

**BỘ GIÁO DỤC VÀ ĐÀO TẠO  
TRƯỜNG ĐẠI HỌC BÁCH KHOA HÀ NỘI**

-----

**PHÙNG NGỌC VŨNG**

**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 VŨNG**

**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 VŨNG**

## 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 tượng 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 hỗ trợ học sinh học trực tuyến 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 giảng dạy hết sức 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 hết sức, nhưng bài luận vẫn chắc chắn 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 Nuget cài đặt SkiaSharp .....	40
Bảng 2 Chi tiết dữ liệu mô tả video.....	48
Bảng 3 Cấu trúc dữ liệu bài trắc nghiệm tương tác.....	49
Bảng 4 Các yêu cầu kiểm thử.....	65
Bảng 5 Các tình huống kiểm thử .....	66

## DANH MỤC KÝ HIỆU VÀ TỪ VIẾT TẮT

<b>Ký hiệu</b>	<b>Từ Viết Tắt</b>	<b>Nghĩa</b>
<b>E-Learning</b>	Electronics Learning	Học trực tuyến
<b>SDK</b>	Software Development Kit	Bộ công cụ phát triển phần mềm
<b>OS</b>	Operating system	Hệ điều hành
<b>XML</b>	eXtensible Markup Language	Ngôn ngữ đánh dấu mở rộng
<b>PCL</b>	Portable Class Library	Lớp thư viện khả chuyển
<b>MVVM</b>	Model – View – ViewModel	Mô hình phát triển phần mềm
<b>PDF</b>	Portable Document Format	Định dạng dữ liệu khả chuyển
<b>CD</b>	Compact Disk	Đĩa quang
<b>SQLITE</b>		Chương trình quản lý cơ sở dữ liệu
<b>HTML</b>	HyperText Markup Language	Ngôn ngữ đánh dấu siêu văn bản
<b>WPF</b>	Windows Presentation Foundation	Công nghệ xây dựng ứng dụng trên máy tính
<b>DLL</b>	Dynamic link library	Thư viện liên kết động
<b>LMS</b>	Learning Management System	Hệ quản trị đào tạo

## DANH MỤC HÌNH ẢNH

Hình 1	Cấu trúc khối hệ thống Moddle .....	13
Hình 2	Giao diện hệ thống Sakai 11 .....	14
Hình 3	Hệ thống Dokeos .....	15
Hình 4	Apache Cordova .....	20
Hình 5	So sánh công nghệ phát triển ứng dụng đa nền tảng .....	22
Hình 6	Công nghệ hiển thị chữ viết tay.....	23
Hình 7	Công nghệ hiển thị dữ liệu PDF .....	24
Hình 8	Quá trình hình thành và phát triển của Xamarin .....	25
Hình 9	Kiến trúc tầng của xamarin.....	27
Hình 10	Kiến trúc liên kết của Xamarin.....	27
Hình 11	Các nền tảng hỗ trợ bởi Xamarin .....	28
Hình 12	Chia sẻ mã nguồn trong Xamarin.Form .....	31
Hình 13	Chia sẻ mã nguồn trong từng nền tảng.....	31
Hình 14	Kết quả ứng dụng tạo bằng Xamarin.Form .....	32
Hình 15	Hoạt động chia sẻ mã nguồn của Shared Project .....	33
Hình 16	Hoạt động chia sẻ mã nguồn của PCL.....	35
Hình 17	Mô hình MVVM.....	38
Hình 18	Render chữ viết tay với SkiaSharp .....	41
Hình 19	Hiển thị dữ liệu PDF với PDFjs .....	42
Hình 20	Mô hình hệ thống.....	44
Hình 21	Cấu trúc dữ liệu mô tả video .....	46
Hình 22	Cài đặt môi trường phát triển xamarin .....	50
Hình 23	Tạo dự án mới với Xamarin .....	51
Hình 24	Biểu đồ usecase tổng quát .....	53
Hình 25	Usecase liệt kê bài giảng .....	54
Hình 26	Usecase quản lý đăng tải khoản.....	55
Hình 27	Usecase điều khiển bài giảng .....	55
Hình 28	Sequence đăng nhập hệ thống .....	56
Hình 29	Activity đăng nhập hệ thống.....	57
Hình 30	Sequence Đăng kí tài khoản .....	57
Hình 31	Activity Đăng kí tài khoản .....	58
Hình 32	Sequence Liệt kê bài giảng.....	58
Hình 33	Activity Liệt kê bài giảng .....	59
Hình 34	Sequence Học bài .....	59
Hình 35	Activity Học ngoại tuyến.....	60



## MỤC LỤC

LỜI CAM ĐOAN.....	1
LỜI MỞ ĐẦU .....	2
LỜI CẢM ƠN.....	3
DANH MỤC BẢNG BIỂU .....	4
DANH MỤC KÝ HIỆU VÀ TỪ VIẾT TẮT .....	5
DANH MỤC HÌNH ẢNH.....	6
CHƯƠNG 1: ĐẶT VẤN ĐỀ .....	9
1.1. Giới thiệu việc giáo dục trực tuyến.....	9
1.1.1 Các loại hình đào tạo trực tuyến .....	9
1.1.2. Lợi ích học trực tuyến.....	10
1.2. Các vấn đề của việc học trực tuyến hiện nay .....	11
1.2.1. Vấn đề phía người học .....	12
1.2.2. Vấn đề phía nội dung học tập .....	12
1.2.3. Vấn đề về yếu tố công nghệ.....	12
1.3. Các hệ thống học trực tuyến hiện nay .....	12
1.3.1. Moddle .....	13
1.3.2. Sakai 11.....	14
1.3.3. Dokeos .....	15
CHƯƠNG 2: GIẢI PHÁP KHẮC PHỤC VẤN ĐỀ VÀ ĐỊNH HƯỚNG CÔNG NGHỆ	17
2.1. Giải pháp .....	17
2.2. Phương pháp tiếp cận và định hướng công nghệ .....	18
2.2.1. Phương pháp tiếp cận.....	18
2.2.2. Công nghệ thực hiện .....	19
2.2.2.1. Nền tảng Xamarin .....	25
2.2.2.2. Những ưu điểm, nhược điểm của Xamarin .....	27
2.2.2.2.1. Ưu điểm.....	27
2.2.2.2.2. Nhược điểm.....	29
2.2.2.3. Phát triển ứng dụng di động đa nền tảng dựa trên Xamarin Form.....	30

2.2.2.3.1. Shared Project .....	32
2.2.2.3.2. Portable Class Libraries (PCL) .....	34
2.2.2.4. Mô hình MVVM .....	38
2.2.2.5. Hiển thị dữ liệu chữ viết tay với SkiaSharp .....	39
2.2.2.6. Hiển thị bài giảng PDF với PDFjs.....	41
CHƯƠNG 3: XÂY DỰNG ỨNG DỤNG HỖ TRỢ HỌC TRỰC TUYẾN - ELEARNINGCLIENT .....	44
3.1. Kiến trúc hệ thống.....	44
3.2. Thiết kế ứng dụng E-LearningClient .....	45
3.2.1. Mô hình hóa dữ liệu video .....	45
3.2.2. Cài đặt môi trường phát triển.....	50
3.2.3. Thiết kế ứng dụng .....	52
3.2.3.1 Các biểu đồ Usecase của ứng dụng .....	53
3.2.3.1.1 Usecase tổng quát .....	53
3.2.3.1.2. Usecase liệt kê bài giảng.....	54
3.2.3.1.3. Usecase quản lý đăng tài khoản .....	55
3.2.3.1.4. Usecase điều khiển bài giảng.....	55
3.2.3.2. Biểu đồ tuần tự và biểu đồ hoạt động.....	56
3.2.3.2.1. Đăng nhập hệ thống .....	56
3.2.3.2.2. Đăng kí tài khoản .....	57
3.2.3.2.3. Liệt kê bài giảng.....	58
3.2.3.2.4. Học bài .....	59
3.2.3.3. Thiết kế giao diện .....	61
3.2.4. Một số kịch bản kiểm thử và kết quả.....	64
3.2.4.1. Yêu cầu kiểm thử .....	64
3.2.4.2. Tình huống kiểm thử .....	66
CHƯƠNG 4: KẾT LUẬN VÀ HƯỚNG PHÁT TRIỂN .....	67
4.1. Kết quả đạt được .....	67
4.2. Định hướng phát triển .....	67

## CHƯƠNG 1: ĐẶT VẤN ĐỀ

### 1.1. 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, sự 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...

#### 1.1.1. 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 lý 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.

### **1.1.2. 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 lý 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 ra 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)...

## **1.2. 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. [1]

### **1.2.1. 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.

### **1.2.2. 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.

### **1.2.3. 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 đáng kể tới tiến độ, chất lượng học tập.

## **1.3. Các hệ thống học trực tuyến hiện nay**

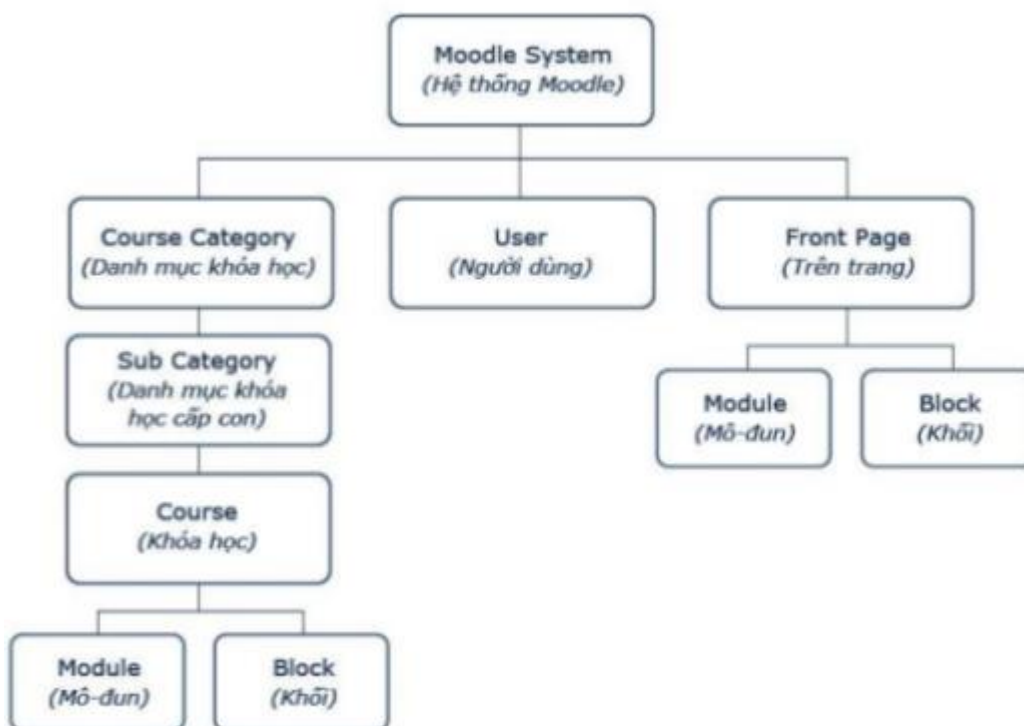
Hệ thống quản lý học tập nói chung (LMS) là hệ thống quản lý các quá trình học tập và phân phát nội dung khoá học tới người học. LMS bao gồm nhiều mô-đun khác nhau giúp quá trình học tập trên mạng được thuận tiện và dễ dàng phát huy hết các điểm mạnh của internet. Hiện nay, có rất nhiều hệ thống LMS được triển khai với nhiều mục đích, ưu nhược điểm khác nhau, tiếp theo tôi xin trình bày một số hệ thống LMS phổ biến và vấn đề tồn tại của những hệ thống này.

### 1.3.1. Moddle

Là hệ thống quản lý học tập mã nguồn mở cho phép tạo các khóa học trên mạng internet hay các website học tập trực tuyến [2]. Moddle bao gồm hầu hết các tính năng cơ bản của một hệ thống LMS như :

- Tạo lập và quản lý khóa học
- Đưa nội dung bài giảng tới người học
- Trợ giúp giáo viên tổ chức các hoạt động nhằm quản lý khóa học như trao đổi, đánh giá, thảo luận, kiểm tra cuối khóa, bài tập lớn...
- Quản lý học viên
- Quản lý tài nguyên của từng khóa học
- Báo cáo tiến trình học, tương tác của học viên

Hình ảnh bên dưới là cấu trúc khối của hệ thống Moddle



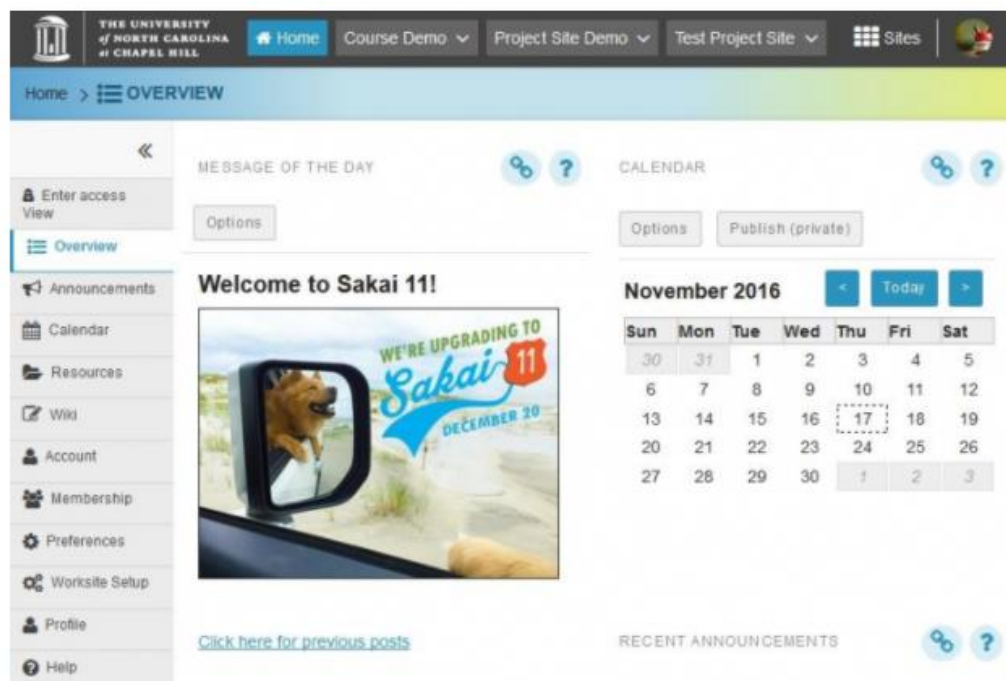
**Hình 1 Cấu trúc khối hệ thống Moddle**

Hệ thống moddle hướng tới mọi đối tượng tham gia vào hệ thống LMS như giáo viên, học sinh, nhà quản lý... Tuy hệ thống đang được sử dụng rộng rãi và có rất nhiều ưu việt trong việc tạo, quản lý bài giảng trực tuyến nhưng moddle cũng đang gặp một số hạn chế như sau :

- Là hệ thống bao gồm tất cả, hướng mọi đối tượng nên việc cài đặt , tiếp cận còn phức tạp đối với đối với một số đối tượng không thành thạo công nghệ
- Được biết đến như một forum chia sẻ tài liệu tĩnh, không đề cập đến bài giảng video trực tuyến hay học thời gian thực
- Chú trọng phát triển website hơn là ứng dụng. Tuy có hỗ trợ cho thiết bị di động nhưng ứng dụng được phát triển trên các nền tảng riêng biệt.

