chương 2 : Giải pháp khắc phục vấn đề và định hướng công nghệ * Chương 3 : Xây dựng ứng dụng E-LearningClient * Chương 4 : Kết luận và hướng phát triển |
|  |

# LỜI CẢM ƠN

*Trước tiên, tôi xin dành những lời cảm ơn chân thành tới tất cả các thầy cô Viện Công Nghệ Thông Tin và Truyền Thông, bộ môn Công Nghệ Phần Mềm đã hết sức giảng dạy tận tình, đem lại nguồn kiến thức giúp tôi có thể thực hiện đề tài.*

*Tôi xin gửi lời cảm ơn sâu sắc đến TS Nguyễn Thanh Hùng –Trưởng bộ môn Công Nghệ Phần Mềm, người đã theo sát và trực tiếp hướng dẫn tôi hoàn thành đề tài. Trong suốt quá trình nghiên cứu và tìm hiểu, thầy luôn tận tình chỉ bảo, sát sao trong công việc, định hướng cách giải quyết vấn đề.*

*Dù đã cố gắng hế sực, nhưng bài luận văn chắc chắc còn nhiều sai sót, tôi kính mong thầy cô thông cảm, chỉ bảo, tạo điều kiện cho tôi hoàn thiện thêm.*

*Tôi xin chân thành cảm ơn !*

*Hà Nội, ngày 19 tháng 9 năm 2017*

Học viên

Phùng Ngọc Vững

# DANH MỤC BẢNG BIỂU

[Bảng 1 Các đối tượng View trong Xamarin Form 30](#_Toc493967732)

[Bảng 2 Các đối tượng Cell trong Xamarin Form 31](#_Toc493967733)

[Bảng 3 Chi tiết dữ liệu mô tả video 34](#_Toc493967734)

[Bảng 4 Nuget cài đặt SkiaSharp 35](#_Toc493967735)

[Bảng 5 Các yêu cầu kiểm thử 56](#_Toc493967736)

[Bảng 6 Các tình huống kiểm thử 56](#_Toc493967737)

# DANH MỤC HÌNH ẢNH

[Hình 1 Quá trình hình thành và phát triển của Xamarin 11](#_Toc493952979)

[Hình 2 Kiến trúc tầng của xamarin 12](#_Toc493952980)

[Hình 3 Kiến trúc liên kết của Xamarin 12](#_Toc493952981)

[Hình 4 Chia sẻ mã nguồn trong Xamarin.Form 16](#_Toc493952982)

[Hình 5 Chia sẻ mã nguồn trong từng nền tảng 17](#_Toc493952983)

[Hình 6 Kết quả ứng dụng tạo bằng Xamarin.Form 17](#_Toc493952984)

[Hình 7 Hoạt động chia sẻ mã nguồn của Shared Project 18](#_Toc493952985)

[Hình 8 Hoạt động chia sẻ mã nguồn của PCL 20](#_Toc493952986)

[Hình 9 Mô hình MVVM 23](#_Toc493952987)

[Hình 10 Giao diện thiết kế bới XAML 25](#_Toc493952988)

[Hình 11 Các Page hỗ trợ trong Xamarin.Form 29](#_Toc493952989)

[Hình 12 Các layout hỗ trợ trong Xamarin.Form 29](#_Toc493952990)

[Hình 13 Cấu trúc dữ liệu mô tả video 30](#_Toc493952991)

[Hình 14 Render chữ viết tay với SkiaSharp 32](#_Toc493952992)

[Hình 15 Hiển thị dữ liệu PDF với PDFjs 33](#_Toc493952993)

[Hình 16 Cài đặt môi trường phát triển xamarin 35](#_Toc493952994)

[Hình 17 Tạo dự án mới với Xamarin 36](#_Toc493952995)

[Hình 18 Kết nối với máy MAC 37](#_Toc493952996)

[Hình 19 Mô hình hệ thống 38](#_Toc493952997)

[Hình 20 Biểu đồ usecase tổng quát 39](#_Toc493952998)

[Hình 21 Usecase liệt kê bài giảng 40](#_Toc493952999)

[Hình 22 Usecase quản lý đăng tài khoản 41](#_Toc493953000)

[Hình 23 Usecase điều khiển bài giảng 42](#_Toc493953001)

[Hình 24 Sequence đăng nhập hệ thống 43](#_Toc493953002)

[Hình 25 Activity đăng nhập hệ thống 44](#_Toc493953003)

[Hình 26 Sequence Đăng kí tài khoản 45](#_Toc493953004)

[Hình 27 activity Đăng kí tài khoản 45](#_Toc493953005)

[Hình 28 Sequence Liệt kê bài giảng 46](#_Toc493953006)

[Hình 29 Activity Liệt kê bài giảng 47](#_Toc493953007)

[Hình 30 Sequence Học bài 47](#_Toc493953008)

[Hình 31 Activity Học bài 48](#_Toc493953009)

MỤC LỤC

[LỜI CAM ĐOAN 1](#_Toc493959013)

[LỜI MỞ ĐẦU 2](#_Toc493959014)

[LỜI CẢM ƠN 3](#_Toc493959015)

[DANH MỤC BẢNG BIỂU 4](#_Toc493959016)

[DANH MỤC HÌNH ẢNH 5](#_Toc493959017)

[CHƯƠNG 1 : ĐẶT VẤN ĐỀ 7](#_Toc493959018)

[1. Giới thiệu việc giáo dục trực tuyến 7](#_Toc493959019)

[1.1 Các loại hình đào tạo trực tuyến 8](#_Toc493959020)

[1.2 Lợi ích học trực tuyến 8](#_Toc493959021)

[2. Các vấn đề của việc học trực tuyến hiện nay 10](#_Toc493959022)

[2.1. Vấn đề phía người học 10](#_Toc493959023)

[2.2. Vấn đề phía nội dung học tập 10](#_Toc493959024)

[2.3 Vấn đề về yếu tố công nghệ 10](#_Toc493959025)

[CHƯƠNG 2 : GIẢI PHÁP KHẮC PHỤC VẤN ĐỀ VÀ ĐỊNH HƯỚNG CÔNG NGHỆ 12](#_Toc493959026)

[1. Giải pháp 12](#_Toc493959027)

[2. Phương pháp tiếp cận và định hướng công nghệ 13](#_Toc493959028)

[2.1 Phương pháp tiếp cận 13](#_Toc493959029)

[2.2 Công nghệ thực hiện 13](#_Toc493959030)

[2.2.1. Nền tảng Xamarin 13](#_Toc493959031)

[2.2.2 Những ưu điểm , nhược điểm của Xamarin 16](#_Toc493959032)

[2.2.2.1 Ưu điểm 16](#_Toc493959033)

[2.2.2.2 Nhược điểm 18](#_Toc493959034)

[2.2.3 Phát triển ứng dụng di động đa nền tảng dựa trên Xamarin Form 18](#_Toc493959035)

[2.2.3.1 Shared Project 21](#_Toc493959036)

[2.2.3.2 Portable Class Libraries (PCL) 22](#_Toc493959037)

[2.2.4 Mô hình MVVM 26](#_Toc493959038)

[2.2.5 Các đối tượng giao diện trong Xamarin Form 27](#_Toc493959039)

[2.2.5.1. Ngôn ngữ thiết kế giao diện XAML 27](#_Toc493959040)

[2.2.5.2. Các thành phần giao diện trong Xamarin.Form 28](#_Toc493959041)

[2.2.6 Mô hình hóa dữ liệu video 33](#_Toc493959042)

[2.2.7 Render dữ liệu chữ viết tay với SkiaSharp 34](#_Toc493959043)

[2.2.8 Hiển thị bài giảng PDF 36](#_Toc493959044)

[CHƯƠNG 3 : XÂY DỰNG ỨNG DỤNG E\_LEARNINGCLIENT 38](#_Toc493959045)

[1. Cài đặt môi trường phát triển 38](#_Toc493959046)

[2. Ứng dụng E-LearningClient 41](#_Toc493959047)

[2.1 Kiến trúc hệ thống 41](#_Toc493959048)

[2.2. Thiết kế ứng dụng 42](#_Toc493959049)

[2.2.1 Các biểu đồ Usecase của ứng dụng 42](#_Toc493959050)

[2.2.1.1 Usecase tổng quát 42](#_Toc493959051)

[2.2.1.2. Usecase liệt kê bài giảng 43](#_Toc493959052)

[2.2.1.3 Usecase quản lý đăng tài khoản 44](#_Toc493959053)

[2.2.1.3 Usecase điều khiển bài giảng 45](#_Toc493959054)

[2.2.2. Biểu đồ tuần tự và biểu đồ hoạt động 46](#_Toc493959055)

[2.2.2.1 Đăng nhập hệ thống 46](#_Toc493959056)

[2.2.2.2 Đăng kí tài khoản 48](#_Toc493959057)

[2.2.2.3 Liệt kê bài giảng 49](#_Toc493959058)

[2.2.2.4 Học bài 50](#_Toc493959059)

[2.2.3 Thiết kế giao diện 52](#_Toc493959060)

[2.3 Một số kịch bản kiểm thử và kết quả 56](#_Toc493959061)

[2.3.1. Yêu cầu kiểm thử 56](#_Toc493959062)

[2.3.2 Tình huống kiểm thử 56](#_Toc493959063)

[CHƯƠNG 4 : KẾT LUẬN VÀ HƯỚNG PHÁT TRIỂN 58](#_Toc493959064)

[4.1 Kết quả đạt được 58](#_Toc493959065)

[4.2 Định hướng phát triển 58](#_Toc493959066)

# CHƯƠNG 1 : ĐẶT VẤN ĐỀ

## 1. Giới thiệu việc giáo dục trực tuyến

Giáo dục trực tuyến là việc học mà học sinh tham gia thông qua các bài giảng do giáo viên thiết kế. Bài giảng được thiết kế theo đúng trình tự của một tiết học và đảm bảo được các tiêu chí như: công nghệ, nội dung, kiến thức, sư phạm và phương pháp truyền đạt để đáp ứng được nhu cầu tự học của học sinh ở mọi lúc, mọi nơi (online hoặc offline), giảm chi phí, thời gian đi lại, không cần trường lớp. Việc đưa các bài giảng trực tuyến đến người học được thực hiện thông qua các phương tiện điện tử: tài liệu được gửi cho học sinh bằng email, học sinh học trên website, học qua đĩa CD - Rom multimedia…

### 1.1 Các loại hình đào tạo trực tuyến

Hiện nay có 5 loại hình thức đào tạo trực tuyến đó là:

* *Đào tạo dựa trên công nghệ (TBT - Technology-Based Training)* là hình thức đào tạo có sự áp dụng công nghệ, đặc biệt là dựa trên công nghệ thông tin.
* *Đào tạo dựa trên máy tính không nối mạng (CBT - Computer-Based Training)* là hình thức đào tạo sử dụng các ứng dụng (phần mềm) đào tạo trên các đĩa CD-ROM hoặc cài trên các máy tính độc lập, không nối mạng, không có giao tiếp với thế giới bên ngoài. Thuật ngữ này được hiểu đồng nhất với thuật ngữ CD-ROM Based Training.
* *Đào tạo dựa trên web (WBT - WebBased Training)* là hình thức đào tạo sử dụng công nghệ web. Nội dung học, các thông tin về người học và quản lí khóa học được lưu trữ trên máy chủ và người dùng có thể dễ dàng truy nhập thông qua trình duyệt web. Người học có thể giao tiếp với nhau và với giáo viên, sử dụng các chức năng trao đổi trực tiếp, diễn đàn, e-mail... và có thể nghe được giọng nói và nhìn thấy hình ảnh của người giao tiếp với mình.
* *Đào tạo trực tuyến (Online Learning/Training)* là hình thức đào tạo có sử dụng kết nối mạng để thực hiện việc học: lấy tài liệu học, xem chương trình, giao tiếp giữa người học với nhau và với giáo viên...
* *Đào tạo từ xa (Distance Learning)* là hình thức đào tạo trong đó người dạy và người học không ở cùng một chỗ, thậm chí không cùng một thời điểm. Ví dụ như việc đào tạo sử dụng công nghệ hội thảo cầu truyền hình hoặc công nghệ web.

### 1.2 Lợi ích học trực tuyến

Học trực tuyến có rất nhiều lợi ích đa dạng và phong phú khi xét ở các góc độ khác nhau: về phía người học, về phía cơ sở đào tạo, về xã hội... Sau đây là một số lợi ích cơ bản:

*Thứ nhất*, học trực tuyến giúp người học vượt qua rào cản về không gian và thời gian. Với hình thức học này, người học có thể đăng kí và theo học bất cứ thời gian nào mình muốn. Học viên có thể học bất cứ lúc nào, tại bất kì nơi đâu… Tận dụng được nguồn giảng viên chất lượng cao từ nhiều nơi trên thế giới, nội dung truyền tải nhất quán, phù hợp với yêu cầu của người học.

*Thứ hai*, Học trực tuyến giúp cho người học chủ động hơn: dễ dàng tự định hướng và tự điều chỉnh việc học tập của bản thân. Việc đăng kí và chứng thực học viên đơn giản và thuận tiện. Người học có khả năng tự kiểm soát cao thông qua việc tự đặt cho mình tốc độ học phù hợp, bỏ qua những phần hướng dẫn đơn giản không cần thiết mà vẫn đáp ứng được tiến độ chung của khóa học. Đối với học viên, kèm theo việc tăng khả năng tiếp tục đáp ứng được công việc, giảm thời gian học, học viên còn có thể học mọi lúc, mọi nơi, cho phép học viên có thể hoàn thành chương trình đào tạo một cách thuận tiện ngoài giờ làm việc hay ở nhà.

*Thứ ba,* Học trực tuyến giúp cho người học rèn luyện và phát triển khả năng tự học. Trong suốt quá trình học trực tuyến, học viên phải tự xây dựng kế hoạch học tập, lựa chọn môn học, tài liệu cần thiết, tự thực hiện các yêu cầu của khóa học... nhờ thế mà khả năng tự học mỗi ngày một tốt hơn.

*Thứ tư,* Học trực tuyến làm tăng lượng thông tin một cách rõ rệt, kiến thức thu được rất đa dạng và phong phú. Nhờ tính tương tác và hợp tác cao, dễ tiếp cận và thuận tiện, đào tạo trực tuyến tạo một môi trường giao tiếp thuận lợi giữa học viên với giáo viên, giữa học viên với nhau... Khi mọi người được trao đổi với giáo viên và bạn bè trong lớp, họ có thể tiếp thu nhiều thông tin hơn từ các nguồn khác nhau. Mặt khác, kết quả đào tạo cũng được tự động hóa và được thông báo nhanh chóng, chính xác, khách quan.

*Thứ năm,* Rút ngắn thời gian đào tạo. Học viên tận dụng được mọi thời gian rảnh rỗi, giảm thiểu thời gian rời khỏi văn phòng hoặc gia đình. Cơ sở đào tạo cũng dễ dàng kiểm soát thời gian thực hiện khóa học.

*Cuối cùng,* Chi phí cho việc học tập được giảm thiểu: chi phí cho người học, chi phí cho tổ chức và quản lí đào tạo. Nội dung khóa học có thể sử dụng lại được với các học viên khác nhau. Cắt giảm được chi phí in ấn, xuất bản và phân phối tài liệu, lương của giáo viên, chi phí thuê phòng học, chi phí đi lại, ăn ở cho học viên. Tiết kiệm được một khoản tiền lớn do giảm được chi phí đi lại.

Ưu điểm nổi bật của lớp học truyền thống đó là học sinh dễ dàng trao đổi trực tiếp với bạn bè, thầy cô giáo, đây cũng chính là thách thức không nhỏ đối với lớp học trực tuyến. Chính vì vậy khi thiết kế bài giảng, giáo viên phải đặt r a tiêu chí: “Học có sự hợp tác, phối hợp”. Không chỉ giáo viên là người cung cấp kiến thức, đưa ra các vấn đề mà học sinh có thể kết nối, trao đổi thông tin với giáo viên thông qua hòm thư trực tuyến. Ngược lại giáo viên cũng nắm được các thông tin phản hồi trong bài dạy của mình. Học sinh cũng có thể dễ dàng trao đổi thông tin với nhau qua các diễn đàn (forum), hội thoại trực tuyến (chat), thư từ (email)…

## 2. Các vấn đề của việc học trực tuyến hiện nay

Bên cạnh những lợi ích mà giáo dục trực tuyến mang lại, không thể không kể đến những vấn đề hạn chế nói chung của việc học trực tuyến hiện nay.

### 2.1. Vấn đề phía người học

Tham gia học tập trực tuyến đòi hỏi người học phải có khả năng làm việc độc lập với ý thức tự giác cao độ. Bên cạnh đó, cũng cần thể hiện khả năng hợp tác, chia sẻ qua mạng một cách hiệu quả với giảng viên và các thành viên khác .Người học cũng cần phải biết lập kế hoạch phù hợp với bản thân, tự định hướng trong học tập, thực hiện tốt kế hoạch học tập đã đề ra.

### 2.2. Vấn đề phía nội dung học tập

Trong nhiều trường hợp, không thể và không nên đưa ra các nội dung quá trừu tượng, quá phức tạp. Đặc biệt là nội dung liên quan tới thí nghiệm, thực hành mà Công nghệ thông tin không thể hiện được hay thể hiện kém hiệu quả. Hệ thống học trực tuyến cũng không thể thay thế được các hoạt động liên quan tới việc rèn luyện và hình thành kỹ năng, đặc biệt là kỹ năng thao tác và vận động.

### 2.3 Vấn đề về yếu tố công nghệ

Sự hạn chế về kỹ năng công nghệ của người học sẽ làm giảm đáng kể hiệu quả, chất lượng dạy học trực tuyến . Bên cạnh đó, hạ tầng công nghệ thông tin (mạng internet, băng thông, chi phí…) cũng ảnh hưởng đảng kể tới tiến độ, chất lượng học tập.

Chi tiết hơn vào việc tạo bài giảng, trình chiếu hiển thị bài giảng, ta còn thấy các bài giảng thường là những video dàn dựng, quay lại quá trình truyền tải kiến thức cũng như tài liệu đến với học viên. Để bài giảng có chất lượng tốt, các video thường được thực hiện một cách công phu ở các studio hay hỗ trợ bởi các thiết bị ghi hình, ghi âm chuyên nghiệp. Ngoài ra, bài giảng video đơn thuần mang tính tương tác một chiều, không có chiều sâu. Để soạn được bài giảng tương tác cao lại cần có những phần mềm chuyên dụng. Điều này dẫn đến các vấn đề là :

* Chi phí tốn kém cho việc tạo bài giảng ở các studio
* Việc tạo bài giảng hỗ trợ bởi các thiết bị ghi hình , ghi âm đắt tiền
* Sản phẩm video dàn dựng có dung lượng lớn, gây khó khăn cho việc lưu trữ, truyền tải, triển khai bài giảng trực tuyến thời gian thực khó khăn. Đặc biệt là khi có nhiều học viên truy cập trong một thời gian ngắn.
* Bài giảng ít tương tác, để soạn bài tương tác tốt cần những phần mềm chuyên dụng với chi phí bản quyền đắt đỏ.

Dựa vào những phân tích về hạn chế của việc tạo bài giảng video, trình chiếu bài giảng cùng các vấn đề đặt ra, ở chương tiếp theo, tôi xin trình bày giải pháp khắc phục và định hướng công nghệ thực hiện

# CHƯƠNG 2 : GIẢI PHÁP KHẮC PHỤC VẤN ĐỀ VÀ ĐỊNH HƯỚNG CÔNG NGHỆ

## 1. Giải pháp

Từ những vấn đề nêu trên, trong khuôn khổ đề tài tốt nghiệp, tôi xin đề ra giải pháp đơn giản hóa việc tạo bài giảng video, giảm dung lượng lưu trữ , truyền dẫn; không cần đến sự hỗ trợ của các thiết bị ghi hình hiện đại, không cần dùng đến những phần mềm tạo bài giảng chuyên dụng, bài giảng hiển thị tốt trên máy học viên với đa nền tảng, tính tương tác giữa học viên và người thầy cao, hỗ trợ tối đa các loại thiết bị.

Bằng việc định dạng lại dữ liệu bài giảng, với đầu vào là một tệp bao gồm :

* Nội dung bài giảng lưu trữ tọa độ nét bút của người thầy, các trang tài liệu cơ bản (pdf) , cấu trúc thông tin bài giảng, thông tin các hành động tương tác (kiểm tra giữa bài, câu hỏi trắc nghiệm …)
* Phần âm thanh ghi âm lời giảng của thầy xuyên suốt bài giảng.
* Hình ảnh người thầy nhằm hiển thị thông tin giảng viên

Qua đó, chương trình sẽ thu thập đủ thông tin người thầy đã viết ra, lời thầy giảng dạy trong suốt bài giảng, hiển thị lại trên máy tính của học viên một cách liên tục, đồng bộ giữa lời giảng và nét bút. Từ đó, dữ liệu hiển thị ra sẽ tương tự như một video giảng bài truyền thống, khắc phục được tất cả các hạn chế về mặt chi phí, dung lượng, truyền tải cho người thầy khi tạo bài giảng, đồng thời tăng tính tương tác giữa bài giảng với học viên đồng thời không làm mất tính gần gũi giữa người thầy và học viên khi hình ảnh người thầy sẽ hiển thị liên tục trong suốt bài giảng. Ngoài ra, với việc hỗ trợ chạy đa nền tảng, người dùng có thể truy cập ứng dụng , học tập trên nhiều loại thiết bị khác nhau, ở nhiều vị trí địa lý khác nhau, chỉ cần có smartphone và internet. Thông qua ứng dụng, học viên có thể xem lại bài giảng, làm bài luyện tập, bài kiểm tra, xem tài liệu, khóa học, tải tài liệu về máy, quản lý tài khoản học của mình. Như vậy, có thể thấy được các mục đích của đề tài bao gồm :

* Xây dựng kênh tương tác, chia sẻ và lưu trữ tài liệu giữa giáo viên và học viên
* Cấu trúc lại dữ liệu bài giảng theo hướng đơn giản hóa việc tạo bài, giảm dung lượng bài giảng mà không làm mất đi bản chất việc học hay sự gần gũi giữa học viên và người thầy
* Hỗ trợ việc học trực tuyến thời gian thực một cách thông suốt và đơn giản hơn trong việc truyền tải
* Tạo ta một nơi tương tác giữa học viên và giảng viên ở cả khía cạnh kiến thức và hỏi đáp với phương pháp tiếp cận các hình thức tương tác phong phú
* Tạo ra một mơi có thể nhận, phát thông báo cho toàn thể học viên trong cùng lớp hoặc xin ý kiến riêng người thầy
* Tạo ra một nơi thảo luận, hỏi đáp như một forum

## 2.Phương pháp tiếp cận và định hướng công nghệ

### 2.1 Phương pháp tiếp cận

Để hiện thực hóa các giải pháp nêu trên, chúng ta tiếp cận đề tài theo 2 hướng:

*Hướng thứ nhất là đơn giản hóa việc lưu trữ, truyền tải bài giảng*:

Để đơn giản hóa lưu trữ, truyền tải bài giảng, chúng ta cần tạo ra một cấu trúc dữ liệu mới, mô tả quá trình dạy học của giáo viên trong cả bài giảng, bao gồm :

* Dữ liệu bài giảng (nét bút, trình tự trang tài liệu trình chiếu) được lưu trữ dưới tệp XML.
* Dữ liệu âm thanh : ghi lại quá trình giảng bài của người giáo viên với định dạng 3gp
* Hình ảnh người giáo viên : hiển thị xuyên suốt bài giảng, tăng sự gần gũi giữa người thầy và học viên, được lưu trữ dưới định dạng jpg

*Hướng thứ hai là hỗ trợ chương trình chạy trên đa nền tảng:*

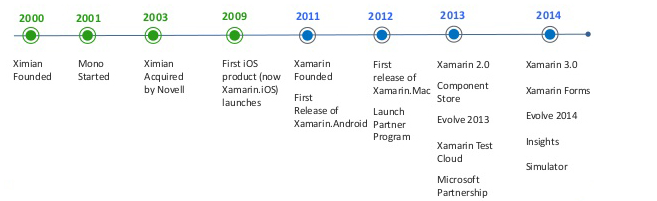
Để hỗ trợ chương trình chạy trên đa nền tảng, chúng ta lựa chọn công nghệ Cross-platform Xamarin, được phát triển bởi Microsoft. Đây là công nghệ mới giúp tạo ra các ứng dụng đa nền tảng với cộng đồng phát triển đang ngày một lớn. Chi tiết công nghệ, tôi xin trình bày ở phần tiếp theo.

Như vậy , bằng các phương pháp tiếp cận này, chúng ta đang dần hiện thực hóa được giải pháp khắc phục cho các vấn đề nêu trên. Tiếp theo, tôi xin trình bày chi tiết công nghệ thực hiện, cách cài đặt và sử dụng.

### 2.2 Công nghệ thực hiện

#### 2.2.1. Nền tảng Xamarin

Lập trình ứng dụng di động là xu hướng mạnh mẽ của ngành công nghệ thông tin và đang rất cần nguồn nhân lực khổng lồ. Trên thị trường ứng dụng di động hiện nay, 3 hệ điều hành chiếm thị phần cao nhất là: Android, iOS và Window Phone, tiếp sau là một số hệ điều hành khác như BlackBerry, Sailfish, Firefox, … Xamarin là nền tảng hỗ trợ phát triển các ứng dụng di động trên từng nền tảng hay đa nền tảng. Các môi trường di động được hỗ trợ bởi Xamarin là iOS, Android, Window phone và Windows 8.1. Xamarin giúp sử dụng ngôn ngữ C# để xây dựng App native(ứng dụng gốc) cho cả iOS, Android lẫn Window một cách nhanh chóng và dễ dàng hỗ trợ đầy đủ các tính năng mạnh mẽ. Sản phẩm Xamarin đơn giản hóa việc tạo ra và duy trì hiệu suất cao, nền tảng ứng dụng di động với mục tiêu là điện thoại, máy tính bảng và các thiết chạy iOS, Android và Windows. Chỉ phải lập trình một lần để xây dựng ứng dụng gốc cho nhiều nền tảng trên một cơ sở dữ liệu code C# được chia sẻ, sử dụng cùng một IDE. Được xây dựng và phát triển từ năm 2011 (khởi nguồn từ dự án Mono open source - 2001) , giờ đây cộng đồng phát triển Xamarin đang ngày càng lớn mạnh, hứa hẹn là một giải pháp tối ưu cho việc phát triển ứng dụng di động trong tương lai. Hình ảnh dưới đây là quá trình hình thành và phát triển của nền tảng Xamarin.