### 1.3.2. Sakai 11

Sakai 11, giống như Moddle, là một giải pháp LMS mã nguồn mở hỗ trợ hầu hết các chức năng cơ bản của một hệ thống LMS. Điểm khác của Sakai 11 là được phát triển trên Java. Sakai 11 có giao diện dễ sử dụng hơn, tăng tối ưu cho việc sử dụng trên thiết bị di động, hỗ trợ tốt việc giao bài tập đề nghị cho học sinh và thiết kế bài giảng.



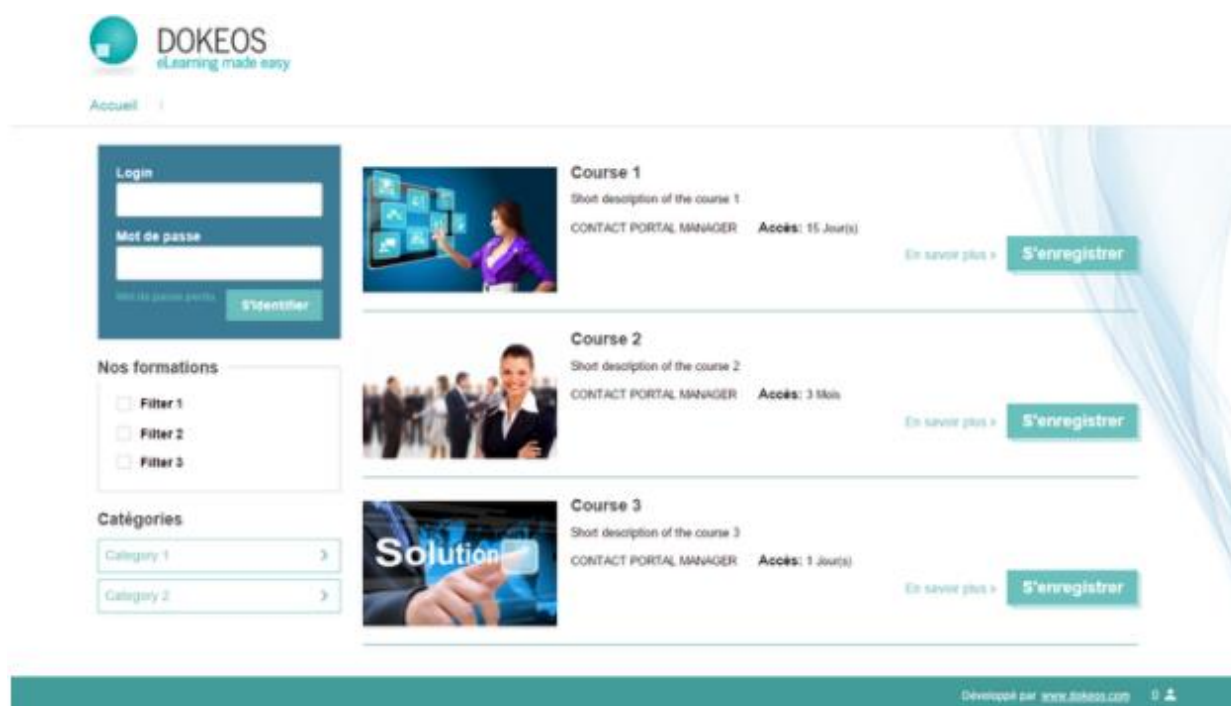
**Hình 2** Giao diện hệ thống Sakai 11



Sakai được tích hợp với Google Gốc và bao gồm các công cụ hỗ trợ tra cứu, kiểm thử online, trình chiếu slide và cho phép sử dụng Dropbox. Tuy nhiên, Do việc Sakai ra đời với định hướng đến các dự án giáo dục lớn, có chiến lược phát triển theo đa hướng (website, ứng dụng) nên hạn chế đối tượng sử dụng. Cộng đồng phát triển của Sakai cũng khá hẹp, không nhiều plug-in hay add-ons hỗ trợ.

### 1.3.3. Dokeos

Dokeos là một giải pháp LMS mã nguồn mở khác được build trên PHP. Công cụ này được phát triển tại Pháp và được sử dụng trên 60 quốc gia khác nhau.



**Hình 3 Hệ thống Dokeos**

Dokeos xây dựng như một công cụ đăng ký khóa học, thiết kế các mẫu bài thi, học nhóm riêng rẽ và hỗ trợ chat giữa các thành viên. Dokeos cũng hỗ trợ trình chiếu các loại tài liệu cơ bản như word office và power point. Tuy nhiên, Dokeos gặp vấn đề về phân chia cấp độ người học, quản lý học viên, chưa hỗ trợ việc hỏi đáp như một forum. Ngoài ra, bài giảng video được tải lên với định dạng gốc nên tốc độ truyền tải còn khá chậm.

*Tóm lại*, qua việc trình bày các vấn đề của việc học trực tuyến cũng như các vấn đề của một số hệ thống học trực tuyến hiện hành, tôi xin nêu ra các chi tiết các khó khăn mà đề tài cần giải quyết như sau :

*Thứ nhất*, cần hỗ trợ tốt bài giảng video (chữ viết tay và tài liệu) từ việc tạo bài giảng đến việc truyền tải, hiển thị, giảm thiểu chi phí tạo bài giảng. 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. Do đó gặp phải những vấn đề :

- 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.

*Thứ hai*, bài giảng video đơn thuần chỉ 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. Với nội dung bài giảng liên tục, nên dễ gây ra sự nhàm chán, bài giảng không có tính tương tác với học sinh.

*Thứ ba*, Các hệ thống hiện hành chủ yếu được xây dựng trên website. Trong khi đó, ngày càng nhiều thiết bị, nền tảng được học sinh, sinh viên sử dụng. Vì thế, cần có một giải pháp hỗ trợ việc học tập cho học sinh trên đại đa số thiết bị.

Theo những phân tích về hạn chế của việc tạo bài giảng video, trình chiếu bài giảng cùng các vấn đề của hệ thống học trực tuyến hiện nay, 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

### 2.1. Giải pháp

Theo vấn đề nêu trên (*Chương 1*), 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 tải; 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 gắn 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 ra 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 nơi có thể nhận, phát thông báo cho toàn thể học viên trong cùng lớp hoặc xin ý kiến riêng người thầy
- ‘R2Tạo ra một nơi thảo luận, hỏi đáp như một forum

## **2.2. Phương pháp tiếp cận và định hướng công nghệ**

### **2.2.1. 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 mô hình hóa lại dữ liệu video theo 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 này sẽ được ghi cùng với tọa độ nét bút, trang tài liệu người thầy sử dụng và thời gian tương tác với màn hình.
- Dữ liệu âm thanh: ghi lại quá trình giảng bài của người giáo viên với định dạng mp3. Dữ liệu âm thanh cũng ghi lại cùng với trục thời gian giảng dạy, qua đó giúp cho việc đồng bộ dữ liệu giữa dữ liệu hiển thị và dữ liệu âm thanh.
- Hình ảnh người giáo viên: hiển thị xuyên suốt bài giảng, nhằm không làm mất đi sự gắn 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:*

Hiện nay có nhiều bộ thư viện, KIT hỗ trợ tốt cho lập trình viên phát triển ứng dụng di động đa nền tảng như: PhoneGap, AppCelerator, Xamarin, React Native, Sencha Touch... Mỗi thư viện phát triển đều có những ưu điểm và nhược điểm riêng. Trong khuôn khổ luận văn, tôi sẽ nghiên cứu và sử dụng công nghệ phát triển với mục đích tạo ra ứng dụng đa nền tảng (android, iOS), chia sẻ tốt mã nguồn, hỗ trợ thương mại hóa ứng dụng, công cụ phát triển thân thiện, tiệm cận nhất với ứng dụng native, hỗ trợ kiểm thử.

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.

### **2.2.2. Công nghệ thực hiện**

#### ***Công nghệ phát triển ứng dụng di động đa nền tảng [3]***

Với sự đa dạng của thiết bị di động và sự khác biệt của mỗi nền tảng, ngày nay các nhà phát triển ứng dụng gặp nhiều khó khăn để cho ra các ứng dụng có tính thương mại cao vì người dùng phân bố ở nhiều platform khác nhau. Trong ngành giáo dục trực tuyến cũng vậy, học sinh và phụ huynh có thể sử dụng các thiết bị và nền tảng bất kỳ. Do đó, việc tạo ra ứng dụng hỗ trợ được đại đa số người dùng đang trở nên rất cấp thiết. Để khắc phục việc phải tạo mã nguồn nhiều lần trên từng nền tảng, hiện nay có nhiều công nghệ phát triển ứng dụng đa nền tảng với những ưu nhược điểm nhất định.

*Apache Cordova:* Tiền thân là PhoneGap, là một framework nổi tiếng của Adobe System. Tư tưởng của Cordova là tạo ra ứng dụng di động lai (hybrid mobile application) sử dụng HTML , CSS và Javascript. Những ứng dụng này có thể chạy trên nhiều nền tảng khác nhau như Android, iOS, Windows phone ... dựa trên các API để giao tiếp với thiết bị.



**Hình 4 Apache Cordova**

Ưu điểm nổi bật của Cordova là thời gian phát triển ứng dụng nhanh hơn rất nhiều so với việc phát triển từng ứng dụng native, chỉ cần sử dụng JavaScript nên không cần phải biết từng ngôn ngữ lập trình cho mỗi hệ điều hành.

Tuy nhiên, điểm trừ lớn nhất của công nghệ này lại là tốc độ đáp ứng, hiệu năng của ứng dụng. Hiệu suất của các hybrid app chậm hơn nhiều so với native app nên không thể dùng Cordova để xây dựng ứng dụng lớn, phức tạp yêu cầu nhiều dữ liệu và chức năng. Ngoài ra, các plugin của Cordova có thể không tương thích được với một số thiết bị và nền tảng, một số APIs cũng chưa được hỗ trợ giao tiếp với thiết bị.

*React native*: là công nghệ được tạo bởi Facebook, cho phép các nhà phát triển sử dụng JavaScript để làm ứng dụng di động trên cả Android và iOS với cảm nhận và giao diện native. Một ví dụ để làm rõ cách vận hành của React Native là wrapper của code native. Có rất nhiều thành tố được tạo nên khi đang wrapping – “bọc” chức năng native của iOS hoặc Android. React Native đã đạt được rất nhiều traction kể từ ngày ra mắt vì nó đã thay đổi game về mặt cơ bản theo nhiều cách.

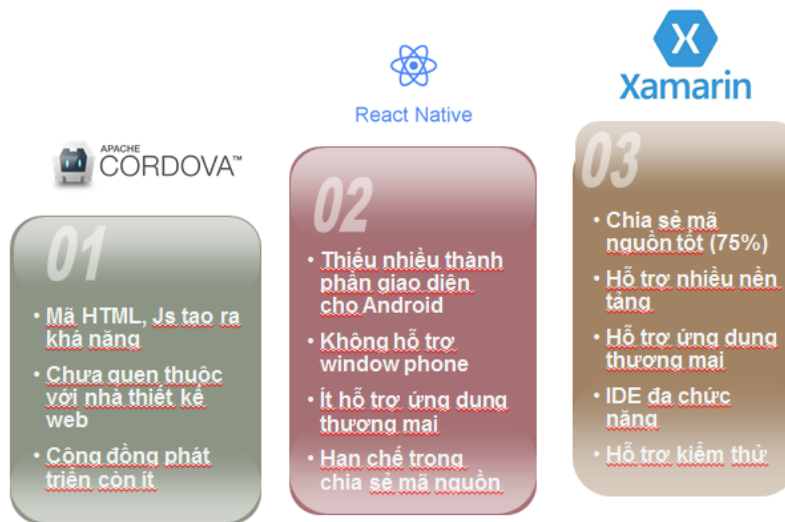
Phát triển ứng dụng di động sử dụng React native chỉ cần dùng đến ngôn ngữ Javascript, sau đó trình biên dịch sẽ dịch mã JavaScript thành mã native của từng nền tảng, qua đó giúp ứng dụng có thể chạy mượt mà hơn.

Ưu điểm của React native là việc phát triển ứng dụng tiệm cận native chỉ cần dùng đến một ngôn ngữ, không cần quan tâm đến sự khác biệt của hệ sinh thái mỗi nền tảng, khả năng sử dụng lại ở mức tốt cho từng nền tảng, cộng đồng phát triển tương đối mạnh. Tuy nhiên, việc không hỗ trợ cross-platform lại là điểm yếu của React native, tức là với React native chưa thể viết mã nguồn một lần cho nhiều nền tảng, ngoài ra React native chưa hỗ trợ windows phone, thiếu một số thành phần giao diện phát triển ứng dụng android như Map, Modal, Webview... Đây là điểm trừ lớn nhất của React native, khiến cho công nghệ này không thể phát triển được các ứng dụng có quy mô lớn, phức tạp.

*Xamarin [4]* : một công nghệ lập trình ứng dụng di động đa nền tảng hiện đại, Xamarin có những đặc điểm riêng biệt, hiếm có so với các frameworks hiện tại trên thị trường khi mà khả năng truy cập và trải nghiệm người dùng native vẫn đang bị đặt nghi vấn. Xamarin sở hữu logic ứng dụng được chia sẻ trên nhiều nền tảng. Xamarin cũng chia sẻ mã nguồn ứng dụng cơ bản của lớp giao diện như lệnh gọi của web services, các tương tác với cơ sở dữ liệu và tích hợp backend integration.

Ưu điểm lớn nhất của Xamarin [5] là sự làm việc cộng tác và chia sẻ mã nguồn dễ dàng. Sử dụng Xamarin có thể viết mã nguồn một lần và biên dịch cho nhiều nền tảng. Xamarin hỗ trợ nguồn đối tượng giao diện dồi dào như các thành phần giao diện gốc, các plugin đã được tùy biến hay các web service bên thứ ba. Ngoài ra, công cụ phát triển Xamarin là visual studio, một IDE rất chuyên nghiệp với nhiều chức năng hỗ trợ biên dịch, kiểm thử hay triển khai trên thiết bị thật. Hơn nữa, cộng đồng phát triển Xamarin đang ngày càng lớn mạnh, hứa hẹn đây sẽ là công nghệ phát triển ứng dụng di động của tương lai.