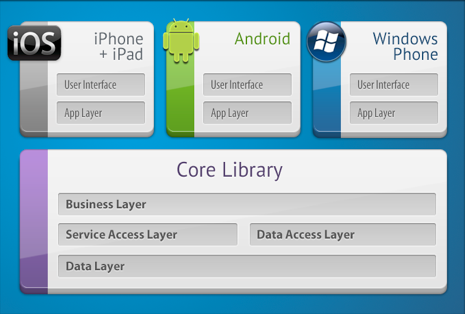
Hình 1 Quá trình hình thành và phát triển của Xamarin

***Kiến trúc Xamarin***

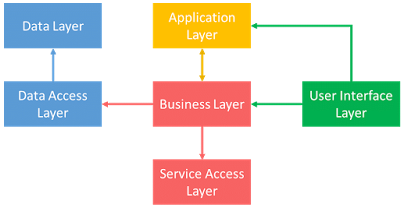
Ứng dụng di động đa nền tảng thông thường được chia kiến trúc thành 6 tầng :

* *Data Layer* : Chính là cơ sở dữ liệu của ứng dụng, quản lý dữ liệu lưu trữ một cách vật lý (ví dụ SQLite-Net) như cơ sở dữ liệu, XML files. Tầng này, đơn giản chỉ được sử dụng bởi Data Access Layer.
* *Data Access Layer* : Bao gồm các APIs hỗ trợ việc truy xuất, thêm xóa sửa dữ liệu như các phương thức truy cập danh sách, tập hợp dữ liệu, các dữ liệu cơ bản đồng thời hỗ trợ cập nhật dữ liệu.
* *Business Layer* : Đây là lớp xử lý chính của ứng dụng, bao gồm các thực thể mô tả (Model) và các phương thức xử lý logic.
* Service Access Layer : Thực hiện chức năng liên kết với các dịch vụ bên ngoài ứng dụng. Ví dụ như sử dụng dịch vụ trên cloud hoặc các dịch vụ do bên thứ 3 cung cấp.
* *Application Layer* : Chứa các lớp có sử dụng đến API đặc thù của nền tảng và thường không thể chia sẻ mã nguồn ở lớp này cho nền tảng khác, với chức năng kết nối/quy đổi dữ liệu dữa tầng UI Interface và tầng Bussiness.
* *User Interface Layer* : Bao gồm các đối tượng giao diện để giao tiếp với người dùng.

Xamarin với sứ mệnh phát triển ứng dụng di động đa nền tảng cũng tuân theo một cách chặt chẽ kiến trúc trên

****

Hình 2 Kiến trúc tầng của xamarin



Hình 3 Kiến trúc liên kết của Xamarin

#### 2.2.2 Những ưu điểm , nhược điểm của Xamarin

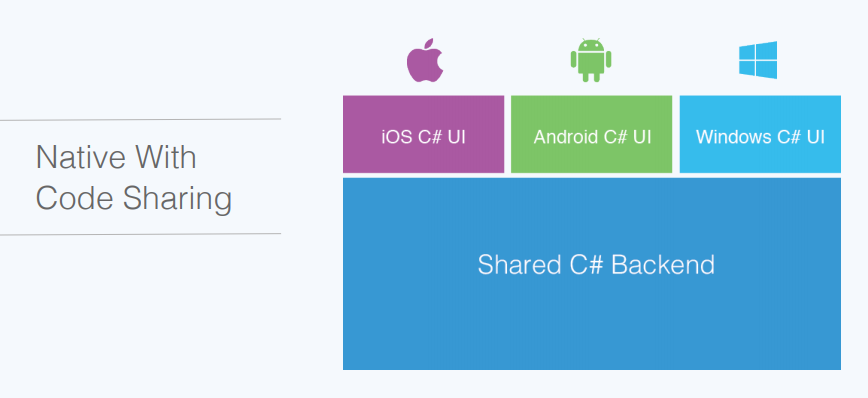
##### 2.2.2.1 Ưu điểm

*Thứ nhất,* Write Once, Run Anywhere : Đây chính là câu nói tôn chỉ của Xamarin, cũng là ưu điểm lớn nhất của nó. Trước đây , để một ứng dụng cùng trước năng chạy trên nhiều nền tảng, nhà phát triển phải viết ứng dụng nhiều lần trên mỗi nền tảng riêng biệt. Mọi chi phí (phát triển, triển khai, bảo trì) đều bị nhân lên.

|  |
| --- |
|  |

Ngôn ngữ native trên từng nền tảng

*Thứ hai,* Chia sẻ code ở mọi nơi : Khi tạo ứng dụng trên Xamarin, người lập trình sử dụng cùng ngôn ngữ, API và cấu trúc dữ liệu để chia sẻ trung bình 75% code trên tất cả các nền tảng phát triển điện thoại di động. Logic ứng dụng này có thể dễ dàng chia sẻ trên nhiều nền tảng. Qua đó có thể giảm đáng kể chi phí và thời gian phát triển ứng dụng di động cho 3 nền tảng phổ biến nhất. Việc sử dụng xamarin, có thể tự tạo ra các plugin, control riêng biệt giống như thư viện dùng chung cho nhiều nền tảng.



*Thứ ba,* Hiệu năng như native : Không giống như phương pháp kết hợp truyền thống dựa trên các công nghệ web, một ứng dụng đa nền tảng được xây dựng với Xamarin cũng có thể xem vào hàng native. Các số liệu hiệu năng là tương đương khi so sánh với các số liệu hiệu năng của Java cho Android và Objective-C hoặc Swift cho ứng dụng phát triển ứng dụng iOS native. Hơn thế nữa, performance liên tục được cải thiện để phù hợp hoàn toàn với tiêu chuẩn của lập trình native. Nền tảng Xamarin cung cấp là giải pháp để testing và theo dõi hoạt động của ứng dụng. Xamarin Test Cloud kết hợp với công cụ Xamarin Test Recorder cho phép bạn chạy các UI test tự động và xác định các vấn đề về performance trước khi ứng dụng release. Tuy nhiên, dịch vụ này có tính phí.

*Thứ tư,* Ngôn ngữ phát triển là C# : Đây là ngôn ngữ phổ biến ,được đánh giá là dễ dùng, hỗ trợ bởi tập APIs rộng lớn của .Net Framework

*Thứ năm,* một đột phát của Xamarin trong việc phát triển ứng dụng di động đa nền tảng là Xamarin Form. Xamarin.Forms như một APIs : Xamarin.Forms là một API giúp xây dựng code giao diện người dùng có thể được chia sẻ trên các ứng dụng iOS, Android và Windows Phone với 100% C#. Xamarin.Forms bao gồm hơn 40 controls và bố cục, được ánh xạ tới các điều khiển gốc trong quá trình chạy. Để phục vu cho mục đích đa nền tảng của đề tài, tôi cũng sử dụng Xamarin.Forms. Ở phần tiếp theo, tôi sẽ trình bày chi tiết hơn về Xamarin.Forms.

*Cuối cùng ,* Chi phí bản quyền đã bao gồm trong visual studio : Trước năm 2016, mỗi bản quyền Xamarin studio professional là 999$ / năm, khá đắt đỏ cho nhà phát triển với mục đích thương mại. Từ năm 2016, sau khi sáp nhập vào Microsoft, bản quyền của Xamarin cũng được bao gồm trong bản quyền của visual studio. Đây là bước đi của MS giúp Xamarin đến với đa số nhà phát triển, qua đó làm cộng đồng xamarin ngày càng lớn mạnh.

##### 2.2.2.2 Nhược điểm

*Thứ nhất ,* Ứng dụng thực hiện chậm hơn : Cái giá của việc “Cross” chính là việc ứng dụng được thực hiện chậm hơn ứng dụng native. Việc ánh xạ, chuyển đổi các thành phần giao diện tương ứng, các service riêng biệt cho từng nền tảng chính là nguyên nhân của việc chậm này. Các kỹ sư phát triển Xamarin sớm nhận ra điều này và đang khắc phục để rút ngắn khoảng cách về thời gian thực thi với ứng dụng native.

*Thứ hai,* Dung lượng binary nặng hơn và yêu cầu bộ nhớ nhiều hơn : Ứng dụng Xamarin lớn hơn,nặng hơn so với ứng dụng native. So sánh với ứng dụng native nó chiếm nhiều hơn vài Mb so với Java/Objective C tương ứng. Kích thước của một ứng dụng code bằng xamarin là 3Mb, trong khi code bằng Objective C chỉ chiếm 172 Kb. Càng sử dụng nhiều API, càng nhiều lưu trữ bị chiếm trên thiết bị

*Thứ ba,* Khó khăn trong việc shared code ở các chức năng phức tạp.

Như vậy, chúng ta đã hiểu được các ưu, nhược điểm của Xamarin, cũng như tính lợi hại của việc sử dụng các ứng dụng phát triển từ Xamarin. Trong phần tiếp theo, tôi xin trình bày chi tiết công nghệ phát triển ứng dụng di động đa nền tảng sử dụng công nghệ Xamarin.Form. Đây cũng là công cụ chính để phát triển ứng dụng của đề tài.

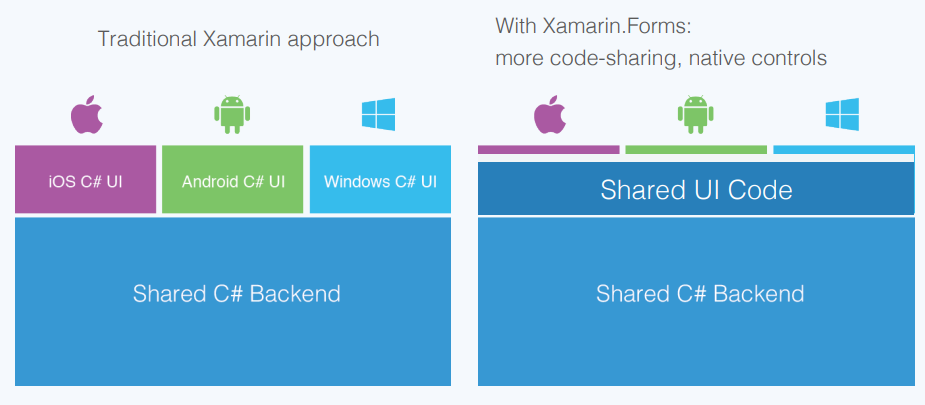
#### 2.2.3 Phát triển ứng dụng di động đa nền tảng dựa trên Xamarin Form

Xamarin Form là một công cụ để xây dựng các ứng dụng đa nền tảng có thể chạy được trên Windows, Android, iOS. Được phát triển bởi Microsoft nên có thể dùng C# để lập trình các ứng dụng bằng Xamarin. Công bố chính thức từ tháng 5/2014, Xamarin.Form đang thể hiện là công cụ xây dựng ứng dụng di động đa nền tảng hiệu quả bậc nhất hiện nay. Các phiên bản nền tảng ứng dụng được hỗ trờ bởi Xamarin.Form hiện nay :

* Android 4.0 +
* iOS 6.1 +
* Windows 10
* Windows 8.1
* Windows Phone 8.1

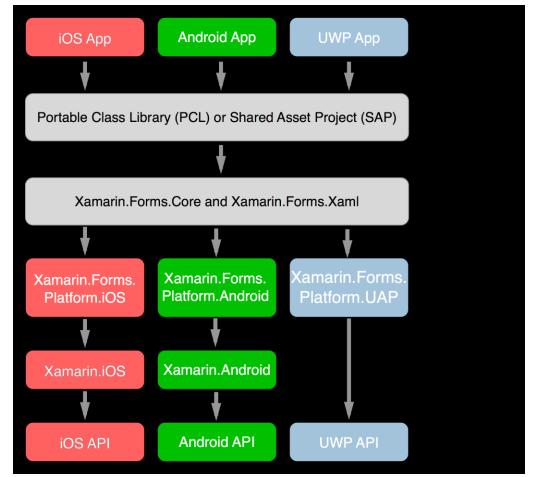
Xamarin.Form cho phép tạo khuôn mẫu giao diện ứng dụng một cách nhanh chóng, kể cả những ứng dụng phức tạp. Nó gần như tạo ra được ứng dụng native mà không gặp bất kì giới hạn nào về thư viện, APIs, hiệu năng… Ngoài ra, một ứng dụng có thể tạo bằng sự kết hợp của cả Xamarin Form và code native mà không gặp bất kì sự xung đột nào.

Xamarin Form được phát triển trên Xamarin truyền thống, nên nó được kế thừa hoàn toàn giống kiến trúc của Xamarin. Điểm mạnh của nó là có thể chia sẻ tối đa các thành phần code cho nhiều nền tảng (từ module giao diện cho đến business logic), nhờ đó, hiệu suất chia sẻ code cao rõ rệt. Hình bên dưới mô tả khác biệt của sự chia sẻ tài nguyên giữa xamarin form và xamarin truyền thống

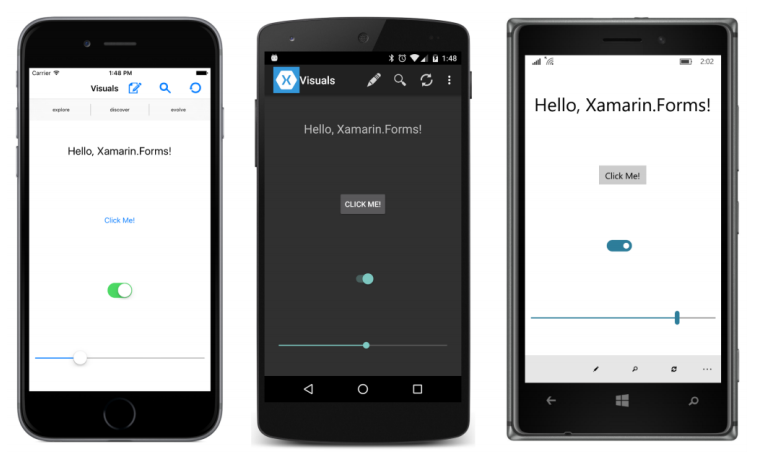


Hình 4 Chia sẻ mã nguồn trong Xamarin.Form

Như hình trên, có thể dễ dàng nhận ra việc chia sẻ tài nguyên của xamarin form tốt hơn rõ rệt. Với xamarin form , nhà phát triển có thể tạo ra những module dùng chung (giao diện và logic) cho mọi nền tảng. Nhưng cũng không mất tính native, chúng ta cũng có thể tạo những control, logic nativie vào dự án. Hình bên dưới mô tả chi tiết hơn về kiến trúc xamarin form trong một ứng dụng



Hình 5 Chia sẻ mã nguồn trong từng nền tảng

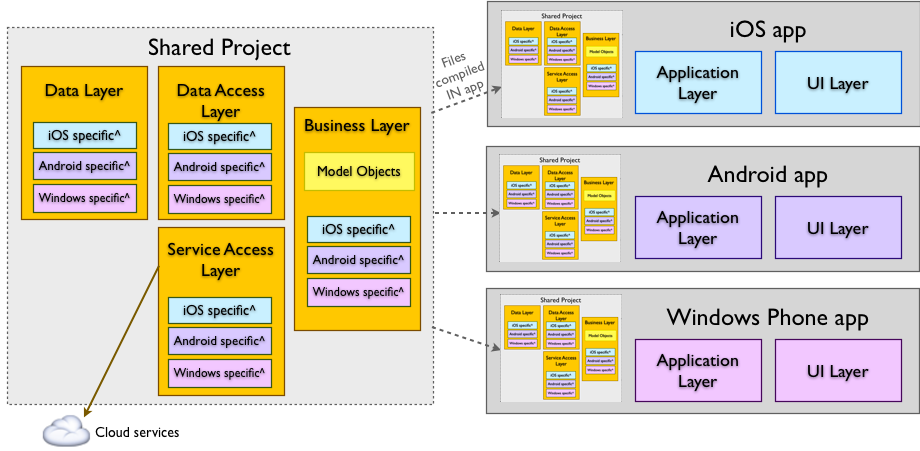
Và kết quả sau khi chuyển đổi các thành phần giao diện

Hình 6 Kết quả ứng dụng tạo bằng Xamarin.Form

Xamarin Form đang tuân theo quy tắc MVVM (*sẽ được trình bày ở phần tiếp*) , các tầng thành phần giống như ứng dụng Xamarin truyền thống. Chỉ khác là có nhiều thành phần có thể gom về một module dùng chung. Có 2 cách tiếp cần Xamarin Form theo hướng dùng chung, chia sẻ tài nguyên phổ biến nhất là thư viện khả chuyển (Portable Library) và dự án chia sẻ (Shared Project).

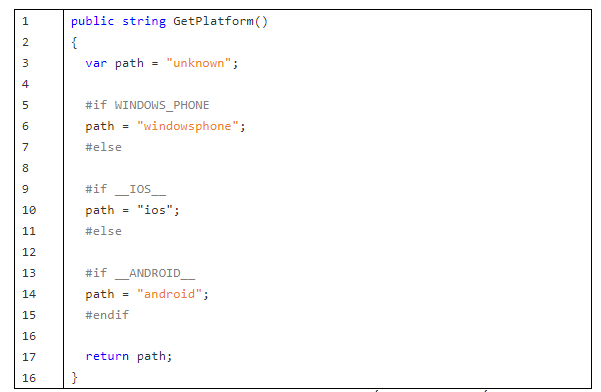
##### 2.2.3.1 Shared Project

Shared Project là kiến trúc hỗ trợ nhà phát triển viết code có thể chia sẻ giữa nhiều nền tảng trong một ứng dụng xamarin. Sauk hi biên dịch, chúng tạo ra các thành phần tham chiếu để có thể liên kết đến các project nền tảng riêng biệt. Ngoài chia sẻ code, Shared project có thể chia sẻ cả file dùng chung cho tất cả những project con. Không giống như thư viện (chuyên dùng để chia sẻ code), Shared project không sinh ra bất kì file thư viện nào sau khi biên dịch, thay vào đó, mã nguồn được biên dịch thành các thành phần riêng biệt mà nó được tham chiếu.



Hình 7 Hoạt động chia sẻ mã nguồn của Shared Project

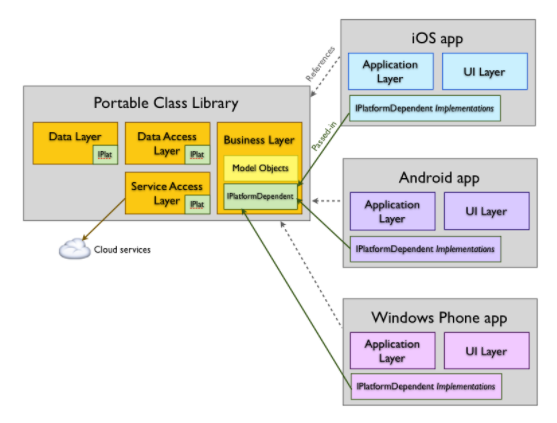
Mã nguồn trong shared project có thể bao gồm các thành phần code của từng nền tảng sau khi biên dịch (phụ thuộc vào nền tảng đích người phát triển hướng tới). Shared project chỉ có thể chia sẻ đến các project đích , không thể chia sẻ mã nguồn cho các project khác thậm chí project cùng loại. Hình bên dưới là đoạn mã nguồn thực thi việc shared code tương ứng với mỗi nền tảng



Có thể thấy, Shared Project tuy đơn giản và trực quan nhưng lại rất khó quản lý đối với các dự án lớn và có nhiều người cùng tham gia, vì khi viết một dòng lệnh trong Shared ta phải chắc chắn các project liên kết đều có thể hiểu được nó. Ngoài ra, khi tạo một Class mới hoặc khai báo hàm, biến cần phải đảm bảo trong tất cả project liên kết không bị trùng để tránh tình trạng “ambiguous”.

##### 2.2.3.2 Portable Class Libraries (PCL)

Một cách chia sẻ mã nguồn khác là sử dụng Portable Class Library (PCL). Đối với Class Library thông thường, sau khi biên dịch, file DLL chỉ có thể hoạt động trên một loại nền tảng nhất định. Ngược lại, khi tạo project PCL, ta có thể lựa chọn các nền tảng cần hỗ trợ. Từ danh sách các nền tảng được chọn, Visual Studio tính toán và chọn một phiên bản thích hợp của thư viện .NET Portable để import. Ví dụ trong .NET Portable phiên bản 344 hỗ trợ các nền tảng .Net 4.5, Xamarin.iOS, Xamarin.Android, Windows 8, Windows 8.1 và Silverlight 5. Thư viện .NET Portable thực chất bao gồm các Interface để lập trình viên giao tiếp kèm theo Attribute cho biết thư viện thực sự cần gọi đến là gì. Cách thức hoạt động của PCL được mô tả trong Hình bên dưới



Hình 8 Hoạt động chia sẻ mã nguồn của PCL

PCL được hỗ trợ từ phiên bản Xamarin Android 4.10.1, Xamarin iOS 7.0.4. Việc giao tiếp với PCL chẳng khác gì sử dụng một thư viện thông thường. Điều này giúp lập trình viên dễ dàng làm quen sử dụng và quản lý các dự án lớn. Tuy nhiên, có một điều bất tiện trong PCL: Microsoft không cung cấp cơ chế chỉ định một phần đặc biệt chỉ được sử dụng trong một nền tảng cụ thể như Shared Project. Nhưng thật may mắn, Xamarin có hỗ trợ một công cụ đặc biệt có tên là DependencyService giúp khắc phục hạn chế này khi sử dụng PCL trong lập trình di động đa nền tảng. Qua đó ta có thể thấy, ưu / nhược điểm của PCL.

*Ưu điểm*

* Tập trung hóa code sharing, viết code và test trên cùng 1 project
* Thay đổi 1 thành phần code, trình biên dịch có thể hiểu và cập nhật lại cho toàn solution
* Bằng việc sinh ra các thư viện DLL tương ứng mỗi nền tảng, PCL project có thể được tham chiếu dễ dàng bởi các project khác.

*Nhược điểm*

* Chỉ có PCL gốc thư viện mới có thể chia sẻ cho nhiều project khác nhau, những thư viện của các nên tảng riêng biệt không thể chia sẻ code
* Microsoft không cung cấp cơ chế chỉ định một phần đặc biệt chỉ được sử dụng trong một nền tảng cụ thể như Shared Project , phải dùng đến Dependency service để thực hiện việc này

Để sử dụng DependencyService, đầu tiên ta cần tạo một Interface trong PCL :

|  |  |
| --- | --- |
| 1  2  3  4 | public interface PlayAudio  {     public void Play (String Source);  } |

Với mỗi nền tảng cần tạo các lớp thực hiện chính kế thừa từ interface và đăng ký lớp này với DependencyService.

Trong project Android:

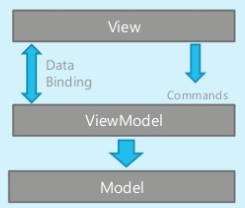
|  |  |
| --- | --- |
| 1  2  3  4  5  6  7  8  9  10  11  12  13  14  15  16  17 | [assembly: Xamarin.Forms.Dependency(typeof(PlayAudioImplement))]  namespace Project.Droid  {      public class PlayAudioImplement:PlayAudio      {          public void Play (String Source)          {              var Player = new MediaPlayer();              var fd = OpenFd(Source);              Player.Prepared += (s, e) => Player.Start();              Player.SetDataSource(fd.FileDescriptor,                                              fd.StartOffset,fd.Length);              Player.Prepare();          }      }  } |

Trong project Windows Mobile:

|  |  |
| --- | --- |
| 1  2  3  4  5  6  7  8  9  10  11  12  13  14  15 | [assembly: Xamarin.Forms.Dependency(typeof(PlayAudioImplement))]  namespace Project.Windows  {      public class PlayAudioImplement:PlayAudio      {          public async void Play (String Source)          {              Folder = await InstalledLocation.GetFolderAsync("Assets");              StorageFile sf = await Folder.GetFileAsync(Source);              MediaElement Player = new MediaElement();              Player.SetSource(await sf.OpenReadAsync(), sf.ContentType);              Player.Play();          }      }  } |

#### 2.2.4 Mô hình MVVM

MVVM không phải là một framework hay thư viện, APIs… chỉ đơn thuần là hướng dẫn người lập trình định nghĩa cấu trúc ứng dụng, hay nói cách khác giúp người lập trình kiến trúc hóa ứng dụng. MVVM là viết tắt của **M**odel – **V**iew – **V**iew**M**odel, hình ảnh dưỡi đânh sẽ mô tả mối quan hện giữa các thành phần trong mô hình này

******

Hình 9 Mô hình MVVM

***View*** : là mục chứa các thành phần định nghĩa giao diện (class) mà không kèm logic (code-behind). Views được sử dụng để kết hợp với các mô hình MVVM., nó dùng để cung cấp một sự chia tách gọn gàng của khái niệm giữa UI và presentation logic và data.

***Model*** : Đây là nơi định nghĩa các cấu trúc dữ liệu, đối tượng kèm thuộc tính đối tượng. Tất cả (hoặc một vài) các thuộc tính đều có thể liên kết (binding) đến một thành phần giao diện ở View. Hay nói cách khác, Model chứa data và bất kỳ liên kết validation, logic nghiệp vụ để chắc chắc tính toàn vẹn của data. Chúng được dùng như một phần của mô hình MVVM.

***ViewModel*** : Thông thường, một file giao diện sẽ có một file VM tương ứng. VM sẽ sử dụng các model nếu cần định nghĩa dữ liệu. Sự liên kết giữa View-ViewModel giúp chúng gửi và nhận dữ liệu. Nhờ đó ta tách code-behind của View và đưa xuống VM. Ngoài ra một lớp VM chứa presentation logic và trạng thái của ứng dụng. VM cần chứa các chức năng của ứng dụng. VM định nghĩa thuộc tính, lệnh điều khiển, và các sự kiện, để chuyển đổi controls trong view cần data-bind.

Như mô hình trên, để thay đổi các thuộc tính đối tượng , dữ liệu từ giao diện, cần thông qua VM bằng việc sử dụng lệnh (ICommand). Hay để dữ liệu sau khi thay đổi cập nhật lên giao diện người dùng cần thông qua hành động Data Binding. Bằng việc sử dụng IDE visual studio của Microsoft, người lập trình có thể dễ dàng thiết kế mô hình này trong dự án code.