Tóm lược các ưu nhược điểm của các công nghệ phát triển ứng dụng di động đa nền tảng theo hình ảnh dưới đây



**Hình 5 So sánh công nghệ phát triển ứng dụng đa nền tảng**

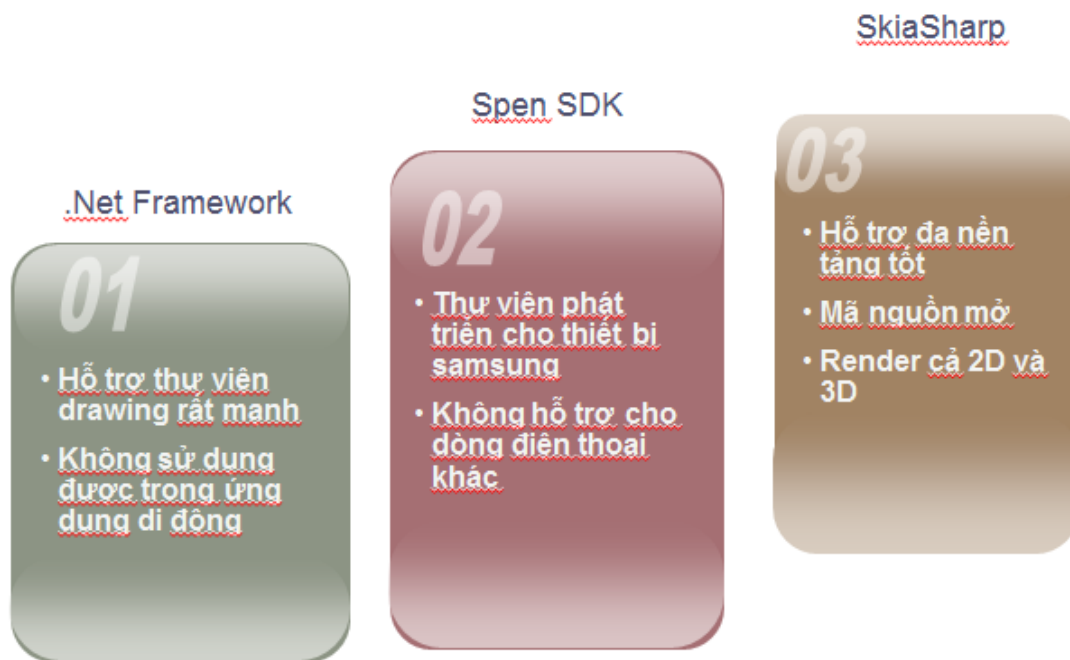
### ***Công nghệ hiển thị bài giảng***

#### ***Bài giảng chữ viết tay***

Việc hiển thị đường nét, chữ viết tay hiện nay có nhiều thư viện, SDK hỗ trợ cho mỗi nền tảng. Phải kể đến như .Net Framework có bộ thư viện System.Drawing rất mạnh trong việc vẽ đường nét hay SPEN SDK là bộ Kit hỗ trợ rất mạnh cho việc dùng bút ở điện thoại Samsung hay UIKit hỗ trợ vẽ cho các ứng dụng iOS.

Hình bên dưới tóm tắt các ưu nhược điểm của các công nghệ thông dụng hiển thị đường nét hiện nay:





**Hình 6 Công nghệ hiển thị chữ viết tay**

Những thư viện .NetFramework hay Spen SDK đều hỗ trợ rất tốt việc vẽ đường nét trên từng nền tảng Windows, Anrdroid. Điểm yếu lớn nhất của các thư viện, SDK này không hỗ trợ đa nền tảng nên chúng ta không thể sử dụng nó trong dự án.

Trong khi đó, thư viện bên thứ ba SkiaSharp là thư viện mã nguồn mở hỗ trợ tốt việc hiển thị đường nét ở cả 2D và 3D, đồng thời có thể build ứng dụng đa nền tảng. Chính vì thế, trong đề tài này, tôi đề xuất sử dụng công nghệ này để hiển thị bài giảng chữ viết tay. Chi tiết công nghệ , tôi xin trình bày ở phần tiếp theo.

#### *Bài giảng PDF*

PDF là định dạng dữ liệu phổ biến nhất hiện nay do đó việc thiết kế bài giảng sử dụng tài liệu PDF đã không còn xa lạ. Trên mỗi nền tảng, có nhiều thư viện hỗ trợ hiển thị định dạng dữ liệu PDF, thậm chí hỗ trợ cả những tính năng nâng cao như trình chiếu, animation... Các thư viện tiện dụng trên từng nền tảng có thể kể đến như muPDF (Anroid ) hay UIToolkit (iOS). Tuy nhiên những thư viện này không hỗ trợ cho việc phát triển ứng dụng đa nền tảng.



### Hình 7 Công nghệ hiển thị dữ liệu PDF

Trước thách thức phải hiển thị tốt dữ liệu trên đa nền tảng, 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. Việc hiển thị dữ liệu sẽ được tiến hành trên webview, việc hỗ trợ các thao tác dữ liệu sẽ được thực thi bởi JavaScript. Chi tiết của việc sử dụng bộ thư viện này để hiển thị dữ liệu PDF tôi sẽ trình bày ở phần tiếp theo.

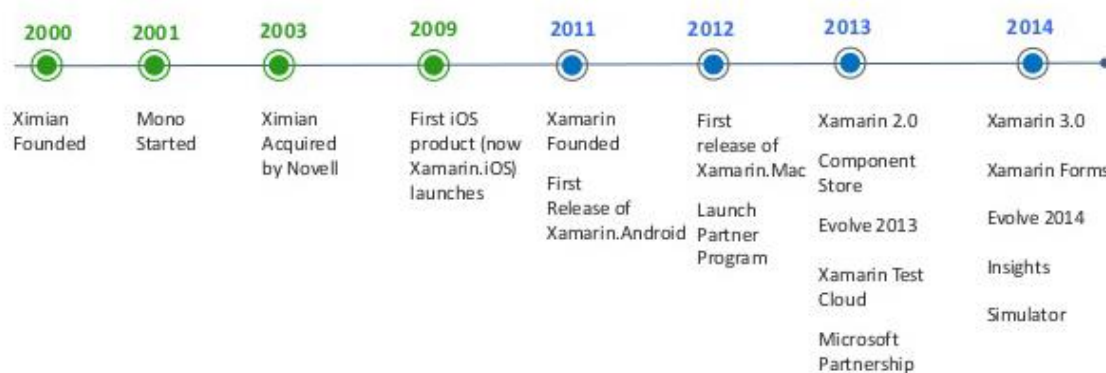
Tóm lại, theo những phân tích trên, để giải quyết được vấn đề, tôi sẽ sử dụng các công nghệ như sau :

- Cho việc phát triển ứng dụng di động đa nền tảng: Sử dụng Xamarin
- Cho việc hiển thị bài giảng XML: Sử dụng công nghệ hiển thị đường nét viết tay - SkiaSharp
- Cho việc hiển thị bài giảng PDF: Sử dụng công nghệ hiển thị PDF – PDFjs
- Cho việc chạy dữ liệu audio: Sử dụng dịch vụ hiện tại trên từng nền tảng

Phần tiếp theo, tôi xin trình bày chi tiết các công nghệ được sử dụng.

### 2.2.2.1. 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 [6]. 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 bị 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 8 Quá trình hình thành và phát triển của Xamarin**

### Kiến trúc Xamarin [7]

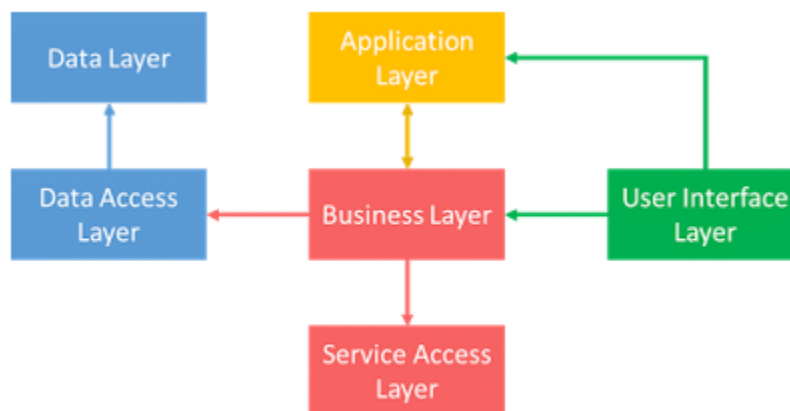
Ứng dụng di động đa nền tảng thông thường được chia kiến trúc thành 6 tầng:

- *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 giữa tầng UI Interface và tầng Business.
- *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 9 Kiến trúc tầng của xamarin**



**Hình 10 Kiến trúc liên kết của Xamarin**

#### **2.2.2.2. Những ưu điểm, nhược điểm của Xamarin**

##### **2.2.2.2.1 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 chức 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.

	iOS	Android	Windows Phone
C/C++	x	x	x (8.1)
Objective-C	x		
Java		x	
C#			x
Visual Basic .NET			x

**Hình 11 Các nền tảng hỗ trợ bởi Xamarin**

*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 phá của Xamarin trong việc phát triển ứng dụng di động đa nền tảng là 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 vụ 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. Tuy nhiên, từ năm 2016, sau khi sáp nhập vào Microsoft, bản quyền của Xamarin đã 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.

#### 2.2.2.2.2. 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.Forms. Đây cũng là công cụ chính để phát triển ứng dụng của đề tài.

### ***2.2.2.3. Phát triển ứng dụng di động đa nền tảng dựa trên Xamarin Form***

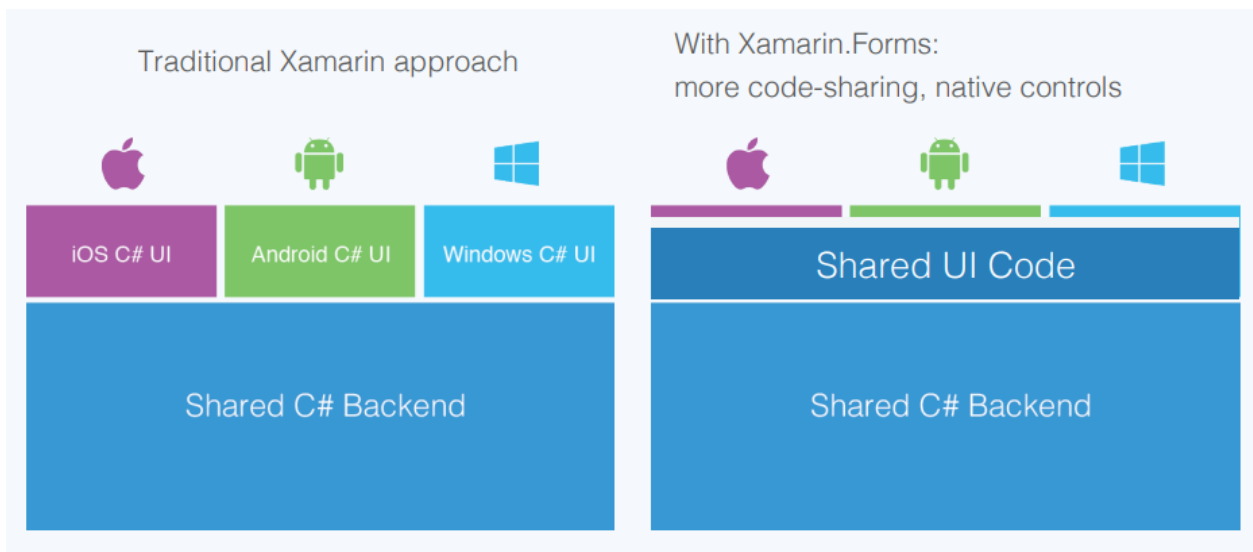
Xamarin.Forms là một công cụ để xây dựng các ứng dụng đa nền tảng có thể chạy được trên Windows, Android, iOS. Được phát triển bởi Microsoft nên có thể dùng C# để lập trình các ứng dụng bằng Xamarin. Công bố chính thức từ tháng 5/2014, Xamarin.Forms đang thể hiện là công cụ xây dựng ứng dụng di động đa nền tảng hiệu quả bậc nhất hiện nay [8]. Các phiên bản nền tảng ứng dụng được hỗ trợ bởi Xamarin.Forms hiện nay:

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

Xamarin.Forms 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.Forms và code native mà không gặp bất kì sự xung đột nào.

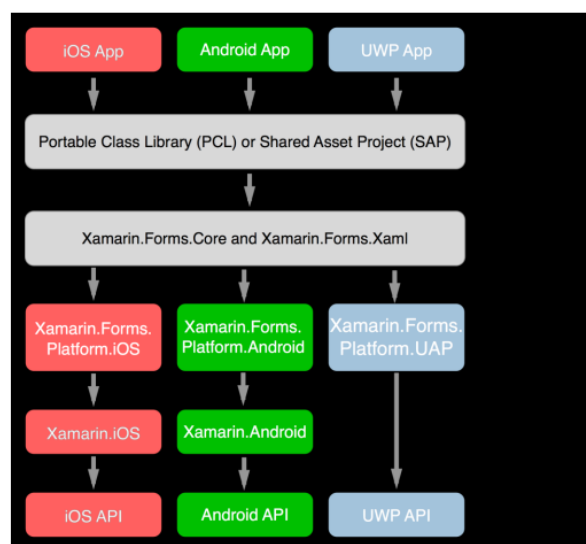
Xamarin.Forms được phát triển trên Xamarin truyền thống, nên nó được kế thừa hoàn toàn giống kiến trúc của Xamarin. Điểm mạnh của nó là có thể chia sẻ tối đa các thành phần code cho nhiều nền tảng (từ module giao diện cho đến business logic), nhờ đó, hiệu suất chia sẻ code cao rõ rệt. Hình bên dưới mô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





**Hình 12 Chia sẻ mã nguồn trong Xamarin.Form**

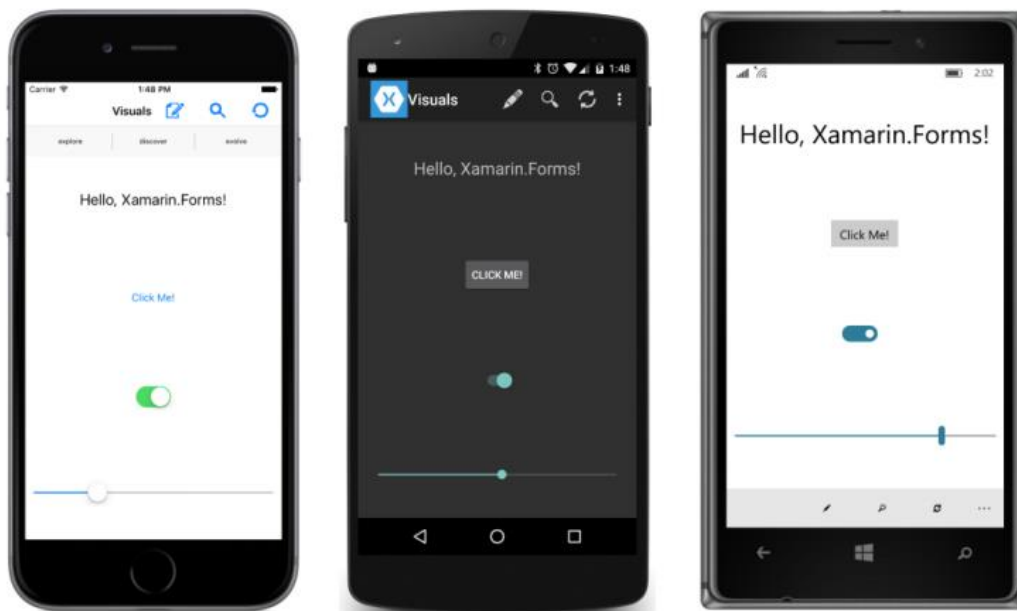
Như hình trên, có thể dễ dàng nhận ra việc chia sẻ tài nguyên của xamarin form tốt hơn rõ rệt. Với xamarin form, nhà phát triển có thể tạo ra những module dùng chung (giao diện và logic) cho mọi nền tảng. Nhưng cũng không mất tính native, chúng ta cũng có thể tạo những control, logic native vào dự án.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



**Hình 13 Chia sẻ mã nguồn trong từng nền tảng**

Tầng trên cùng là ứng dụng phát triển bởi Xamarin theo từng nền tảng. Việc sử dụng các dịch vụ, control giao diện sẽ được đóng gói, chia sẻ qua các thư viện PCL hoặc Shared Project. Tiếp đó xamarin sẽ có bộ chuyển đổi (gọi là adapter), chuyển mã nguồn sang dạng nhị phân tương thích với từng nền tảng. Sau bước này, các ứng dụng sẽ chạy như một ứng dụng di động native.