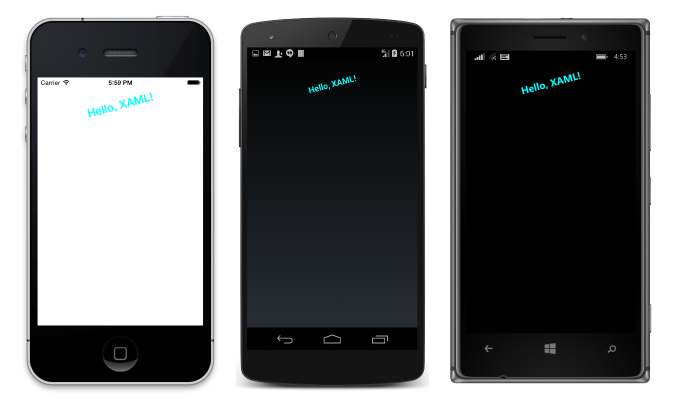
Như vậy, dựa vào mô hình MVVM thông dụng, chúng ta có thể thấy, ngoài những tầng riêng biệt cho mỗi nền tảng, kiến trúc của Xamarin cũng đang tuân theo mô hình MVVM với M tương ứng với Bussiness layer, V tương ứng với UI layer và VM tương ứng với Application layer. Đề tài cũng đang áp dùng mô hình MVVM trong việc thiết kế kết nối các module của dự án.

#### 2.2.5 Các đối tượng giao diện trong Xamarin Form

##### 2.2.5.1. Ngôn ngữ thiết kế giao diện XAML

Trước khi tìm hiểu trong xamarin Form hỗ trợ các thành phần giao diện nào, ta cần biết đến ngôn ngữ thiết kế giao diện trong xamarin form. Đó chính là XAML (e**X**tensible **A**pplication **M**arkup **L**anguage ). XAML là một thanhf phần quan trọng của Xamarin form. Bằng việc sử dụng XAML, nhà phát triển có thể thiết kế giao diện người dùng với toàn bộ control mà xamarin form hỗ trợ. nó cho phép nhà lập trình tạo ra giao diện người dùng bằng các thẻ đánh dấu thay vì code. Công việc thiết kế giao diện bằng Xamarin Form gần giống việc thiết kế một website bằng HTML. Bên dưới là ví dụ của XAML code thuần và giao diện xamarin form tương ứng.

|  |
| --- |
| <?xml version="1.0" encoding="utf-8" ?>  <ContentPage xmlns="http://xamarin.com/schemas/2014/forms"  xmlns:x="http://schemas.microsoft.com/winfx/2009/xaml"  x:Class="XamlSamples.HelloXamlPage"  Title="Hello XAML Page"  Padding="10, 40, 10, 10">  <Label Text="Hello, XAML!"  VerticalOptions="Start"  HorizontalTextAlignment="Center"  Rotation="-15"  IsVisible="true"  FontSize="Large"  FontAttributes="Bold"  TextColor="Aqua" />  </ContentPage> |

******

Hình 10 Giao diện thiết kế bới XAML

##### 2.2.5.2. Các thành phần giao diện trong Xamarin.Form

Xamarin.Form hỗ trợ trên 40 đối tượng giao diện. Dựa vào cấu trúc và chức năng, các đối tượng giao diện trong Xamarin.Form được chia thành 4 nhóm: View, Cell, Page và Layout. Sau đây xin giới thiệu một số đối tượng thường dùng.

***+ View***

View là nhóm các đối tượng chính và cơ bản nhất của giao diện như buttons, labels,…

|  |
| --- |
|  |
| Đối tượng | Mô tả | Minh họa |
| Button | Hiển thị nút nhấn trên màn hình | [https://4.bp.blogspot.com/-c5rAM07oDK0/WM_Z2IVB_7I/AAAAAAAAHxA/65FC1YqJpqwu3D8SoZZ5nkK1g8axZooIwCEw/s320/image001.png](https://4.bp.blogspot.com/-c5rAM07oDK0/WM_Z2IVB_7I/AAAAAAAAHxA/65FC1YqJpqwu3D8SoZZ5nkK1g8axZooIwCEw/s1600/image001.png) |
| Entry | Khung nhập liệu một dòng. | [https://1.bp.blogspot.com/-9sa0We5OLGw/WM_Z2LgGffI/AAAAAAAAHw4/63UgO76w3ngjA2sHFZdsk0CumqdwcXnXwCEw/s320/image003.png](https://1.bp.blogspot.com/-9sa0We5OLGw/WM_Z2LgGffI/AAAAAAAAHw4/63UgO76w3ngjA2sHFZdsk0CumqdwcXnXwCEw/s1600/image003.png) |
| Image | Đối tượng dùng để hiển thị hình ảnh | [https://3.bp.blogspot.com/-PW5oDpUFWKo/WM_Z219lLLI/AAAAAAAAHxE/5YwoF9qh0BU-mC8JSDbW5BoqrLK0ffgawCEw/s320/image005.png](https://3.bp.blogspot.com/-PW5oDpUFWKo/WM_Z219lLLI/AAAAAAAAHxE/5YwoF9qh0BU-mC8JSDbW5BoqrLK0ffgawCEw/s1600/image005.png) |
| Label | Hiển thị một dòng hoặc nhiều dòng văn bản chỉ đọc | [https://3.bp.blogspot.com/-zQ8hQqp73yE/WM_Z29sb6gI/AAAAAAAAHxI/PgpI_14mu8kKK1rMxpsiJIbgEQh4_eXhwCEw/s320/image007.png](https://3.bp.blogspot.com/-zQ8hQqp73yE/WM_Z29sb6gI/AAAAAAAAHxI/PgpI_14mu8kKK1rMxpsiJIbgEQh4_eXhwCEw/s1600/image007.png) |
| ListView | Hiển thị các View con dưới dạng danh sách | [https://3.bp.blogspot.com/-7WFRmqVF_LA/WM_Z3_4dCtI/AAAAAAAAHxQ/I-yfpLwfLXItJkAwHnkUtdCdcLgKmakiwCEw/s320/image009.png](https://3.bp.blogspot.com/-7WFRmqVF_LA/WM_Z3_4dCtI/AAAAAAAAHxQ/I-yfpLwfLXItJkAwHnkUtdCdcLgKmakiwCEw/s1600/image009.png) |
| Switch | Đối tượng có lựa chọn tắt/mở | [https://1.bp.blogspot.com/-VIFL5xWlQL4/WM_Z31R0D1I/AAAAAAAAHxU/F38hnw4S6XEzynvRC4TJMkGVDFVsJMjZgCEw/s320/image011.png](https://1.bp.blogspot.com/-VIFL5xWlQL4/WM_Z31R0D1I/AAAAAAAAHxU/F38hnw4S6XEzynvRC4TJMkGVDFVsJMjZgCEw/s1600/image011.png) |
| TableView | Hiển thị các View con dưới dạng bảng | [https://1.bp.blogspot.com/-6WWDXGCfRFQ/WM_Z36mMwqI/AAAAAAAAHxY/pEY2QRA4r9w57C6lKnzJx-ut3XMz92GpQCEw/s320/image013.png](https://1.bp.blogspot.com/-6WWDXGCfRFQ/WM_Z36mMwqI/AAAAAAAAHxY/pEY2QRA4r9w57C6lKnzJx-ut3XMz92GpQCEw/s1600/image013.png) |
| WebView | Trình duyệt Web tích hợp | [https://1.bp.blogspot.com/-0fy2C6hRsCE/WM_Z4hv-WmI/AAAAAAAAHxc/ugct4sK-Tooz5vfhFlQHLvwxliezWTYWwCEw/s320/image015.png](https://1.bp.blogspot.com/-0fy2C6hRsCE/WM_Z4hv-WmI/AAAAAAAAHxc/ugct4sK-Tooz5vfhFlQHLvwxliezWTYWwCEw/s1600/image015.png) |

Bảng 1 Các đối tượng View trong Xamarin Form

***+ Cell***

Cell bao gồm các thành phần được sử dụng trong TableView. Cell không phải là đối tượng được hiển thị mà là bản mẫu quy định cách tạo ra đối tượng. Các Cell trong Xamarin.Form được tổng hợp trong Bảng bên dưới

|  |  |  |
| --- | --- | --- |
| Đối tượng | Mô tả | Minh họa |
| EntryCell | Bao gồm một Label và một Entry. | [https://3.bp.blogspot.com/-i-0UR3tyk90/WM_c7LRfj1I/AAAAAAAAHyA/zGvaqRsONGAiaMH7kMzuPUL2o2Knm_c-gCLcB/s320/image001.png](https://3.bp.blogspot.com/-i-0UR3tyk90/WM_c7LRfj1I/AAAAAAAAHyA/zGvaqRsONGAiaMH7kMzuPUL2o2Knm_c-gCLcB/s1600/image001.png) |
| TextCell | Bao gồm 2 Label, một tiêu đề và một chi tiết | [https://3.bp.blogspot.com/-Js6bKk_ua8c/WM_c7KNHntI/AAAAAAAAHyI/UUYmWNY5bi0UefqliffLNmO3x3uUc2NSwCEw/s320/image003.png](https://3.bp.blogspot.com/-Js6bKk_ua8c/WM_c7KNHntI/AAAAAAAAHyI/UUYmWNY5bi0UefqliffLNmO3x3uUc2NSwCEw/s1600/image003.png) |
| ImageCell | TextCell kèm theo hình ảnh | [https://3.bp.blogspot.com/-FDsgXo7RFpM/WM_c7weytmI/AAAAAAAAHyQ/Fw5CPqxKyfcoeGeKA5wk2mXOQZvtxoeHwCEw/s320/image005.png](https://3.bp.blogspot.com/-FDsgXo7RFpM/WM_c7weytmI/AAAAAAAAHyQ/Fw5CPqxKyfcoeGeKA5wk2mXOQZvtxoeHwCEw/s1600/image005.png) |
| SwitchCell | Bao gồm một Switch và một Label | [https://2.bp.blogspot.com/-TQrRbMZLxVU/WM_c8-XyGpI/AAAAAAAAHyY/Wa9tp7cgNtgwPal-2xBluNruLJabpXhWwCEw/s320/image007.png](https://2.bp.blogspot.com/-TQrRbMZLxVU/WM_c8-XyGpI/AAAAAAAAHyY/Wa9tp7cgNtgwPal-2xBluNruLJabpXhWwCEw/s1600/image007.png) |

Bảng 2 Các đối tượng Cell trong Xamarin Form

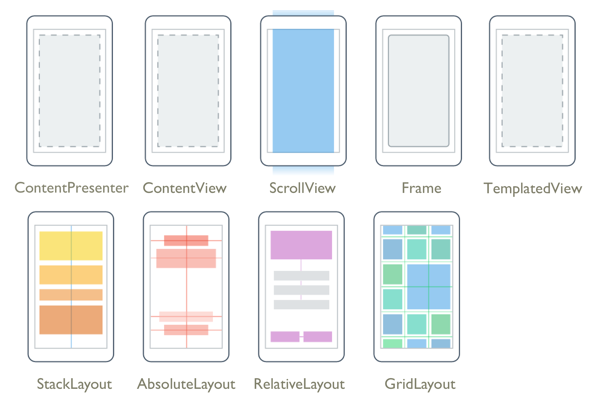
***+ Page***

Page là nhóm các đối tượng chiếm toàn bộ màn hình và chỉ chứa duy nhất một đối tượng con. Page sau khi biên dịch sẽ trở thành View Controller trong iOS, Page trong Windows và giống như Activity trong Android (nhưng không phải Activity). Bảng 3 tổng hợp một số đối tượng Page.

Hình 11 Các Page hỗ trợ trong Xamarin.Form

***+ Layout***

Layout là nhóm các đối tượng đặc biệt, không hiển thị nội dung mà được sử dụng để sắp xếp các View. Một số Layout thường dùng được giới thiệu trong Bảng 4.



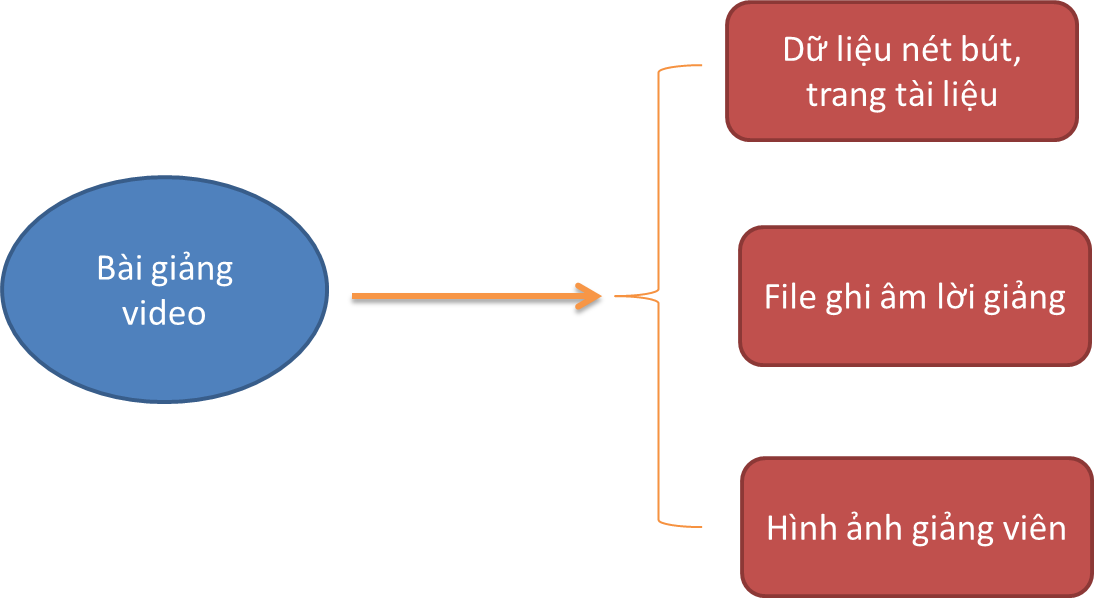
Hình 12 Các layout hỗ trợ trong Xamarin.Form

#### 2.2.6 Mô hình hóa dữ liệu video

Như vấn đề đã nêu, nhược điểm của bài giảng video trong việc học trực tuyến chính là dung lượng lớn (gây khó khăn trong truyền tải, lưu trữ), việc tạo bài giảng cần hỗ trợ bởi những thiết bị ghi hình hay phần mềm chuyên dụng. Ở phần này, tôi xin trình bày việc mô hình hóa dữ liệu video , gúp làm giảm dung lượng lưu trữ, việc tạo ra dữ liệu này cũng đơn giản, không cần hỗ trợ vởi những thiết bị hay phần mềm đặc trưng. Bên cạnh đó, sản phẩm của bài giảng vẫn không mất tính truyền thống, khi mà dữ liệu được hiển thị lại với nội dung tương tự như bài giảng video.