Hình bên dưới là kết quả sau khi chuyển đổi các thành phần giao diện



**Hình 14 Kết quả ứng dụng tạo bằng Xamarin.Form**

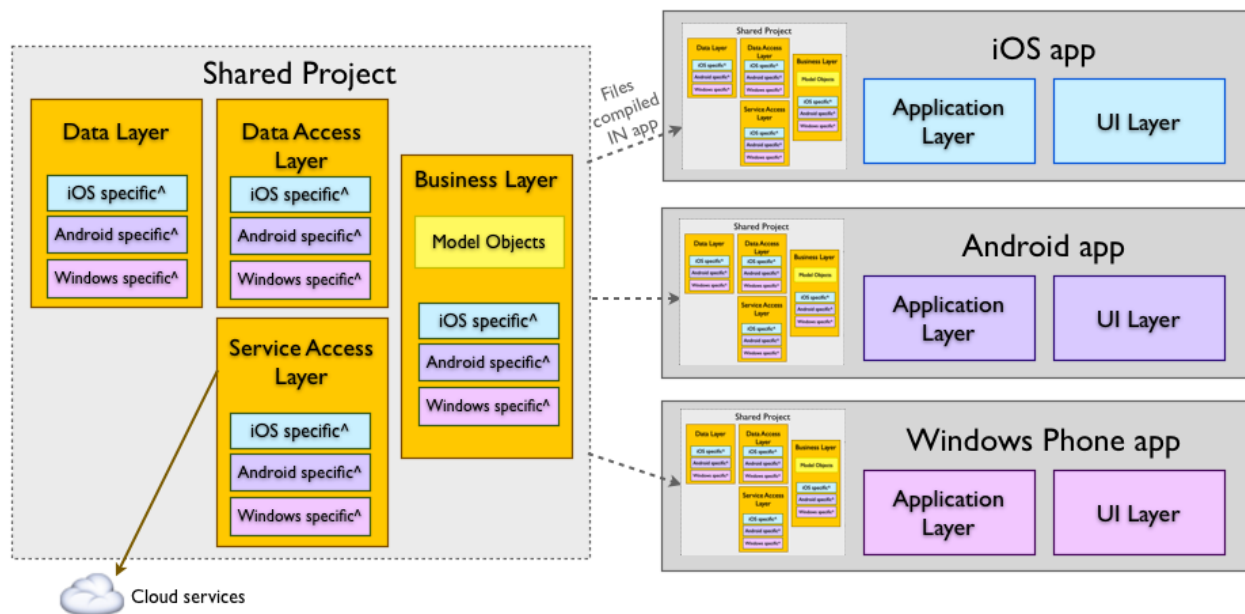
Sau bộ chuyển đổi (adapter), mã nguồn sẽ được biên dịch như một ứng dụng native, các control giao diện lúc này sẽ được hiển thị một cách riêng biệt theo từng nền tảng.

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 cận 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).

#### 2.2.2.3.1. 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. Sau khi biên dịch, chúng tạo ra các thành phần tham

chiều 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 15 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.

```

1 public string GetPlatform()
2 {
3     var path = "unknown";
4
5     #if WINDOWS_PHONE
6     path = "windowsphone";
7     #else
8
9     #if __IOS__
10    path = "ios";
11    #else
12
13    #if __ANDROID__
14    path = "android";
15    #endif
16
17    return path;
16 }

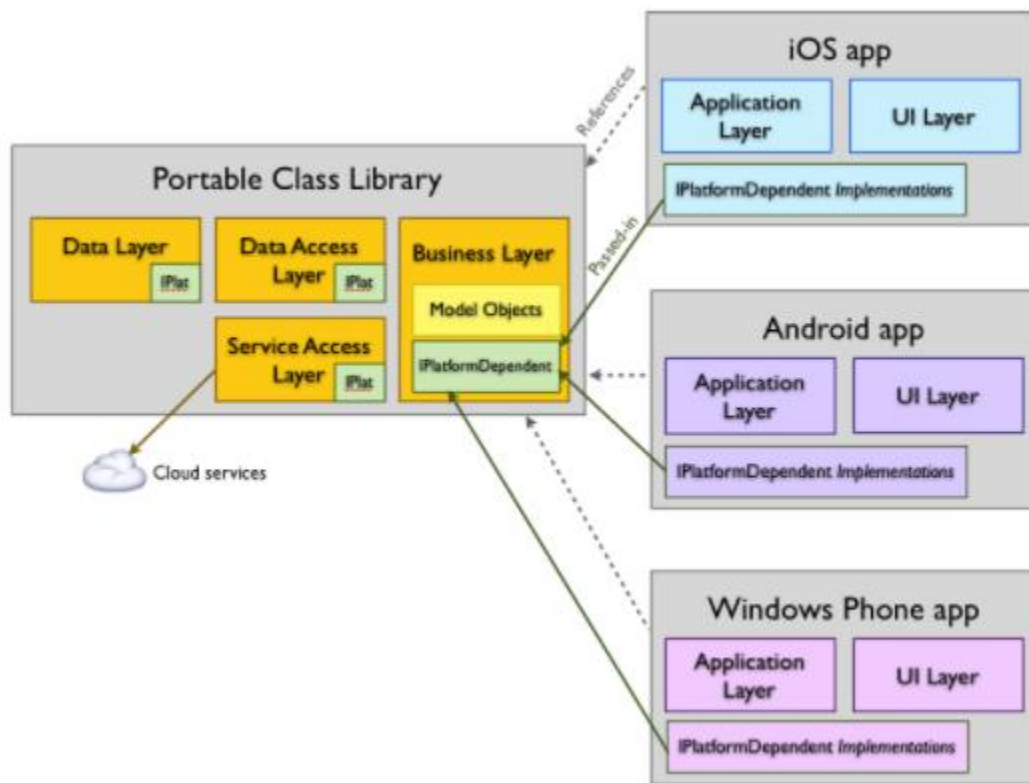
```

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”.

#### 2.2.2.3.2. 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 .NETPortable để import. Ví dụ trong .NETPortable 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 .NETPortable 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 16 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 nền 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	<code>public interface PlayAudio</code>
2	<code>{</code>
3	<code>    public void Play (String Source);</code>
4	<code>}</code>

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

```

Trong project Windows Mobile:

```

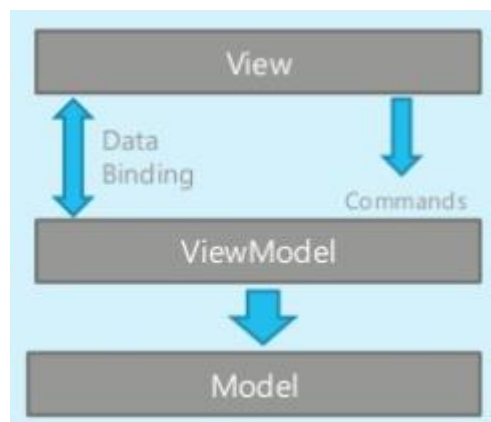
1 [assembly: Xamarin.Forms.Dependency(typeof(PlayAudioImplement))]
2 namespace Project.Windows
3 {
4     public class PlayAudioImplement:PlayAudio
5     {
6         public async void Play (String Source)
7         {
8             Folder = await InstalledLocation.GetFolderAsync("Assets");

```

9	StorageFile sf = await Folder.GetFilesAsync(Source);
10	MediaElement Player = new MediaElement();
11	Player.SetSource(await sf.OpenReadAsync(), sf.ContentType);
12	Player.Play();
13	}
14	}
15	}

#### 2.2.2.4. Mô hình MVVM

MVVM [9] 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 **Model – View – ViewModel**, hình ảnh dưới đây sẽ mô tả mối quan hệ giữa các thành phần trong mô hình này



**Hình 17 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 chắn 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, trong đó: M tương ứng với Bussiness layer, V tương ứng với UI layer và VM tương ứng với Application layer. Đề tài cũng đang áp dụng mô hình MVVM trong việc thiết kế kết nối các module của dự án.

Việc giảm dung lượng bài giảng, chèn tương tác xen giữa 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.

#### **2.2.2.5. *Hiển thị dữ liệu chữ viết tay với SkiaSharp***

Skia là một 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 [10]. Đây chính là nền tảng cốt lõi trong việc render hình ảnh của trình duyệt web google chrome và chrome OS, android, Mozilla 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:

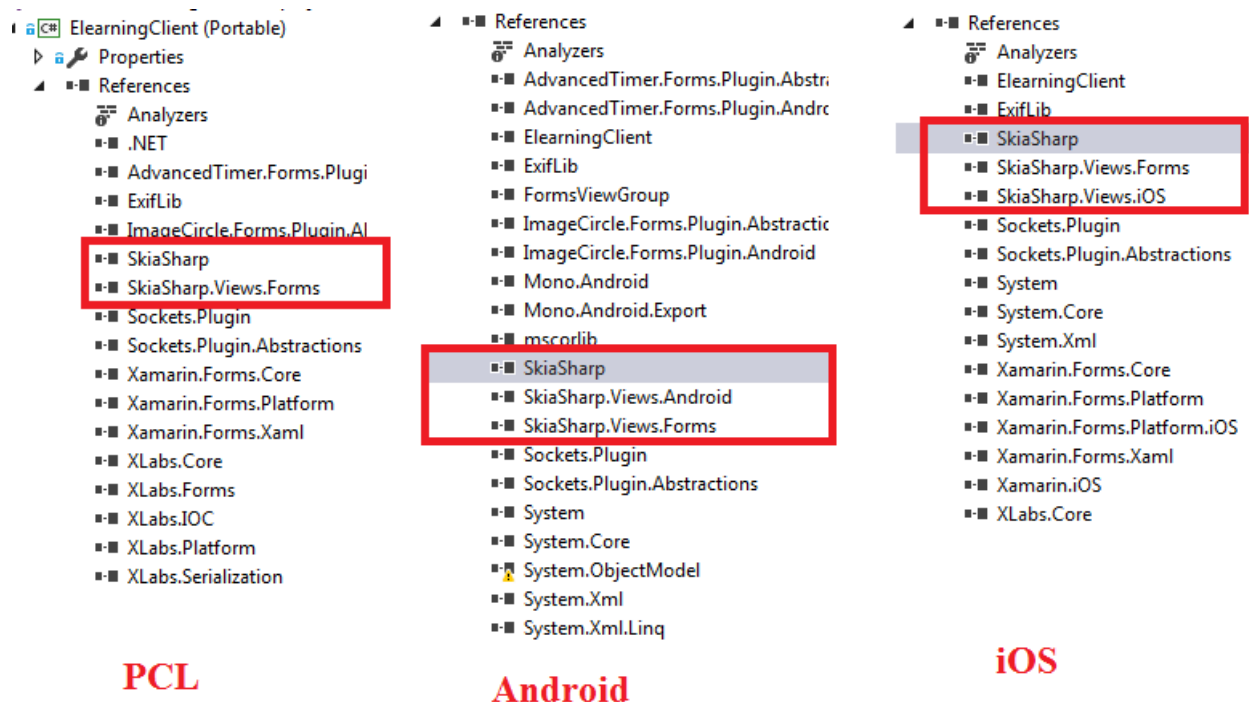
- .NET Core / .NET Standard 1.3
- Xamarin.Android
- Xamarin.iOS
- Xamarin.tvOS
- Xamarin.Mac
- Windows Classic Desktop (Windows.Forms / WPF)
- Windows UWP (Desktop / Mobile / Xbox / HoloLens)

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

### **Bảng 1 Nuget cài đặt SkiaSharp**

```
nuget install SkiaSharp
```

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



**Hình 18 Render chữ viết tay với SkiaSharp**

Với mỗi nền tảng, SkiaSharp đã được tùy biến, tinh chỉnh sao cho phù hợp với các thành phần lõi, do đó mới cần có thư viện phân tán cho từng nền tảng. Việc này hỗ trợ tốt cho việc sử dụng thư viện vẽ, tuy nhiên, nó lại làm tăng dung lượng của bộ cài đặt sau khi trình biên dịch mã nguồn.

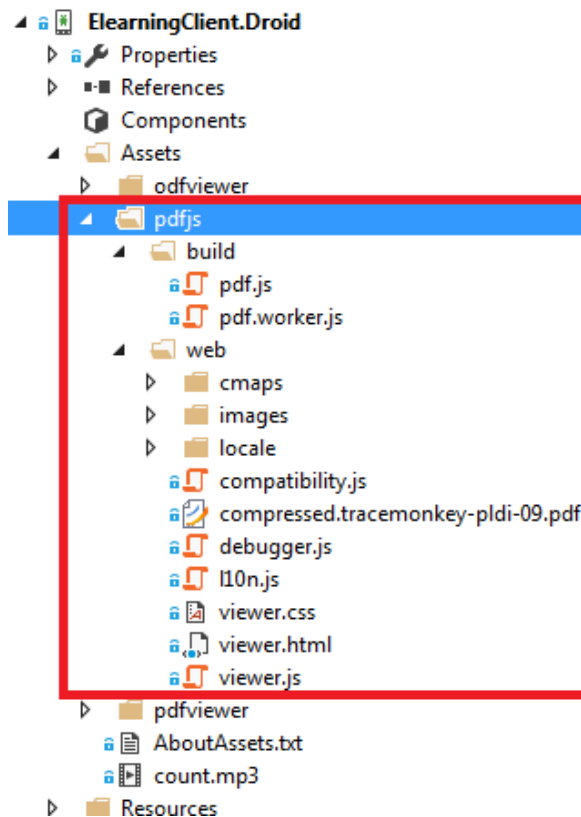
Để render đường nét sử dụng skia, đơn giản chúng ta chỉ cần ghi đè hàm