Để thực hiện được ý tưởng này, chúng ta cần cấu trúc lại dữ liệu video như sau :

* Bài giảng nét bút hoặc trang tài liệu hiển thị sẽ được lưu lại dưới dạng XML , quy định thời giảng, tọa độ nét bút
* Dữ liệu âm thanh sẽ được ghi âm lại trong suốt quá trình giảng
* Hình ảnh người thầy được lưu trữ và luôn hiển thị trên bài giảng



Hình 13 Cấu trúc dữ liệu mô tả video

Từ đây, bài giảng video được thay thế bởi bài giảng thuần text và audio . Do đó, dung lượng của bài giảng được giảm đi rất nhiều lần sau khi nén bài giảng text và audio lại.

Hình ảnh bên dưới là dữ liệu của 2 loại bài giảng chính (pdf và chữ viết tay)

|  |  |
| --- | --- |
|  |  |

Bảng 3 Chi tiết dữ liệu mô tả video

Việc giảm dung lượng bài giảng đã có giải pháp, tiếp đây tôi xin giới thiệu công nghệ hiển thị dữ liệu bài giảng này với mục đích chạy đa nền tảng trên ứng dụng di động.

#### 2.2.7 Render dữ liệu chữ viết tay với SkiaSharp

.Net Framework có bộ thư viện System.Drawing rất mạnh trong việc vẽ đường nét. Nhưng thư viện này không hỗ trợ đa nền tảng, nên chúng ta không thể sử dụng nó trong dự án. Thay vào đó, thư viện bên thứ 3 SkiaSharp hỗ trợ tốt việc render đường nét, đồng thời có thể build ứng dụng đa nền tảng.

Skia là 1 thư viện hình ảnh 2D mã nguồn mở cung cấp tập hợp các APIs có thể làm việc trên nhiều nền tảng phần cứng, phần mềm. Đây chính là nền tảng cốt lõi trong việc render hình ảnh của trình duyệt web google chorme và chorme OS, android, Mozila Firefox. Skia được tài trợ và quản lý bởi google nhưng thư viện này có thể được sử dụng cho mọi người dưới cam kết mã nguồn mở BSD. SkiaSharp cung cấp PCL và thư viện dll cho từng nền tảng như sau :

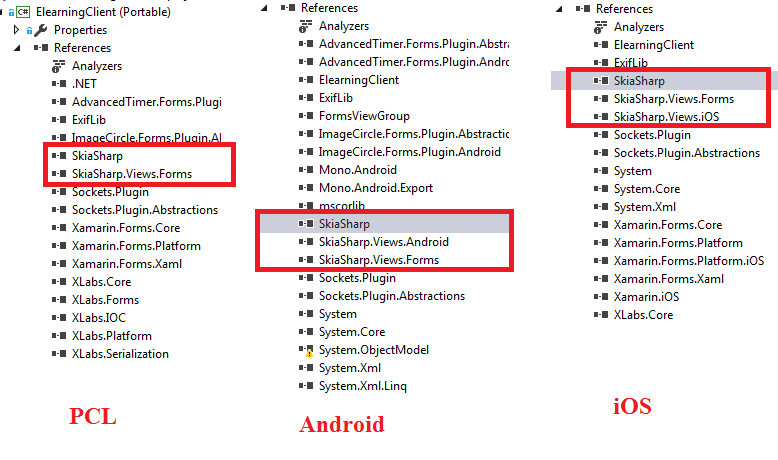
* .NET Core / .NET Standard 1.3
* Xamarin.Android
* Xamarin.iOS
* Xamarin.tvOS
* Xamarin.Mac
* Windows Classic Desktop (Windows.Forms / WPF)
* Windows UWP (Desktop / Mobile / Xbox / HoloLens)

Để sử dụng SkiaSharp, chúng ta cài thư viện qua gói NuGet

|  |
| --- |
| nuget install SkiaSharp |

Bảng 4 Nuget cài đặt SkiaSharp

Sau khi cài đặt NuGet, SkiaSharp PCL và SkiaSharp cho từng nền tảng sẽ tự động được thêm vào project



Hình 14 Render chữ viết tay với SkiaSharp

Để render đường nét sử dụng skia, đơn giản chúng ta chỉ cần ghi đè hàm

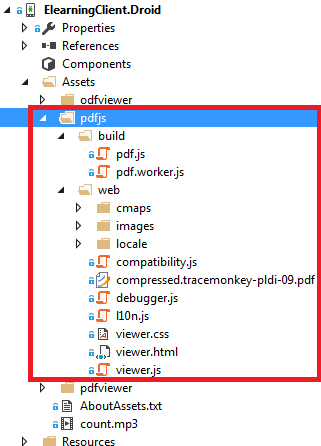
|  |
| --- |
| void OnPainting(object sender, SKPaintSurfaceEventArgs e) |

#### 2.2.8 Hiển thị bài giảng PDF

PDF là định dạng dữ liệu phổ biến nhất hiện nay. Việc thiết kế bài giảng sử dụng tài liệu PDF đã không còn xa lạ. Có nhiều thư viện hỗ trợ hiển thị định dạng dữ liệu PDF trên mỗi nền tảng khác nhau nhưng rất hạn chế thư viện hỗ trợ đa nền tảng. Trước khó khăn đó, tôi lựa chọn phương án hiển thị dữ liệu PDF trên webview (control của xamarin form) cùng với bộ java script phát triển bởi firefox pdf.js.

PDF.js là công cụ hiển thị dữ liệu PDF được build trên HTML5. PDF.js được phát triển bởi Mozila labs với mục tiêu tạo tao ra phương thức chung cho nền tảng dựa web cho việc phân tích và hiển thị PDFs. PDF js hỗ trợ cho cả trình duyệt Firefox và Chorme , có thể nhúng vào các web view của nhà phát triển.

Để sử dụng PDF.js, chúng ta download bộ thư viện trên trang chủ mozila rồi thêm vào project.



Hình 15 Hiển thị dữ liệu PDF với PDFjs

Sau đó, dễ dàng nhúng viewer.html vào webview của ứng dụng trong mỗi nền tảng, từ đó có thể hiển thị tốt tài liệu PDF

Trên đây, tôi vừa giới thiệu cơ sở lý thuyết, nền tảng xamarin và các thành phần cơ bản, quá trình phát triển ứng dụng di động đa nền tảng trên Xamarin.Form. Đồng thời tôi cũng đã đi chi tiết vào công nghệ sử dụng trong project. Phần tiếp, tôi xin trình bày cụ thể việc cài đặt, sử dụng xamarin.form trong ứng dụng hỗ trợ học online cho học sinh , sinh viên.

# CHƯƠNG 3 : XÂY DỰNG ỨNG DỤNG E\_LEARNINGCLIENT

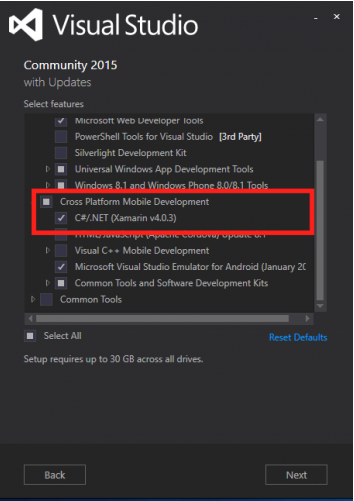
Trong phần này, tôi sẽ trình bày chi tiết về ứng dụng hỗ trợ học sinh, sinh viên trong việc học trực tuyến – ElearningClient. Bao gồm môi trường cài đặt bộ phát triển, các mô hình trong ứng dụng, thiết kế ứng dụng.

## 1. Cài đặt môi trường phát triển

Từ năm 2016, Xamarin chính thức thuộc sở hữu của Microsoft và Xamarin IDE cũng được tích hợp trực tiếp vào visual studio. Để lập trình đa nền tảng với xamarin, người dùng chỉ cần cài đặt visual studio với các yêu cầu như sau :

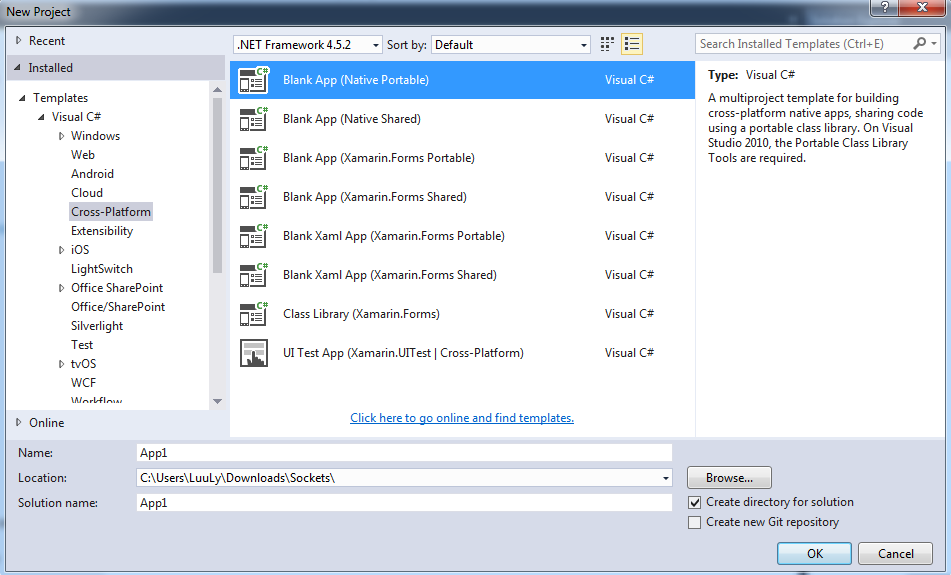
* Xamarin được hỗ trợ ở một trong những phiên bản sau : Visual studio 2012 update 5 trở lên , visual studio 2013 update 3 , visual studio 2015 update 2 , visual studio 2017
* Máy tính windows 8.1 với xung nhịp bộ xử lý tối thiểu 1.6GHz, 1GB RAM, 5GB ổ đĩa, Direcx 9 để hiển thị độ phân giải cao. Để trải nghiệm được mượt mà , tốt nhất nên dùng RAM 3GB trở lên.

Do Xamarin không được cài mặc định khi cài visual studio nên cần chọn thêm xamarin và các tool hỗ trợ đa nền tảng trong quá trình cài đặt.



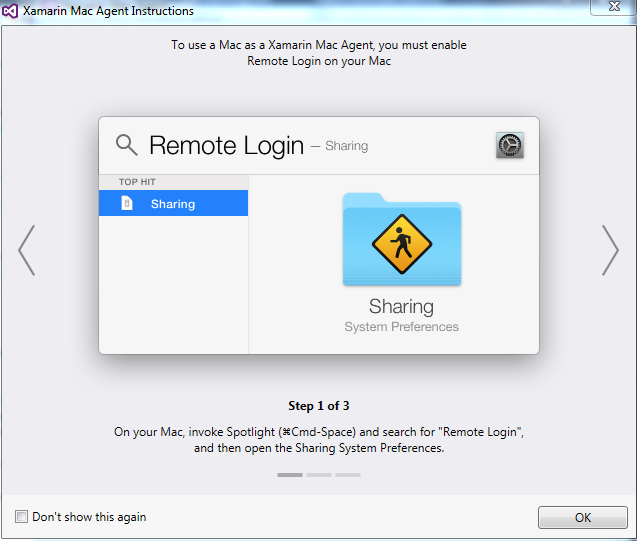
Hình 16 Cài đặt môi trường phát triển xamarin

Tất cả những gói bổ trợ cho việc lập trình đa nền tảng sẽ được cài đặt ở bước này (Android SDK, GWT …). Sau quá trình cài đặt, môi trường phát triển cho ứng dụng di động đa nền tảng đã sẵn sang, người dùng dễ dàng tạo một dự án cho việc phát triển.