```
void OnPainting(object sender, SKPaintSurfaceEventArgs e)
```

#### **2.2.2.6. Hiển thị bài giảng PDF với PDFjs**

PDF.js là công cụ hiển thị dữ liệu PDF được build trên HTML5. PDF.js được phát triển bởi Mozilla labs với mục tiêu tạo ra phương thức chung cho nền tảng dựa web cho việc phân tích và hiển thị PDFs [11]. PDF js hỗ trợ cho cả trình duyệt Firefox và Chrome, 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 mozilla rồi thêm vào project.



**Hình 19** Hiện thị dữ liệu PDF với PDFjs

PDFjs bao gồm bộ mã nguồn html, js, css hỗ trợ cho việc hiển thị, style trang hiển thị. Với các file js, bộ thư viện này hỗ trợ sử dụng những chức năng cơ bản đối với một file PDF như chuyển trang trước, chuyển trang sau, phóng to, thu nhỏ, tự động căn chỉnh vùng hiển thị...

Chúng ta để đường 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.Forms. Đồng thời, tôi cũng đã đi chi tiết vào công nghệ sử dụng trong project. Phần tiếp, tôi xin trình bày cụ thể

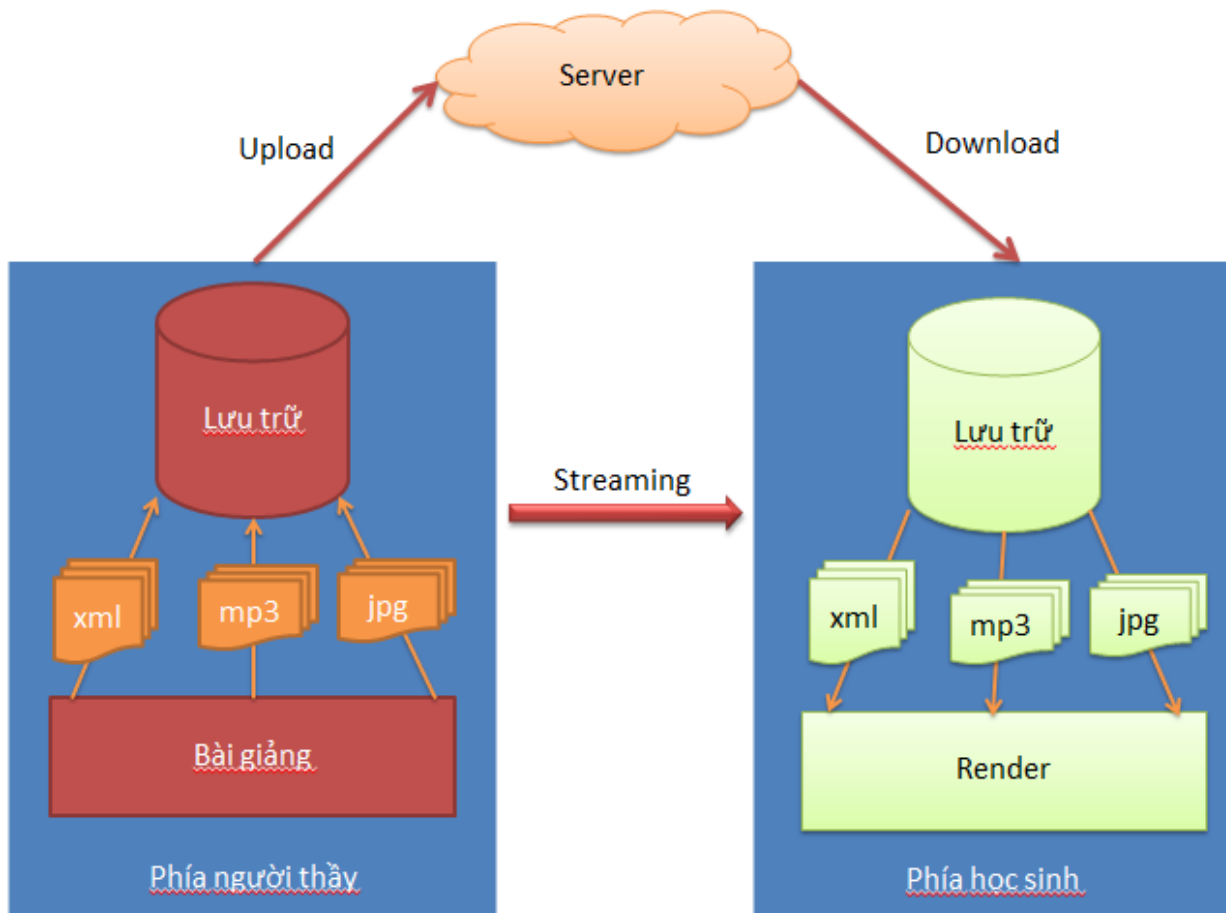
việc cài đặt, sử dụng xamarin.form trong ứng dụng hỗ trợ học online cho học sinh, sinh viên.

## CHƯƠNG 3: XÂY DỰNG ỨNG DỤNG HỖ TRỢ HỌC TRỰC TUYẾN - ELEARNINGCLIENT

Trong phần này, tôi sẽ trình bày chi tiết về hệ thống học trực tuyến giải quyết các vấn đề đã nêu và chi tiết việc thiết kế ứng dụng hỗ trợ học sinh, sinh viên trong việc học trực tuyến – ElearningClient.

### 3.1. Kiến trúc hệ thống

Hệ thống học trực tuyến được tạo ra với nhiệm vụ giải quyết các vấn đề lưu trữ, truyền tải, tương tác, học thời gian thực như đã nêu ở phần trên. Hình bên dưới mô tả các thành phần, chức năng chính của hệ thống.



Hình 20 Mô hình hệ thống

Hệ thống bao gồm 3 phần chính : Phía người thầy, phía học sinh và server truyền tải, lưu trữ. Nhiệm vụ của phía người thầy là tạo ra bài giảng với chuẩn định dạng dữ liệu thống nhất rồi lưu trữ cục bộ và truyền tải lên server. Nhiệm vụ phía học sinh là tải bài giảng từ server, lưu trữ cục bộ và hiển thị bài giảng. Trong khuôn khổ của bài luận văn, tôi chỉ tập trung vào phía học sinh và một số tương tác của phía học sinh với server.

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 hiển thị 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 bên phía client (hiển thị). 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
- Hiển thị bài giảng viết tay
- Hiển thị 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

## **3.2. Thiết kế ứng dụng E-LearningClient**

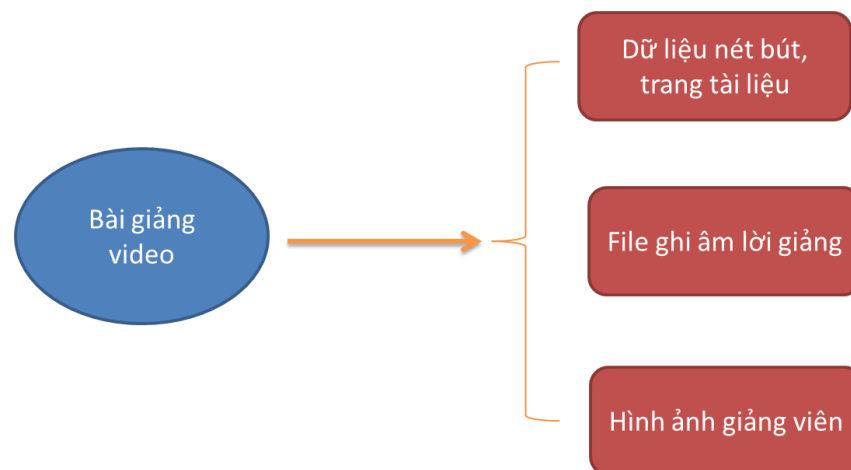
### **3.2.1. Mô hình hóa dữ liệu video**

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, giúp làm giảm dung lượng lưu trữ, việc tạo ra dữ liệu này đơn giản, không cần hỗ trợ với những thiết bị hay phần mềm đặc trưng. Bên cạnh đó, sản phẩm của bài giảng vẫn không mất tính truyền thống, trong khi dữ liệu được hiển thị lại với nội dung tương tự như bài giảng video. Hơn nữa, với mô hình dữ liệu này, người thầy dễ dàng chỉnh sửa bố cục bài giảng bằng cách chèn thêm những phản tương tác, hỏi đáp, câu hỏi trắc nghiệm đánh giá sự hiểu bài của học viên. Qua đó, bài giảng bớt nhàm chán, học viên có thể kiểm tra mức độ hiểu bài ngay trong bài học.

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 gian giảng bài, 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 21 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. Đồng thời, khi bài giảng chạy trên ứng dụng, dữ liệu nét bút, trang tài liệu và âm thanh sẽ được trình chiếu một cách đồng bộ hóa, giống như đang hiển thị một bài giảng video.



Vấn đề đồng bộ hóa được giải quyết bằng việc định dữ liệu thời gian ứng với từng thao tác của người thầy. Cùng với một trục thời gian xác định từ đầu đến cuối bài giảng, dữ liệu hiển thị của bài giảng và audio sẽ được phát ra một cách độc lập, song song với nhau. Tương ứng với mỗi loại bài giảng, dữ liệu hiển thị và dữ liệu âm thanh sẽ được đồng bộ theo cách thức như sau:

*Bài giảng viết tay:* Mỗi nét vẽ được xác định bởi đường nối các điểm (từ lúc bút nhấn xuống cho đến nhấc lên). Để vẽ được nét này, tôi sử dụng timer với timelapse linh động qua từng nét bút. Như vậy, thời gian vẽ nét bút sẽ được phân bố đều trên trục thời gian và sẽ đồng bộ hóa một cách tương đối với dữ liệu âm thanh.

*Bài giảng PDF:* Người thầy chuyển trang tài liệu sẽ được ấn định bởi một mốc thời gian, trục thời gian vẫn là từ đầu đến cuối bài giảng. Bằng việc sử dụng một timer với timelapse cố định (rất nhanh) và một con trỏ đếm thời gian, chương trình sẽ tính toán chính xác thời gian chuyển trang đồng bộ với dữ liệu âm thanh.

Hình ảnh bên dưới là dữ liệu của 2 loại bài giảng chính (pdf và chữ viết tay) và cấu trúc của một bài kiểm tra trắc nghiệm tương tác.

**Bảng 2 Chi tiết dữ liệu mô tả video**

<pre> &lt;ClassRoom&gt;   &lt;ClassInfor&gt;     &lt;Teacher&gt;       &lt;Name&gt;Vung&lt;/Name&gt;       &lt;Avatar&gt;&lt;/Avatar&gt;     &lt;/Teacher&gt;     &lt;SubjectName&gt;Math&lt;/SubjectName&gt;     &lt;DocumentList&gt;       &lt;Document&gt;test.pdf&lt;/Document&gt;     &lt;/DocumentList&gt;   &lt;/ClassInfor&gt;   &lt;LectureDetail&gt;     &lt;Detail&gt;       &lt;Time&gt;1&lt;/Time&gt;       &lt;Document&gt;test.pdf&lt;/Document&gt;       &lt;Page&gt;1&lt;/Page&gt;     &lt;/Detail&gt;     &lt;Detail&gt;       &lt;Time&gt;3&lt;/Time&gt;       &lt;Document&gt;test.pdf&lt;/Document&gt;       &lt;Page&gt;2&lt;/Page&gt;     &lt;/Detail&gt;     &lt;Detail&gt;       &lt;Time&gt;5&lt;/Time&gt;       &lt;Document&gt;test.pdf&lt;/Document&gt;       &lt;Page&gt;3&lt;/Page&gt;     &lt;/Detail&gt;     &lt;Detail&gt;       &lt;Time&gt;8&lt;/Time&gt;       &lt;Document&gt;test.pdf&lt;/Document&gt;       &lt;Page&gt;4&lt;/Page&gt;     &lt;/Detail&gt;     &lt;Detail&gt;       &lt;Time&gt;9&lt;/Time&gt; </pre>	<pre> &lt;HandWritingData&gt;   &lt;DOWN x="18.74949" y="318.18335"/&gt;   &lt;MOVE x="15.155839" y="321.41248"/&gt;   &lt;MOVE x="23.801437" y="318.54794"/&gt;   &lt;MOVE x="33.38451" y="312.298"/&gt;   &lt;MOVE x="53.852707" y="294.4857"/&gt;   &lt;MOVE x="68.279396" y="278.7046"/&gt;   &lt;MOVE x="80.9353" y="261.62143"/&gt;   &lt;MOVE x="92.08083" y="242.0383"/&gt;   &lt;MOVE x="106.611694" y="208.60114"/&gt;   &lt;MOVE x="113.53859" y="186.46594"/&gt;   &lt;MOVE x="118.3822" y="165.47658"/&gt;   &lt;MOVE x="121.71545" y="146.98718"/&gt;   &lt;MOVE x="123.69456" y="132.76857"/&gt;   &lt;MOVE x="125.98617" y="119.22705"/&gt;   &lt;MOVE x="127.80903" y="115.00835"/&gt;   &lt;MOVE x="130.15273" y="114.2271"/&gt;   &lt;MOVE x="133.53804" y="117.664566"/&gt;   &lt;MOVE x="139.68372" y="132.14359"/&gt;   &lt;MOVE x="143.90234" y="147.29968"/&gt;   &lt;MOVE x="148.27722" y="165.6849"/&gt;   &lt;MOVE x="152.80835" y="186.83054"/&gt;   &lt;MOVE x="157.13115" y="209.69489"/&gt;   &lt;MOVE x="162.49559" y="245.00702"/&gt;   &lt;MOVE x="165.15176" y="268.02762"/&gt;   &lt;MOVE x="167.02672" y="289.5378"/&gt;   &lt;MOVE x="168.17252" y="309.64178"/&gt;   &lt;MOVE x="169.11" y="336.41232"/&gt;   &lt;MOVE x="169.68289" y="363.4954"/&gt;   &lt;MOVE x="170.15163" y="374.0682"/&gt;   &lt;MOVE x="172.23491" y="381.0994"/&gt;   &lt;MOVE x="175.04733" y="380.9952"/&gt;   &lt;UP x="175.04733" y="380.9952" time="1000"/&gt;   &lt;DOWN x="244.99335" y="146.83093"/&gt; </pre>
---	--

Hình bên trái là chi tiết dữ liệu mô hình hóa của bài giảng PDF với các trường tương ứng như sau:

*Document List*: Cho biết những tài liệu nào được sử dụng trong suốt bài giảng

*LectureDetail*: Chi tiết mô hình hóa dữ liệu bài giảng với thông tin như sau:

- Time: Thời gian chuyển trang
- Document: Tên tài liệu đang xem
- Page: Trang hiện tại đang hiển thị

Hình bên phải là chi tiết mô hình hóa dữ liệu của bài giảng viết tay với các trường tương ứng như sau:

*DOWN*: Trạng thái nhấn bút xuống với tọa độ bút trên màn hình là x và y

*MOVE*: Trạng thái bút di chuyển, vẽ trên màn hình với tọa độ bút tương ứng là x và y.

UP: Trạng thái bút nhắc lên với tọa độ bút trên màn hình là x và y. Khi đến trạng thái UP, một nét bút được ghi nhận là kết thúc. Trường *time* thể hiện thời điểm kết thúc nét bút. Từ trường *time* ta có thể tính khoảng thời gian giữa nét bút thứ nhất và nét bút thứ hai, qua đó tính toán lại thời gian render nét bút.

**Bảng 3 Cấu trúc dữ liệu bài trắc nghiệm tương tác**

```
<?xml version="1.0" encoding="UTF-8"?>
<QuickTest>
  <Information>
    <Lecture>Math</Lecture>
    <TestTime>300</TestTime>
    <NumberQuestion>5</NumberQuestion>
  </Information>
  <Detail>
    <Quiz>
      <ID>1</ID>
      <Content>Quick add : 123 + 327 = ?</Content>
      <Ans>
        <A>1291</A>
        <A>380</A>
        <A>450</A>
        <A>490</A>
        <Correct>450</Correct>
      </Ans>
    </Quiz>
    <Quiz>
      <ID>2</ID>
      <Content>Quick add : 19 + 381 = ?</Content>
      <Ans>
        <A>23</A>
        <A>456</A>
        <A>400</A>
        <A>490</A>
        <Correct>400</Correct>
      </Ans>
    </Quiz>
    <Quiz>
      <ID>3</ID>
      <Content>Quick Calculate : 1 + 1 x 2 = ?</Content>
      <Ans>
        <A>5</A>
        <A>4</A>
        <A>3</A>
        <A>6</A>
        <Correct>3</Correct>
      </Ans>
    </Quiz>
    <Quiz>
      <ID>4</ID>
      <Content>Quick add : 23 + 97 = ?</Content>
      <Ans>
        <A>120</A>
        <A>230</A>
        <A>130</A>
        <A>342</A>
        <Correct>120</Correct>
      </Ans>
    </Quiz>
  </Detail>
</QuickTest>
```

Việc chèn bài tương tác vào cấu trúc dữ liệu mô hình hóa là một điểm mạnh của loại dữ liệu này với việc dễ dàng chỉnh sửa dữ liệu, tùy biến theo cách riêng của mỗi người.

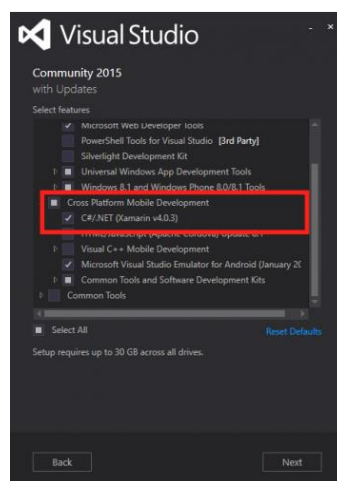
Trong quá trình học, để tránh sự nhầm lẫn của bài giảng, người thầy có thể tạo các bài tương tác xen giữa bài giảng. Trong khuôn khổ luận văn, tôi xin lấy ví dụ bài tương tác là bài kiểm tra trắc nghiệm xen giữa. Người thầy có thể tạo bài tương tác trắc nghiệm theo cấu trúc 4 đáp án, bài giảng sẽ được tạm ngừng cho học viên làm bài tương tác rồi tiếp tục học khi bài tương tác được hoàn thành.

### 3.2.2. 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 :

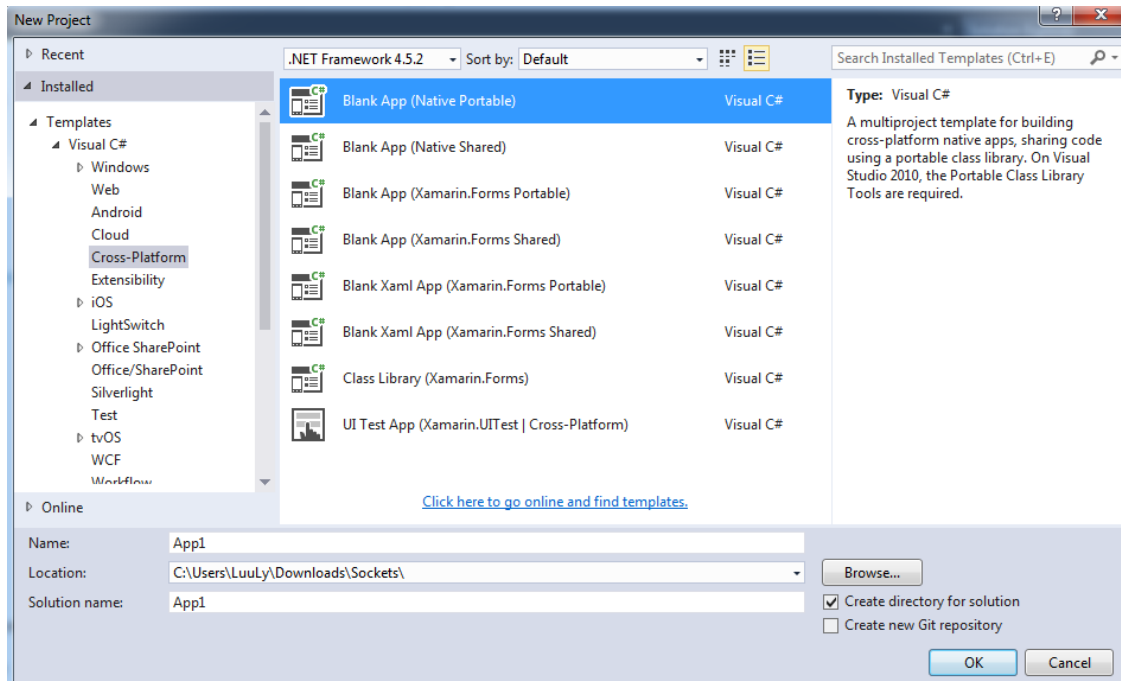
- Xamarin được hỗ trợ ở một trong những phiên bản sau: Visual studio 2012 update 5 trở lên, visual studio 2013 update 3, visual studio 2015 update 2, visual studio 2017
- Máy tính windows 8.1 với xung nhịp bộ xử lý tối thiểu 1.6GHz, 1GB RAM, 5GB ổ đĩa, Directx 9 để hiển thị độ phân giải cao. Để trải 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 22 Cài đặt môi trường phát triển xamarin**

Tất cả những gói hỗ 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 sàng, người dùng dễ dàng tạo một dự án cho việc phát triển.



**Hình 23 Tạo dự án mới với Xamarin**

Nếu dùng hệ điều hành windows, project build ra những gói cài đặt được ngay 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. Như hình trên ta thấy, Xamarin hỗ trợ nhiều loại ứng dụng đa nền tảng, hoặc phát triển trên từng nền tảng với ngôn ngữ thiết kế giao diện XAML, phát triển thư viện sử dụng lại (plugins).

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).

### 3.2.3. Thiết kế ứng dụng

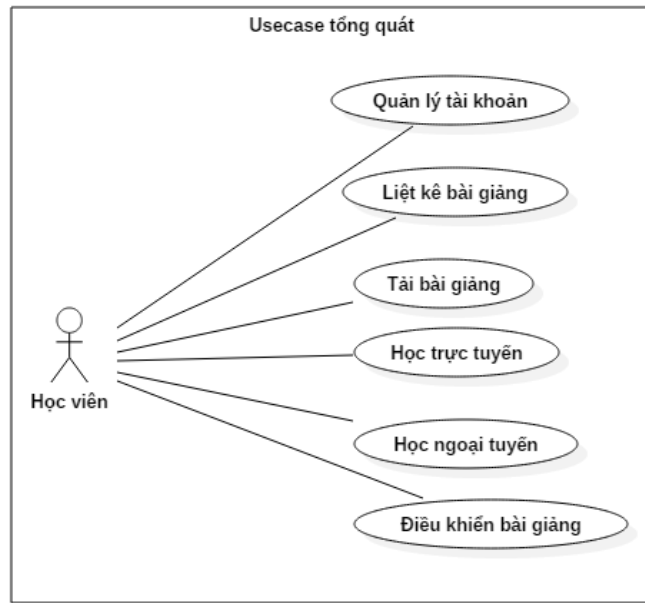
Ứng dụng hỗ trợ học trực tuyến cho học sinh (E-LearningClient) được tạo ra nhằm giải quyết các vấn đề lưu trữ, truyền tải, hiển thị đã nêu trên. Ứng dụng được phát triển trên nền Xamarin.Forms, sử dụng ngôn ngữ C#, ngôn ngữ thiết kế giao diện XAML trên mô hình MVVM. Các chức năng của ứng dụng như:

- Quản lý tài khoản – Đăng kí, đăng nhập
- Liệt kê bài giảng – Liệt kê bài giảng từ server sau khi đăng nhập
- Đồng bộ bài giảng – Định kỳ đồng bộ, liệt kê bài giảng
- Tải bài giảng – Download bài giảng từ server
- Học trực tuyến – Bài giảng thời gian thực giữa người thầy và học viên (đang phát triển)
- Học ngoại tuyến – Học viên tải bài giảng, lưu trữ lại rồi học ngoại tuyến
- Tương tác bài giảng – xen giữa bài giảng là các bài tương tác

Các phần tiếp theo tôi xin trình bày các biểu đồ sử dụng, tuần tự, hoạt động của từng chức năng.

### 3.2.3.1 Các biểu đồ Usecase của ứng dụng

#### 3.2.3.1.1 Usecase tổng quát

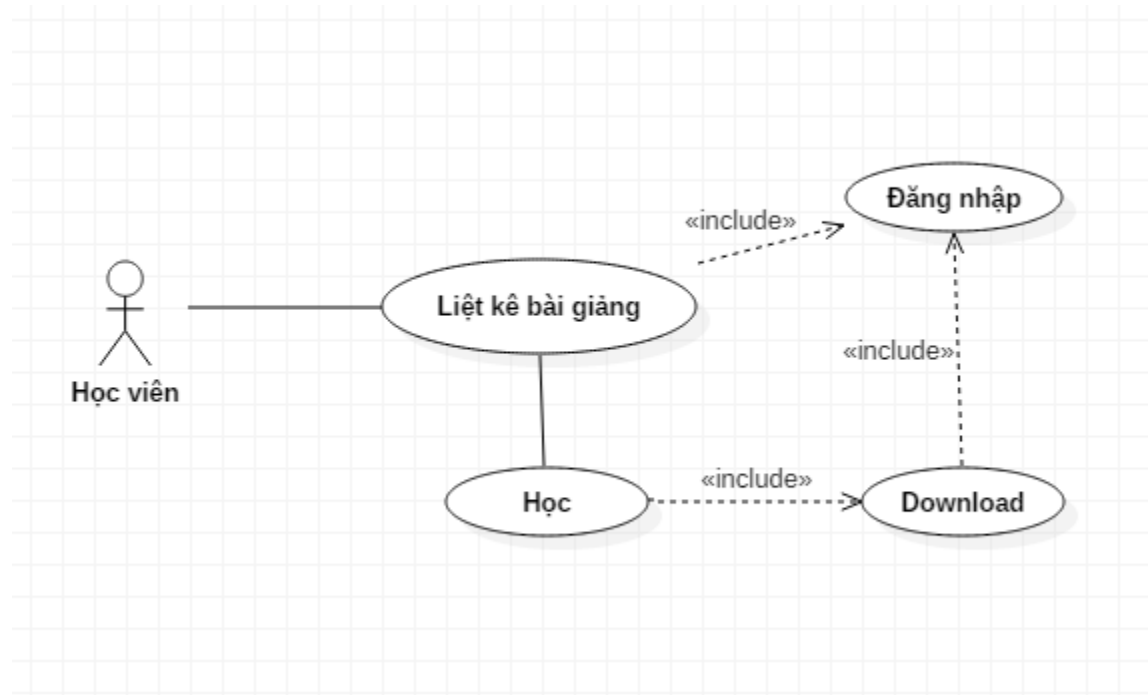


**Hình 24 Biểu đồ usecase tổng quát**

Trong biểu đồ usecase tổng quát, có tác nhân chính là học viên