Hình 17 Tạo dự án mới với Xamarin

Nếu dùng hệ điều hành windows, project build ra những gói cài đặt được ngày cho Android, Windows phone, windows app. Riêng ứng dụng iOS, để tạo ra gói cài đặt, cần phải build trên máy MAC để tương thích API

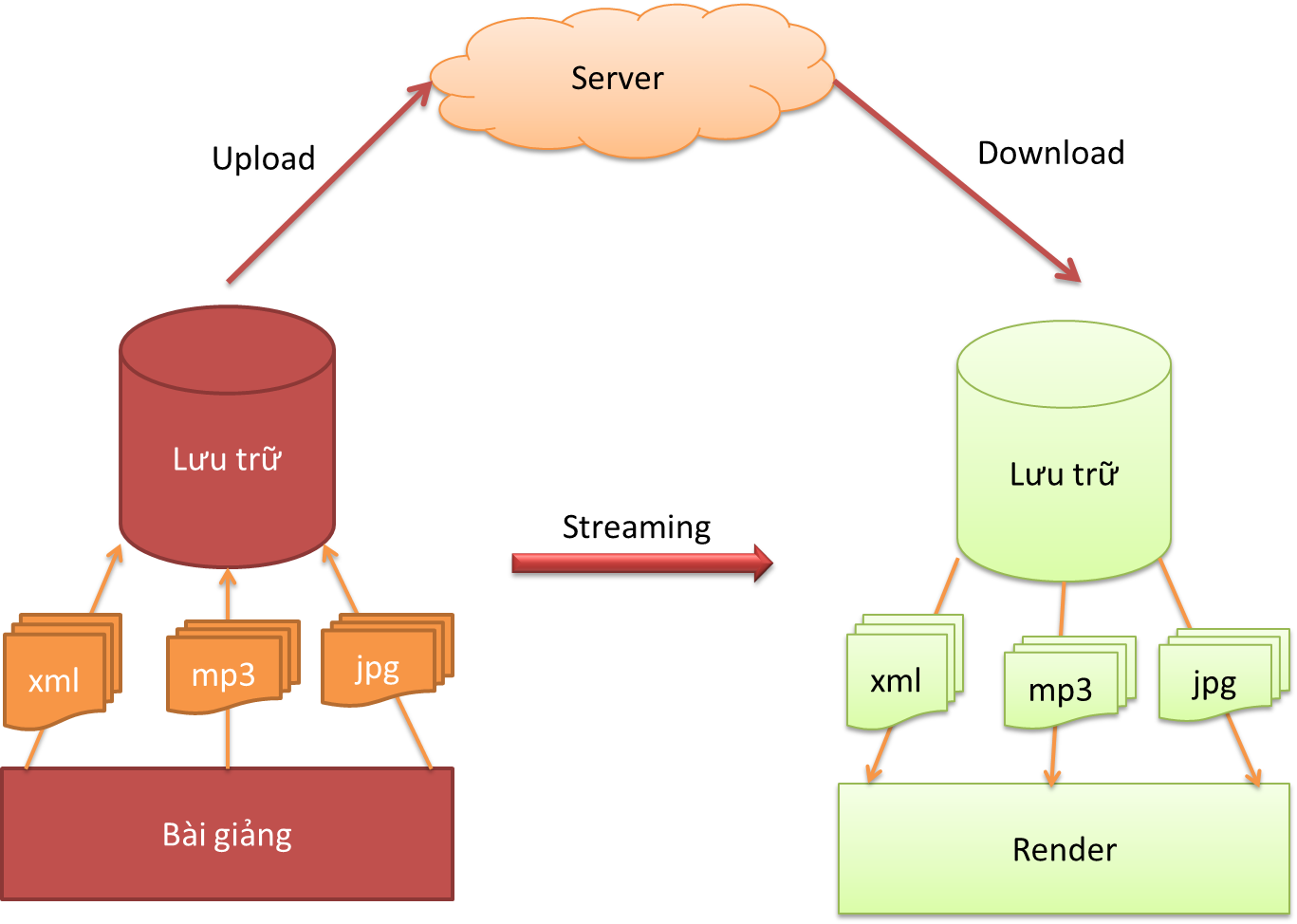


Hình 18 Kết nối với máy MAC

Việc cài đặt môi trường đã hoàn tất, phần tiếp theo, tôi xin trình bày chi tiết về ứng dụng hỗ trợ học trực tuyến cho học sinh, sinh viên (E-Learning Client)

## 2. Ứng dụng E-LearningClient

### 2.1 Kiến trúc hệ thống



Hình 19 Mô hình hệ thống

Hệ thống trên là tổng quan hệ thống E-learning với đầy đủ thành phần người dạy và người học. Việc tạo bài giảng và render bài giảng sẽ được triển khai theo cấu trúc dữ liệu video mới, với đầy đủ thành phần

* Xml : mô tả dữ liệu chữ viết tay hoặc PDF
* Mp3 : mô tả dữ liệu âm thanh được ghi lại trong quá trình giảng bài
* Jpg : mô tả hình ảnh người thầy được hiển thị trên màn hình học viên trong suốt bài giảng

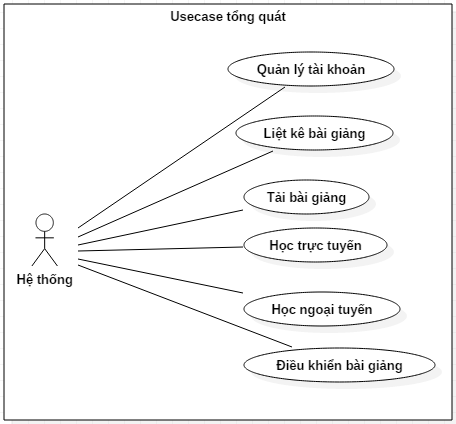
Trong hệ thống này, ứng dụng của chúng ta nằm bên phía client (render) . Đây là ứng dụng hỗ trợ việc học tập trực tuyến cho học sinh , sinh viên với cấu trúc bài giảng mới, chạy đa nền tảng trên android và iOS. Các chức năng của ứng dụng như sau :

* Download , lưu trữ bài giảng
* Render bài giảng viết tay
* Render bài giảng PDF
* Streaming bài giảng trực tiếp từ giảng viên (trong mạng cục bộ)
* Liệt kê, quản lý bài giảng trên server
* Tương tác học viên với bài đánh giá trong quá trình học
* Quản lý đăng nhập học viên

### 2.2. Thiết kế ứng dụng

#### 2.2.1 Các biểu đồ Usecase của ứng dụng

##### 2.2.1.1 Usecase tổng quát

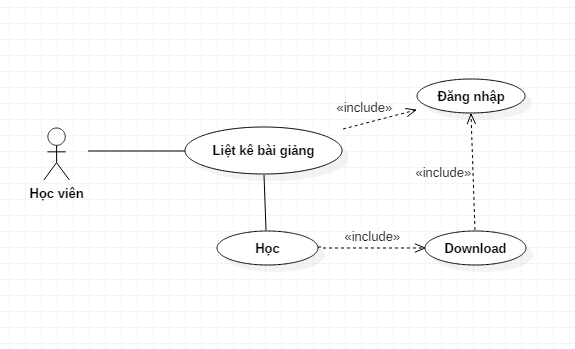
****

Hình 20 Biểu đồ usecase tổng quát

Trong biểu đồ usecase tổng quát, có 2 tác nhân chính : học viên và hệ thống.

* Hệ thống có thể tương tác đến chương trình các hành động như giải mã, giải nén bài giảng, điều khiển bài giảng các chức năng như Play/Pause/Stop như khi đang chơi một video.
* Học viên là tác nhân chính của ứng dụng với các hành động như đăng nhập/đăng xuất, quản lý tài khoản, tải và lưu trữ bài giảng, học (trực tuyến/ngoại tuyến), liệt kê bài giảng đang sẵn có trên server.

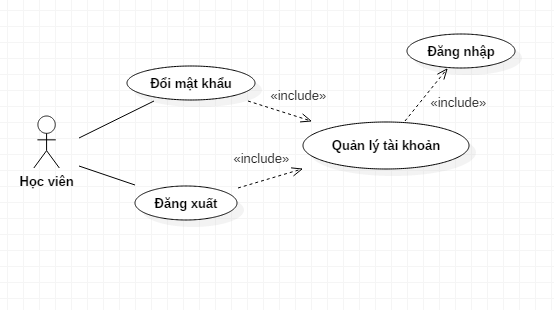
##### 2.2.1.2. Usecase liệt kê bài giảng



Hình 21 Usecase liệt kê bài giảng

Học viên có thể liệt kê bài giảng hệ có trên server bằng cách chọn từ màn hình chính. Từ danh sách bài giảng , học viên có thể chọn để học bài giảng mong muốn. Quá trình liệt kê hoặc học bài chỉ được thực hiện sau khi học viên đã đăng nhập vào hệ thống.

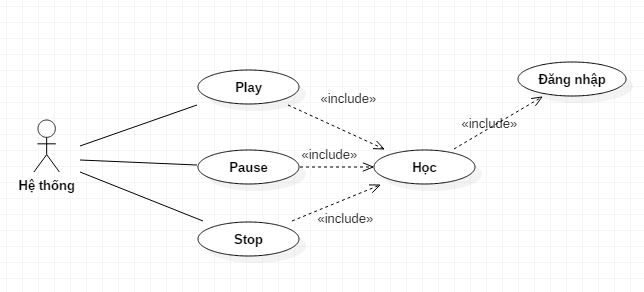
##### 2.2.1.3 Usecase quản lý đăng tài khoản



Hình 22 Usecase quản lý đăng tài khoản

Học viên có thể quản lý tài khoản của mình với những thao các cơ bản như, đăng nhập, đăng xuất, đổi mật khẩu, đổi ảnh đại diện. Ngoài ra, từ giao diện quản lý tài khoản, học viên có thể xem đầy đủ thông tin , trạng thái hiện tại của tài khoản cá nhân

##### 2.2.1.3 Usecase điều khiển bài giảng

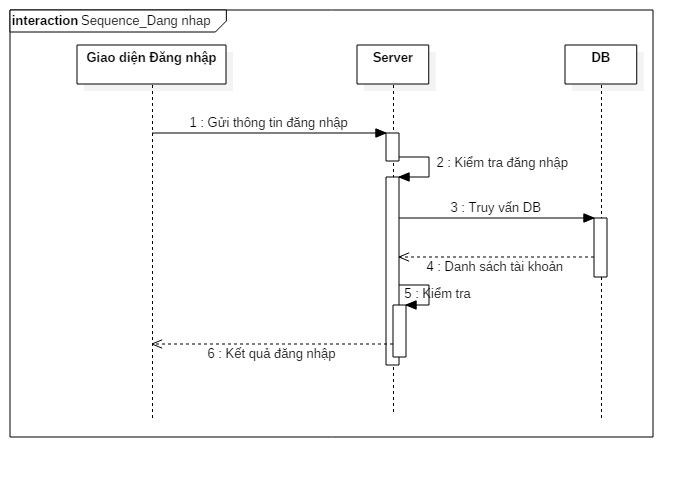
******

Hình 23 Usecase điều khiển bài giảng

Học viên có thể lựa chọn bài giảng để học sau khi đã liệt kê danh sách bài giảng xong. Trong quá trình học, giống như việc học video, bài giảng với cấu trúc dữ liệu mới cũng cung cấp các thao tác cơ bản như Play, Pause, Stop. Qua đó, học viên có thể điều khiển bài giảng của mình một cách dễ dàng.

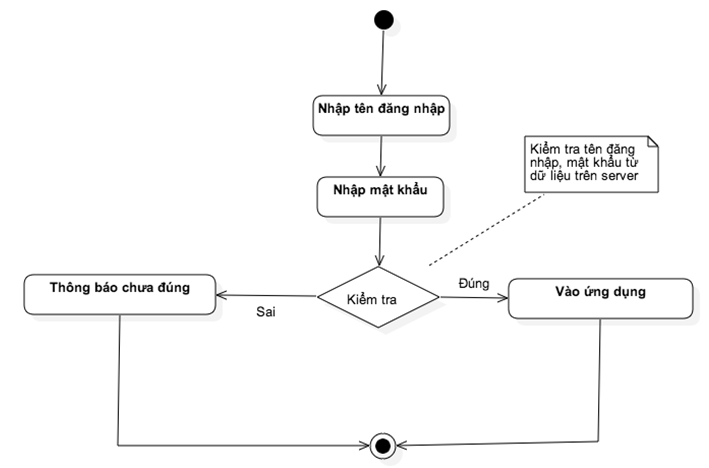
#### 2.2.2. Biểu đồ tuần tự và biểu đồ hoạt động

##### 2.2.2.1 Đăng nhập hệ thống



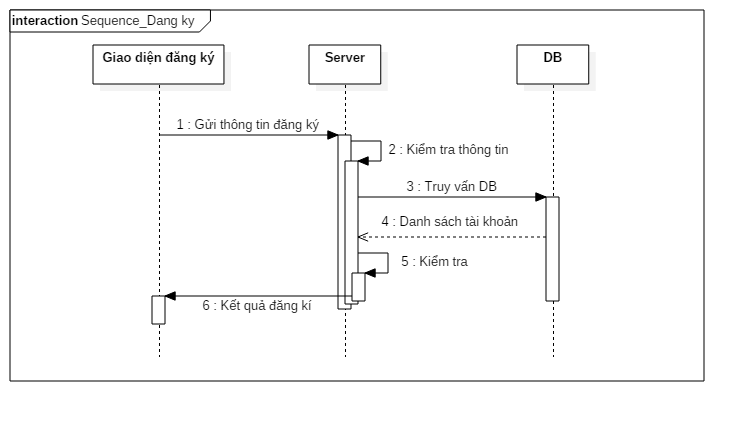
Hình 24 Sequence đăng nhập hệ thống

Người dùng sau khi nhập tên đăng nhập , mật khẩu và nhấn nút “Đăng nhập”, ứng dụng sẽ gửi bản tin kiểm tra tài khoản (kèm thông tin tài khoản) lên server, server sẽ truy vấn cơ sở dữ liệu và kiểm tra tính hợp lệ của tài khoản, sau đó trả về kết quả cho ứng dụng. Sau khi đăng nhập xong, người dùng có thể tiếp tục các hành động khác như liệt kê bài giảng, download bài hay tiến hành học.