- Học viên 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.

### 3.2.3.1.2. Usecase liệt kê bài giảng



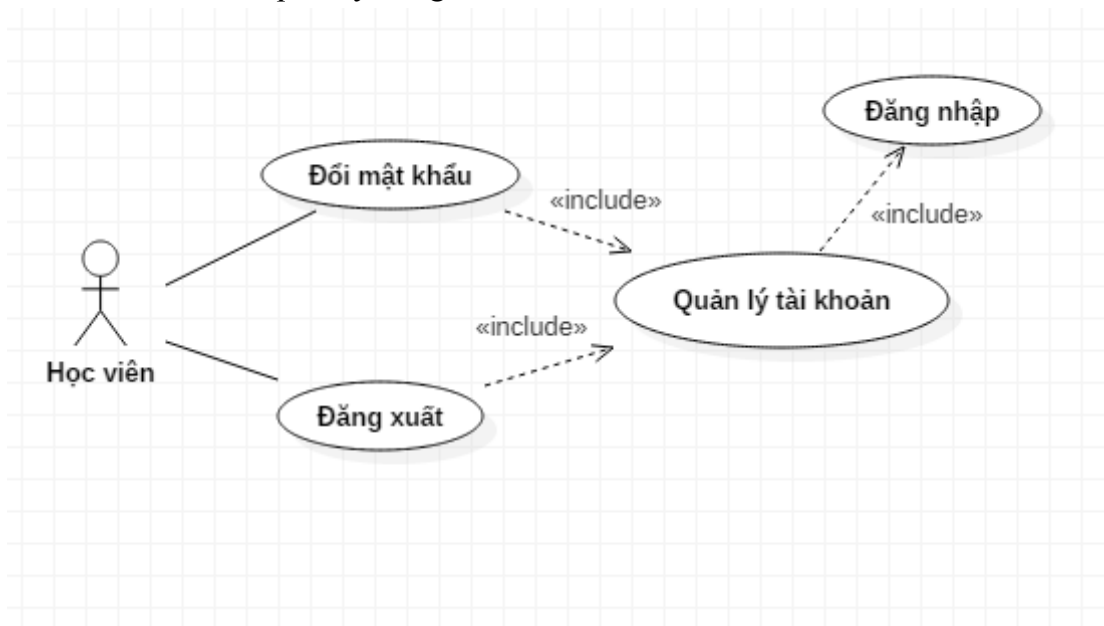
**Hình 25 Usecase liệt kê bài giảng**

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.

Việc kết nối với server phải thông qua một php webservice hỗ trợ cho việc truy vấn cơ sở dữ liệu. Từ đó ứng dụng có thể lấy được các bản ghi chứa thông tin bài giảng, giảng viên. Ngoài ra, qua việc liệt kê bài giảng, học viên có thể quản lý được tiến độ học, lịch trình học của mình, tránh việc tham gia quá nhiều lớp, tải về quá nhiều bài giảng.



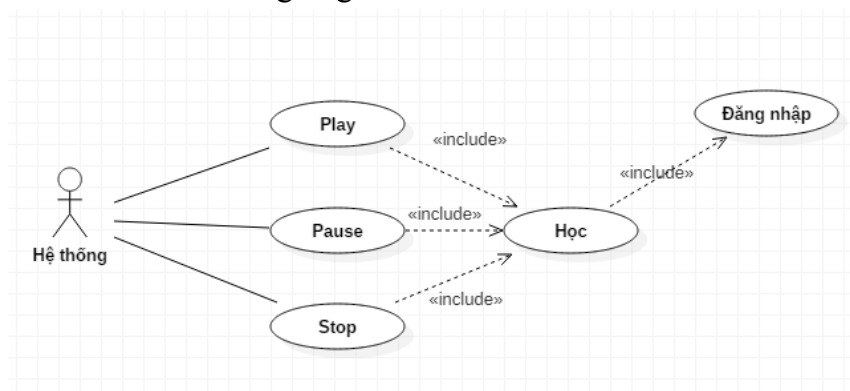
### 3.2.3.1.3. Usecase quản lý đăng tài khoản



**Hình 26 Usecase quản lý đăng tài khoản**

Học viên có thể quản lý tài khoản của mình với những thao tác 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

### 3.2.3.1.4. Usecase điều khiển bài giảng



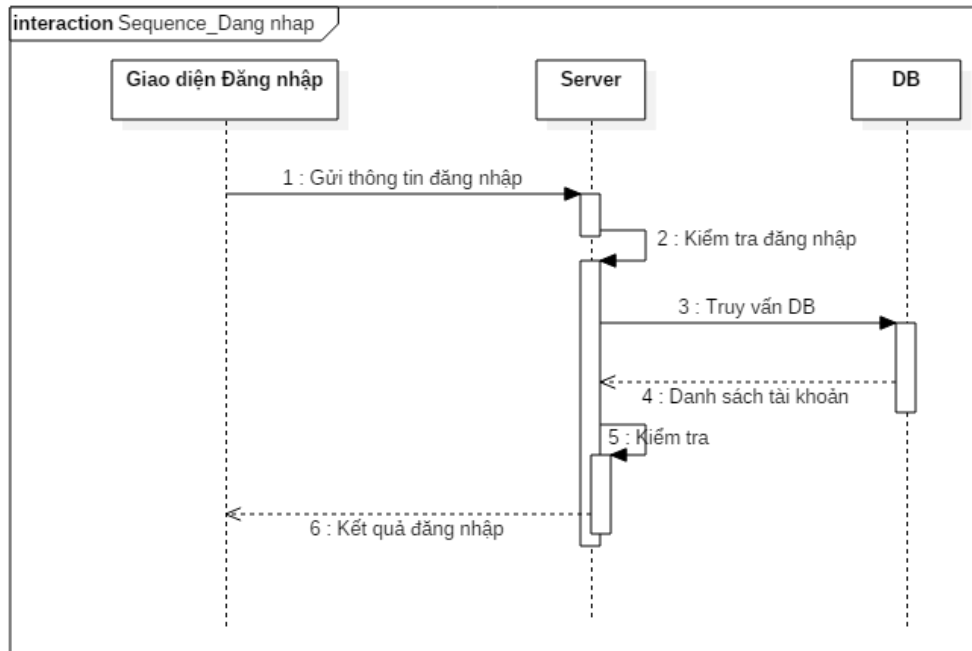
**Hình 27 Usecase điều khiển bài giảng**

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.

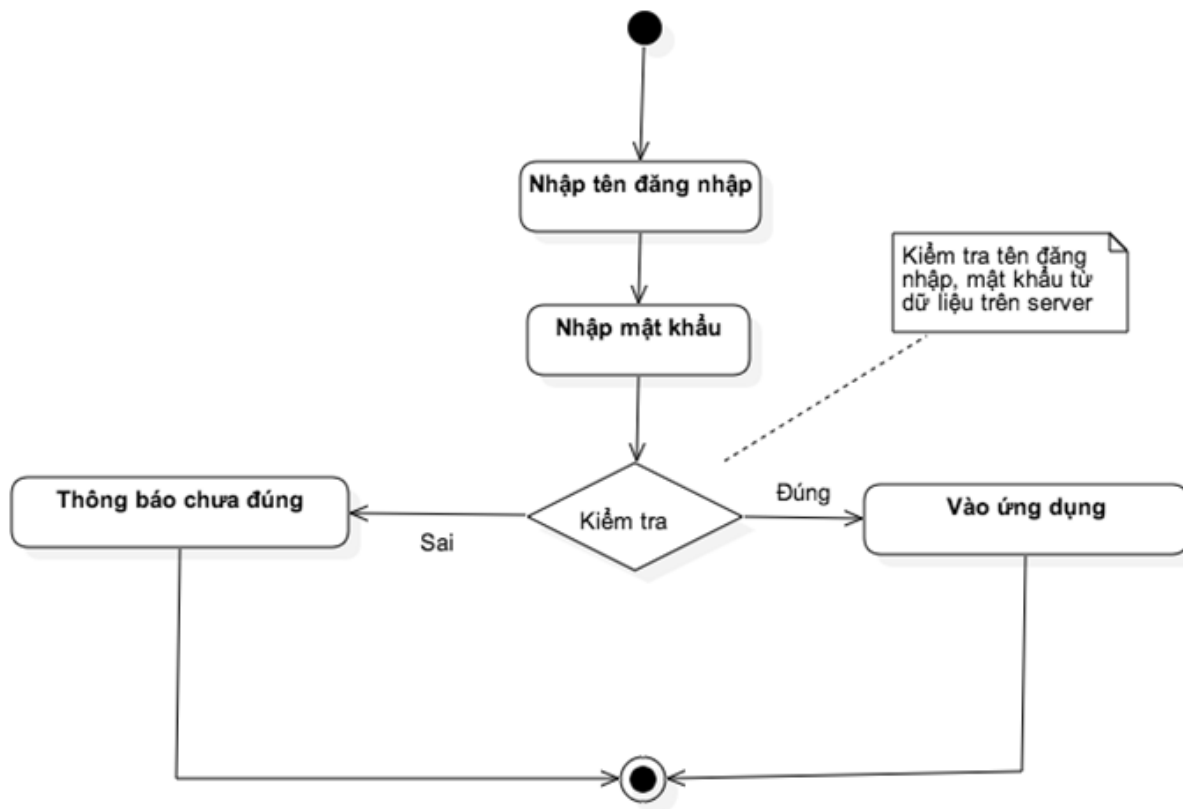
### 3.2.3.2. Biểu đồ tuần tự và biểu đồ hoạt động

#### 3.2.3.2.1. Đăng nhập hệ thống



**Hình 28 Sequence đăng nhập hệ thống**

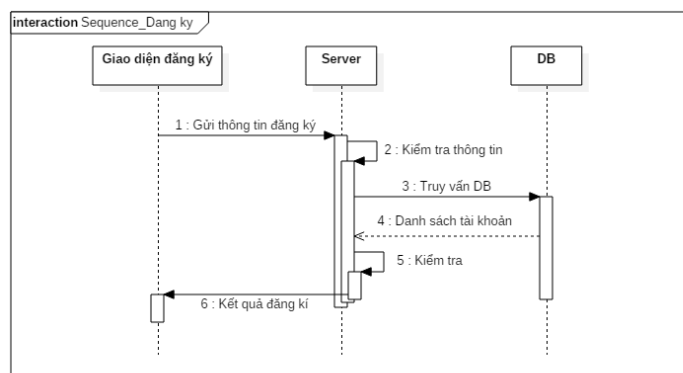
Người dùng sau khi nhập tên đăng nhập, mật khẩu và nhấn nút “Đăng nhập”, ứng dụng sẽ gửi bản tin kiểm tra tài khoản (kèm thông tin tài khoản) lên server, server sẽ truy vấn cơ sở dữ liệu và kiểm tra tính hợp lệ của tài khoản, sau đó trả về kết quả cho ứng dụng. Sau khi đăng nhập xong, người dùng có thể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



**Hình 29 Activity đăng nhập hệ thống**

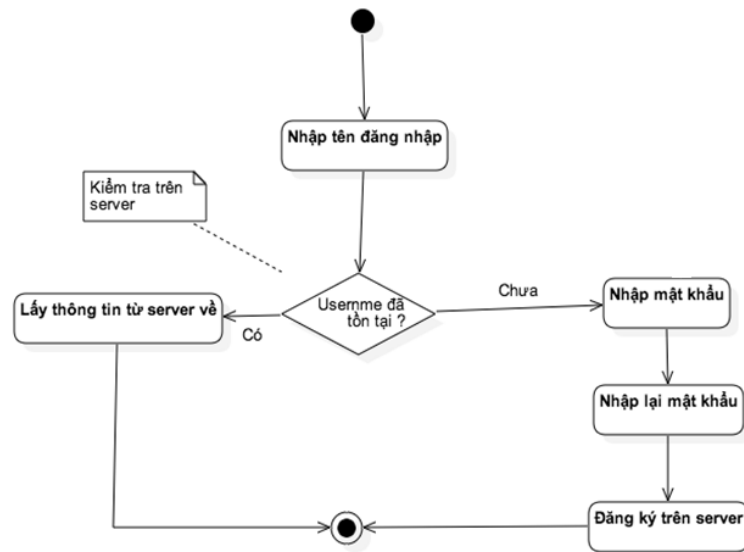
Biểu đồ hoạt động trên mô tả luồng dữ liệu và các tương tác người - ứng dụng trong việc đăng nhập hệ thống. Ứng dụng sẽ kiểm tra thông tin đăng nhập bằng việc gửi thông tin đăng nhập lên server, webserivce sẽ truy vấn cơ sở dữ liệu rồi trả về cho ứng dụng thông tin đăng nhập này có hợp lệ hay không.

#### 3.2.3.2.2. Đăng kí tài khoản



**Hình 30 Sequence Đăng kí tài khoản**

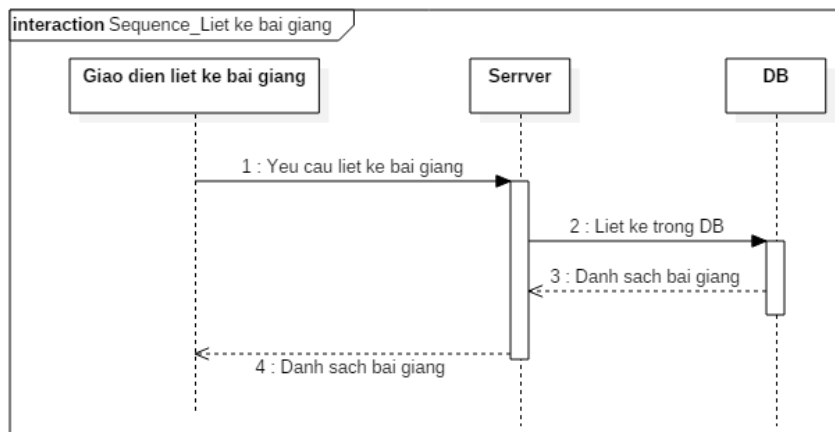
Người dùng nhập cá thông tin tài khoản: tên tài khoản, mật khẩu, nhắc lại mật khẩu, rồi gửi đăng ký lên server. Server sẽ kiểm tra thông tin đăng ký rồi trả về kết quả đăng ký thành công hay không.



**Hình 31 Activity Đăng kí tài khoản**

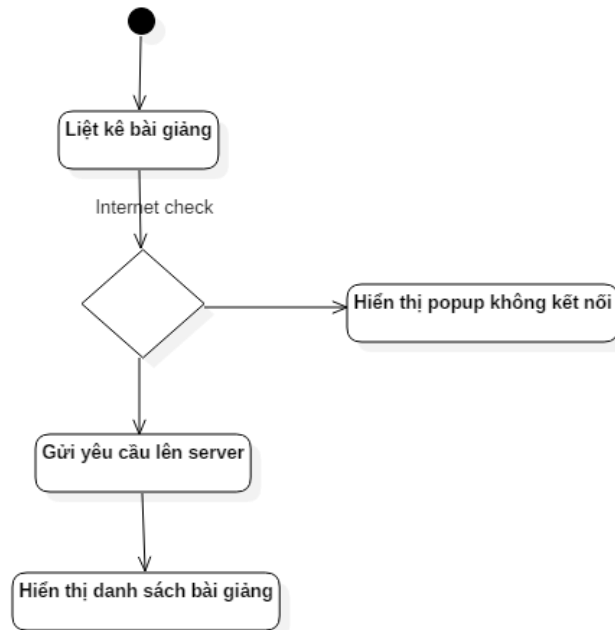
Biểu đồ hoạt động trên mô tả quá trình đăng kí tài khoản học, tương tự như trường hợp đăng nhập, kết quả của quá trình đăng kí sẽ được xử lý bởi webservice, sau đó ứng dụng sẽ hiển thị thông báo tùy theo kết quả trả về.

### 3.2.3.2.3 Liệt kê bài giảng



**Hình 32 Sequence Liệt kê bài giảng**

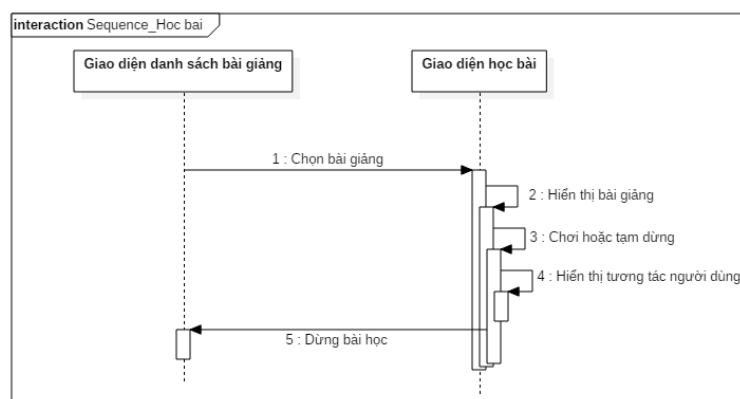
Sau khi đăng nhập, học viên có thể yêu cầu liệt kê bài giảng hiện có trên server. Yêu cầu liệt kê được gửi lên, server truy vấn cơ sở dữ liệu để lấy ra danh sách bài giảng hiện có và trả về cho học viên, Danh sách này được hiển thị trên màn hình ứng dụng, từ đó học viên có thể chọn bài giảng mong muốn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



**Hình 33 Activity Liệt kê bài giảng**

#### 3.2.3.2.4. Học bài

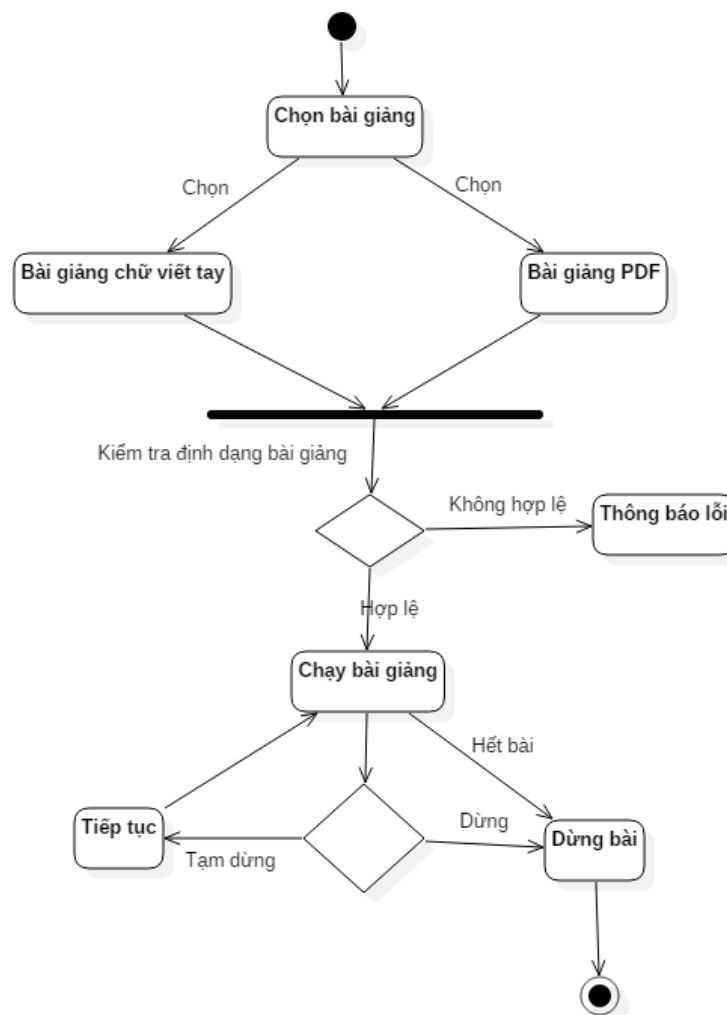
Hình bên dưới là biểu đồ tuần tự hành động học bài của học viên



**Hình 34 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 ngoại tuyến.



**Hình 35 Activity Học ngoại tuyến**

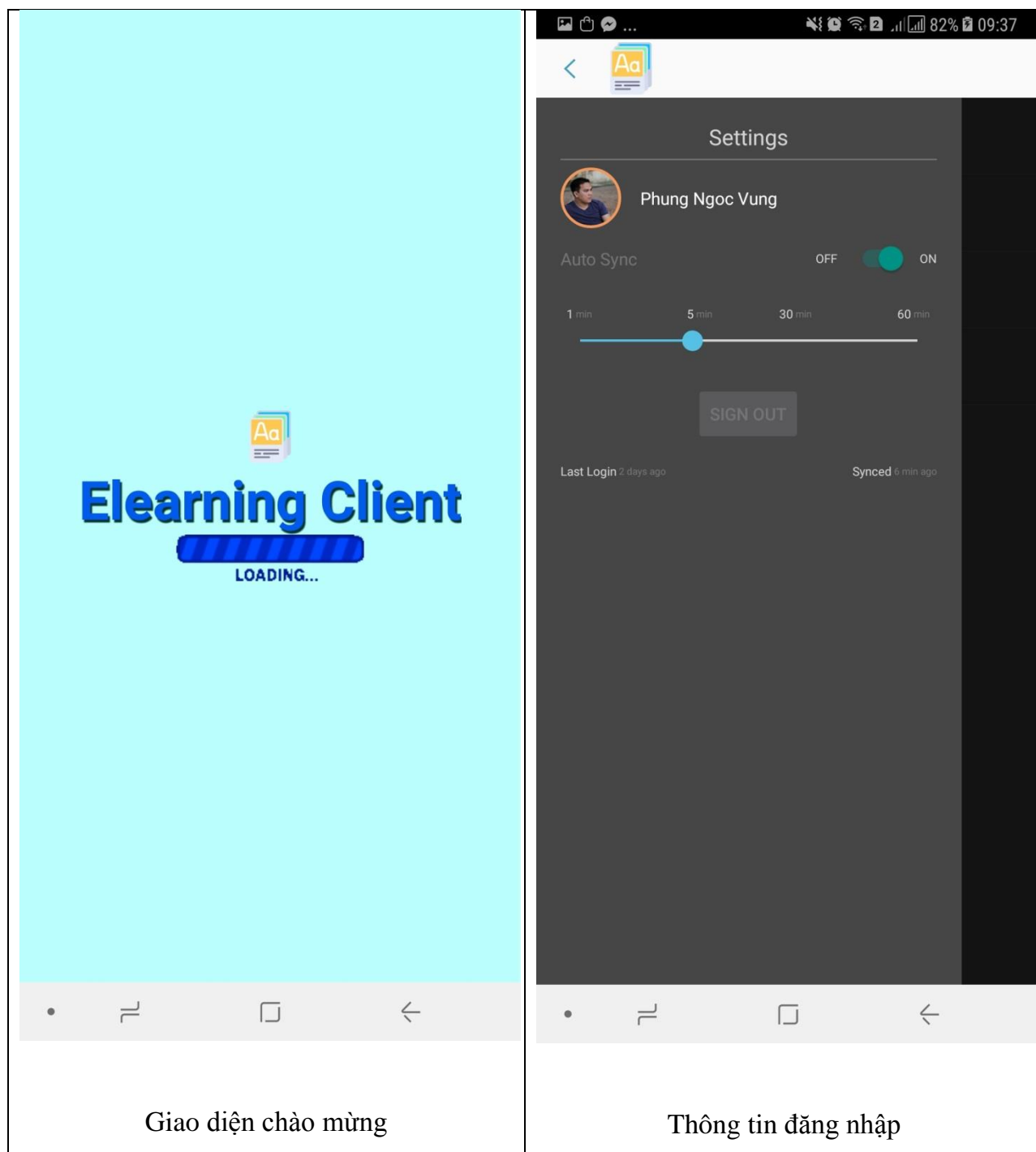
### **3.2.3.3 Thiết kế giao diện**

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.

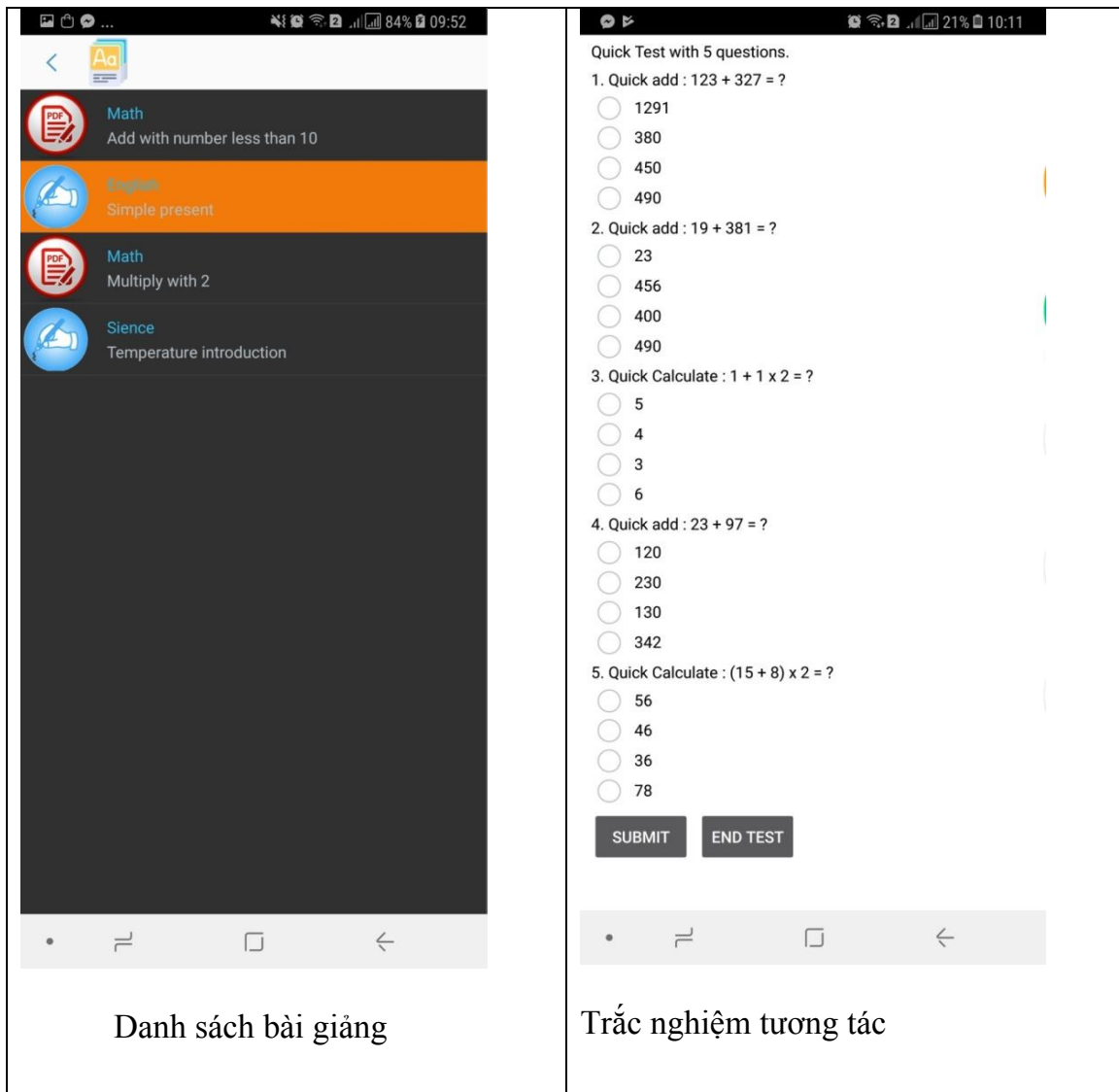
Hình bên dưới là màn hình chào mừng của ứng dụng và thông tin đăng nhập của học viên.





Hình trái là màn hình chào mừng. Trong quá trình loading, các thành phần giao diện được tạo ra. Màn hình bên phải hiển thị thông tin đăng nhập của học viên bao gồm tên, hình ảnh. Ngoài ra có một trục thời gian mô tả cài đặt đồng bộ bài giảng giữa ứng dụng và server. Qua thời gian đồng bộ, ứng dụng sẽ liệt kê lại các bài giảng đang có trên server để đảm bảo danh sách bài giảng đang hiển thị trên ứng dụng là cập nhật nhất.