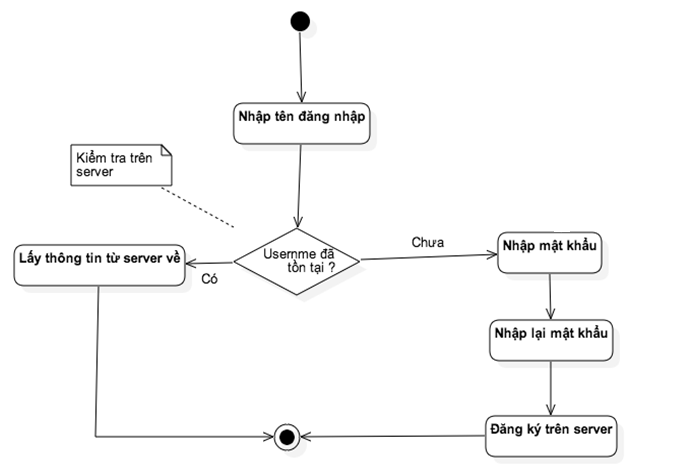
Hình 25 Activity đăng nhập hệ thống

##### 2.2.2.2. Đăng kí tài khoản

******

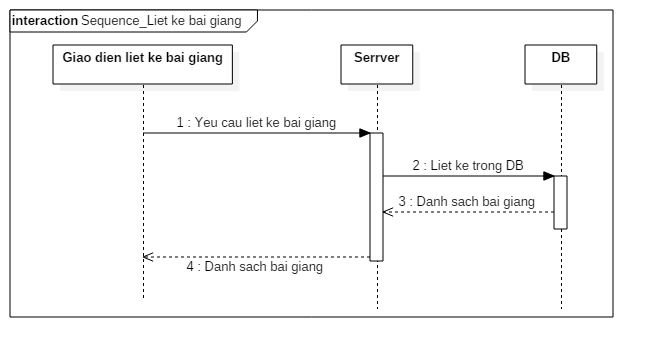
Hình 26 Sequence Đăng kí tài khoản

Người dùng nhập cá thông tin tài khoản : tên tài khoản, mật khẩu, nhắc lại mật khẩu, rồi gửi đăng ký lên server. Server sẽ kiểm tra thông tin đăng ký rồi trả về kết quả đăng ký thành công hay không.

******

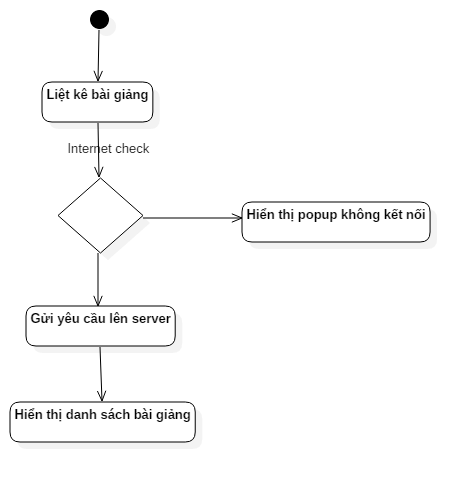
Hình 27 activity Đăng kí tài khoản

##### 2.2.2.3 Liệt kê bài giảng

******

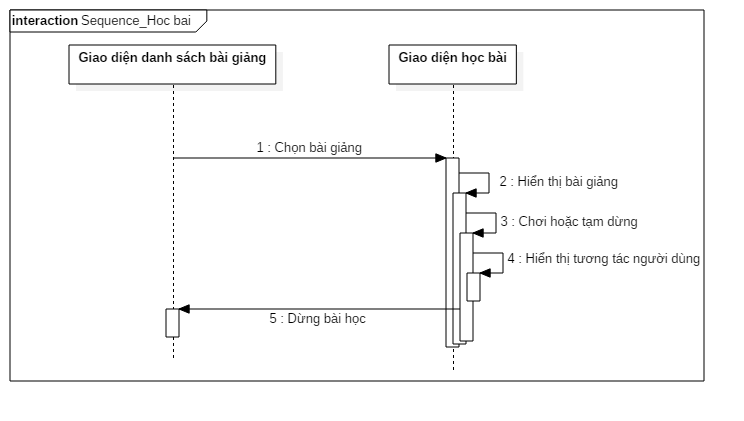
Hình 28 Sequence Liệt kê bài giảng

Sau khi đăng nhập, học viên có thể yêu cầu liệt kê bài giảng hiện có trên server. Yêu cầu liệt kê được gửi lên, server truy vấn cơ sở dữ liệu để lấy ra danh sách bài giảng hiện có và trả về cho học viên, Danh sách này được hiển thị trên màn hình ứng dụng, từ đó học viên có thể chọn bài giảng mong muốn để học.

******

Hình 29 Activity Liệt kê bài giảng

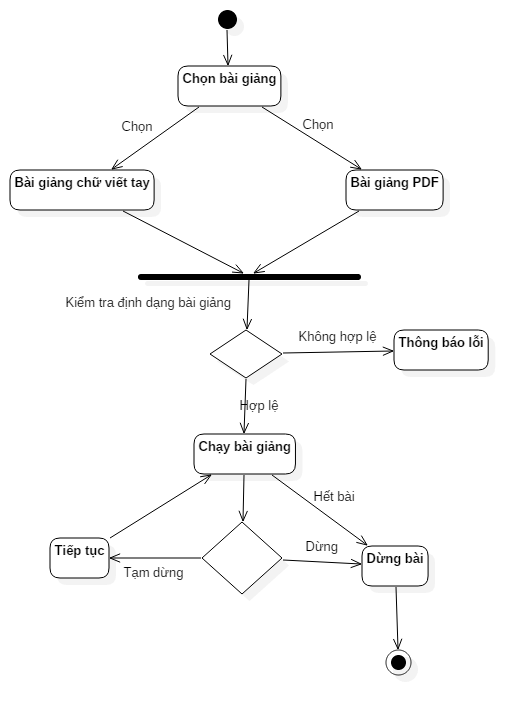
##### 2.2.2.4 Học bài

******

Hình 30 Sequence Học bài

Tại giao diện danh sách bài giảng, học viên có thể chọn bài giảng muốn học để vào giao diện học bài. Tùy thuộc bài giảng là chữ viết tay hay PDF, ứng dụng sẽ chuyển qua màn hình tương ứng. Trên giao diện này, người dùng có thể thao tác với bài giảng như Chơi, tạm dừng/tiếp tục, dừng bài. Ngoài ra, trong quá trình học, tùy vào thiết kế bài giảng của giáo viên, xen giữa bài giảng sẽ có những bài tập tương tác với học viên.

Hình bên dưới là biểu đồ hoạt động của chức năng học bài.

******

Hình 31 Activity Học bài

#### 2.2.3 Thiết kế giao diện

Ứng dụng có những màn hình chính như sau :

+ Giao diện splash

+ Tài khoản

* Giao diện đăng nhập / đăng kí
* Giao diện thông tin tài khoản

+ Danh sách bài giảng (có phân loại bài giảng viết tay và PDF)

+ Download bài giảng

+ Giao diện học bài

+ Học ngoại tuyến

* Bài giảng viết tay
* Bài giảng PDF

+ Học trực tuyến

Màu sắc chủ đạo là màu xanh, tím phù hợp với nhiều lứa tuổi, đối tượng học viên. Các hình bên dưới là một số giao diện ứng dụng.

|  |  |
| --- | --- |
| Giao diện chào mừng | Thông tin đăng nhập |

|  |  |
| --- | --- |
| Danh sách bài giảng |  |

|  |  |
| --- | --- |
| Bài giảng PDF | Bài giảng viết tay |

### 2.3 Một số kịch bản kiểm thử và kết quả

#### 2.3.1. Yêu cầu kiểm thử

|  |  |  |  |
| --- | --- | --- | --- |
| STT | Yêu cầu kiểm thử | Yêu cầu kết quả | Kết quả |
| 1 | Hiển thị màn hinh Splash | Có animation chạy cùng | PASS |
| 2 | Đăng ký tài khoản | Đăng kí được và hiển thị được các lỗi nếu xảy ra | PASS |
| 3 | Đăng nhập | Đăng nhập thành công, hiển thị được các lỗi nếu xảy ra, chuyển sang màn hình chính | PASS |
| 4 | Liệt kê bài giảng | Liệt kê đủ bài giảng trên server | PASS |
| 5 | Tải bài giảng | Tải được bài giảng về , lưu trữ trên bộ nhớ vật lý | PASS |
| 6 | Học bài giảng viết tay | Bài giảng viết tay hiển thị đều nét, nét bút và âm thanh phải đồng bộ nhau | PASS |
| 7 | Hình ảnh người thầy | Hình ảnh người thầy phải được hiển thị trong suốt quá trình học | PASS |
| 8 | Học bài giảng PDF | Bài giảng PDF cần hiển thị đúng thứ tự trang, âm thanh phải được đồng bộ với bài giảng | PASS |
| 9 | Học trực tuyến | Việc học trực tuyến diễn ra thống suốt , đường truyền ổn định, lag ít | Cần cải tiến |

Bảng 5 Các yêu cầu kiểm thử

#### 2.3.2 Tình huống kiểm thử

|  |  |  |  |
| --- | --- | --- | --- |
| STT | Tình huống | Yêu cầu kết quả | Kết quả |
| 1 | Đăng ký tài khoản đã tồn tại | Thông báo trùng | PASS |
| 2 | Liệt kê bài giảng trong điều kiện tốt | Danh sách bài giảng hiển thị đầy đủ | PASS |
| 3 | Đăng nhập khi mất kết nối mạng | Thông báo không có mạng | PASS |
| 4 | Học ngoại tuyến với bài giảng không xác định | Thông báo bài giảng không hợp lệ | PASS |

Bảng 6 Các tình huống kiểm thử

Sau kịch bản kiểm thử diễn ra, hầu hết các chức năng và thành phần giao diện đều hoạt động tốt, ngoại trừ chức năng học online thời gian thực cần cải tiến thêm. Trong phiên bản sau, tôi sẽ đề ra phương án nâng cao hiệu quả hoạt động của ứng dụng cũng như cải tiến việc học thời gian thực. Như vậy, ứng dụng đã được thiết kế và cài đặt, kết quả kiểm nghiệm trên thiết bị thật, ứng dụng đã vượt qua được các kịch bản kiểm thử. Phần tiếp theo, tôi xin nêu ra kết luận, kiến thức công nghệ nghiên cứu được và hướng phát triển tiếp theo.

# CHƯƠNG 4 : KẾT LUẬN VÀ HƯỚNG PHÁT TRIỂN

Trong phần này, tôi sẽ tổng kết lại những kết quả mà đề tài đã đạt được đồng thời nêu ra hướng phát triển tiếp theo. Theo quan điểm chủ quan, đề tài đã đạt được những yêu cầu đặt ra, tháo gỡ và khắc phục được những vấn đề hiện nay của việc học trực tuyến phía học sinh, sinh viên. Ứng dụng đã được cài đặt và kiểm thử trên thiết bị thật và hoạt động nhưng ý muốn.

## 4.1 Kết quả đạt được

Nắm được ưu, nhược điểm và quá trình phát triển ứng dụng di động đa nền tảng trên Xamarin

* Hiểu được cách render đường nét trên nhiều nền tảng
* Hiểu được cách hiển thị tài liệu PDF đa nền tảng
* Làm việc được với giao tiếp mạng socket đa nền tảng
* Củng cố kỹ năng phân tích, thiết kế hệ thống

## 4.2 Định hướng phát triển

Đề tài là một nghiên cứu khoa học nhăm đưa được công nghệ mới vào ứng dụng thực tiễn. Tuy nhiên, việc phát triển ứng dụng trong thời gian ngắn nên không thể thiếu những thiếu sót và những thành phần cần phát triển thêm. Tôi xin liệt kê những việc cần cải tiến như sau :

* Giao diện cần được tối ưu, hiển thị mượt mà
* Cần độc lập hóa dữ liệu mô tả video, để đây là dữ liệu phát triển nói chung hỗ trợ học trực tuyến
* Việc thiết lập môi trường học trực tuyến thời gian thực cần thực hiện như một giao thức với đầu vào là dữ liệu đã được định nghĩa. Dữ liệu đầu vào có thể tạo ra bởi nhiều nguồn, nhiều nền tảng khác nhau
* Hỗ trợ được việc học trực tuyến trong mạng internet

Đây là ứng dụng có tính thực tiễn cao, cần được nghiên cứu phát triển thêm để có thể đi vào thực tế

***TÀI LIỆU THAM KHẢO***

*[1]* [*http://www.vjol.info/index.php/sphcm/article/viewFile/12799/11687*](http://www.vjol.info/index.php/sphcm/article/viewFile/12799/11687)

*[2]JamesMontemagno\_XamarinFormsNativeIOSAndroidAndWindowsPhoneApps FromONECCodebase.pdf*

*[3]MICROSOFT\_PRESS\_EBOOK\_CREATINGMOBILEAPPS WITHXAMARINFORMS\_PDF.PDF*

*[4]*[*http://minhtriettran.blogspot.com/2016/04/xamarin-crossplatform-development\_22.html*](http://minhtriettran.blogspot.com/2016/04/xamarin-crossplatform-development_22.html)

*[5]* [*https://skia.org/*](https://skia.org/)

*[6]* [*https://mozilla.github.io/pdf.js/*](https://mozilla.github.io/pdf.js/)

*[7]* [*https://developer.xamarin.com*](https://developer.xamarin.com)

*[8] https://developer.xamarin.com/guides/xamarin-forms*