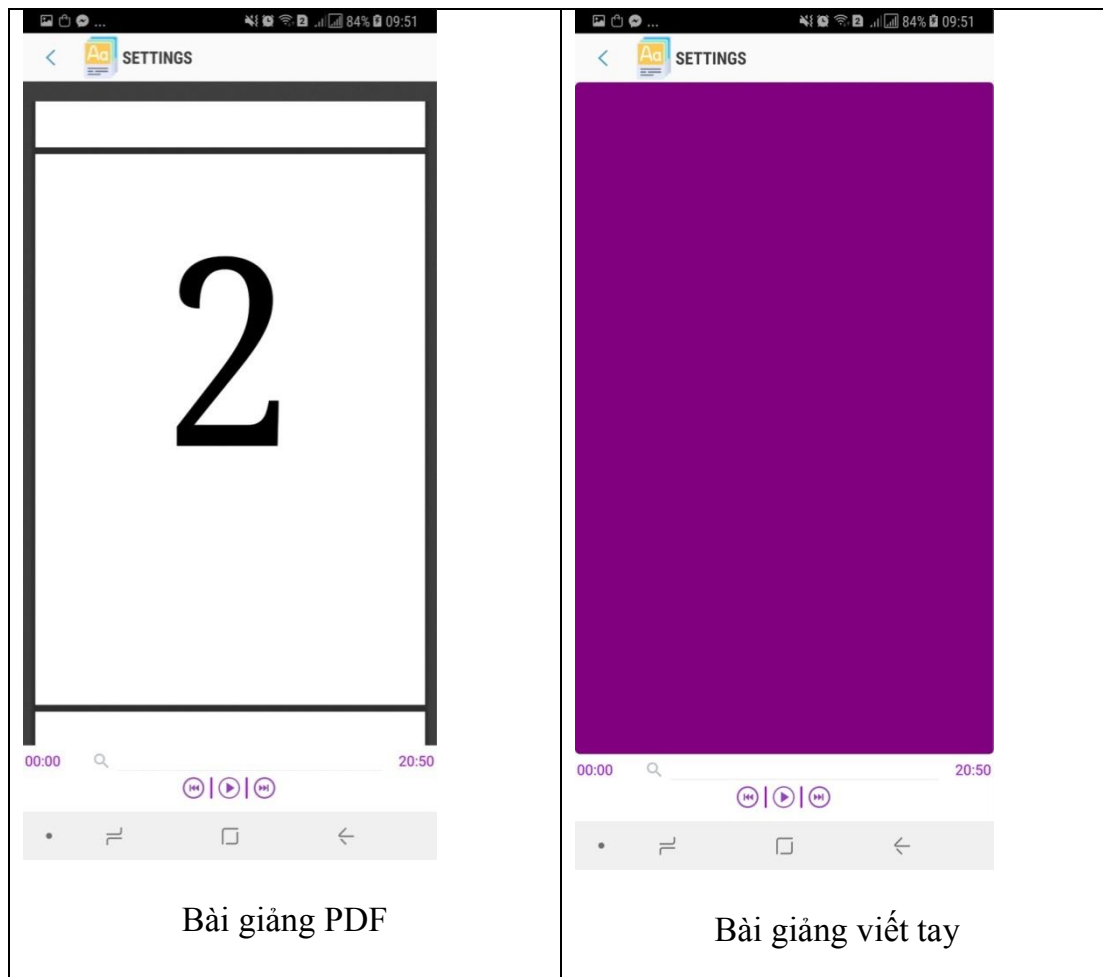
Hai hình bên dưới hiển thị danh sách bài giảng hiện có trên server và giao diện bài trắc nghiệm tương tác



Bằng việc kết nối lên server thông qua webservice, ứng dụng sẽ liệt kê bài giảng hiện có trên server và hiển thị theo dạng danh sách. Hành động này sẽ được định kỳ thực hiện lại thông qua cài đặt đồng bộ bài giảng của người dùng.

Bài trắc nghiệm nhanh bao gồm danh sách các câu hỏi, gồm 4 đáp án. Bài này là nhân tố tương tác giữa bài giảng và học viên. Trong quá trình học, bài trắc nghiệm sẽ được hiển thị ra một cách bất kỳ tùy theo sắp xếp của người thầy. Bài giảng sẽ được dừng lại ngay lập tức để học sinh có thể làm bài trắc nghiệm. Sau khi bài trắc nghiệm được hoàn thành, bài giảng sẽ được tiếp tục

Hai hình bên dưới là giao diện bài giảng: PDF và chữ viết tay.



Bài giảng PDF và chữ viết tay sẽ được hiển thị theo thực thời gian sắp xếp sẵn của người thầy. Bằng cách nhấn nút “Play”, bài giảng sẽ được bắt đầu, các trang PDF bắt đầu được hiển thị, dữ liệu chữ viết tay bắt đầu được render một cách đồng bộ với dữ liệu âm thanh.

### **3.2.4. Một số kịch bản kiểm thử và kết quả**

#### **3.2.4.1. Yêu cầu kiểm thử**

**Bảng 4 Các yêu cầu kiểm thử**

STT	Yêu cầu kiểm thử	Yêu cầu kết quả	Kết quả
1	Hiện thị màn hình 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 thông suốt, đường truyền ổn định, lag ít	Cần cải tiến

Các kịch bản kiểm thử được thực hiện một cách độc lập với quá trình phát triển. Các tính năng được kiểm thử nhiều lần rồi mới đưa đến kết quả. Các chức năng được đánh dấu “PASS” nghĩa là đã vượt qua các tình huống kiểm thử với tỉ lệ thành công 100%, các tính năng được ghi chú “Cần cải tiến” thì không vượt qua được hết các yêu cầu đặt ra, các tính năng này cần được tiếp tục cải tiến.

### 3.2.4.2. Tình huống kiểm thử

**Bảng 5 Các 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

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 hành 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 nhằm 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

## ***TÀI LIỆU THAM KHẢO***

- [1] Trịnh Văn Biểu (2012) , “ *Một số vấn đề về đào tạo trực tuyến E-Learning*”,  
<http://www.vjol.info/index.php/sphcm/article/viewFile/12799/11687>
- [2] Lan Umek, Damijana Keržič, Nina Tomažević and Aleksander Aristovnik (2015),  
*"Moodle E-learning system and students' performance in higher education: the case of public administration programmes"*, University of Ljubljana
- [3] Microsoft Coporation (2016), “*Microsoft platform and tools for mobile application development* “
- [4] Jared Dickson (2013), “*Xamarin Mobile Development*”, Grand Valley State University ScholarWorks@GVSU
- [5] Xamarin Corp (2016) , “*Xamarin Key Strategies for Mobile Excellence*”
- [6] Amer A. Radi (6-2016), “*Evaluation of Xamarin Forms for MultiPlatform*”, Grand Valley State University ScholarWorks@GVSU
- [7] Xamarin Coporation, *Cross-Platform User Interfaces with Xamarin.Form*  
<https://developer.xamarin.com/guides/xamarin-forms>
- [8] Charles Petzold (2016), *Cross-platform C# programming for iOS, Android, and Windows*
- [9] Trần Minh Triết, Lê Trần Lộc và Nguyễn Thành Đạt (4-2016), *Các hướng phát triển ứng dụng đa nền tảng với Xamarin*  
[http://minhtriettran.blogspot.com/2016/04/xamarin-crossplatform-development\\_22.html](http://minhtriettran.blogspot.com/2016/04/xamarin-crossplatform-development_22.html)
- [10] Skia (2012), *Skia Graphics Library*, <https://skia.org/>
- [11] PDF.js , *A general-purpose, web standards-based platform for parsing and rendering PDFs*, <https://mozilla.github.io/pdf.js/>
- [12] Xamarin Coporation, *Xamarin + Visual Studio*.  
<https://developer.xamarin.com/guides/cross-platform/windows/visual-studio/>

[13] James Montemagno (2016) , *Creating Mobile Apps with Xamarin.Forms*