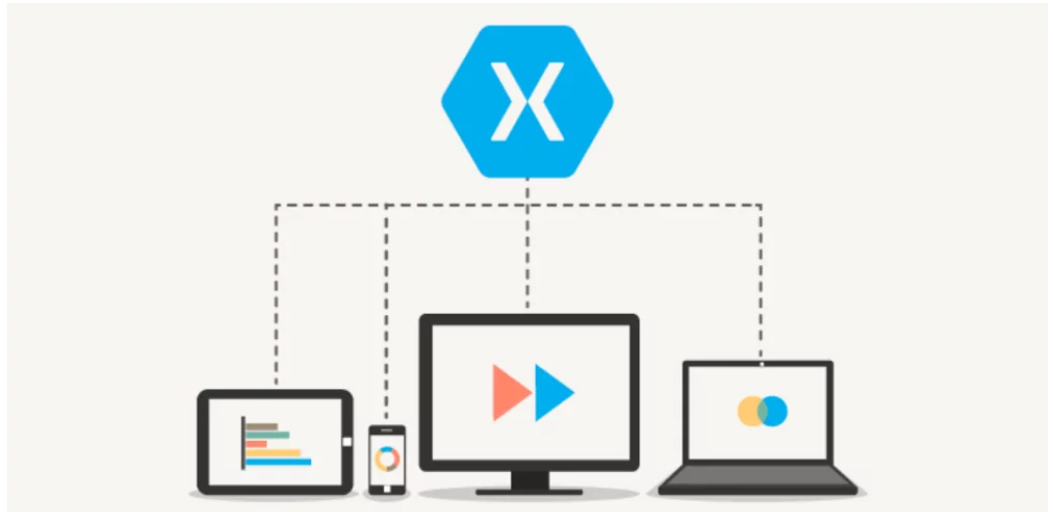
**NOTE: ĐÂY CHỈ LÀ XƯỜNG ĐỂ SAU KHI APP HOÀN THIỆT MỚI CHỈNH SỬA LẠI ĐỂ DƯA VÀO FILE BÁO CÁO CHÍNH**

1. **Sơ lược về Xamrin**



Xamarin nền tảng cách mạng hóa phát triển di động – Nguồn: cameoplus.com

1. **Xamarin là gì?**

**Xamarin** là nền tảng hỗ trợ phát triển các ứng dụng di động được **Microsoft bảo trợ**. Xamarin đề xuất cách tiếp cận phát triển các app đi động theo cách “viết một lần cho nhiều nền tảng”. Xamarin sử dụng ngôn ngữ chính C# để phát triền và tương thích hoàn toàn với hệ sinh thái **Azure Cloud**. Nhờ đó, các **lập trình viên .net** có thể dễ dàng tiếp cận nhanh Xamarin khi cần xây dựng ứng dụng.

1. **Lịch sử hình thành Xamarin**

**Xamarin** là một công ty phần mềm được thành lập vào năm 2011. Và gần đây vào năm 2016, nó đã được **Microsoft** mua lại. Xamarin cung cấp cho nhà phát triển các công cụ có thể giúp họ xây dựng các ứng dụng di động đa nền tảng. Các ứng dụng có thể có tất cả các tính năng gốc và cũng có thể chia sẻ cơ sở mã chung cùng một lúc. Theo thống kê của Xamarin, hơn 15000 công ty dựa vào các công cụ của họ và danh sách này bao gồm nhiều tên tuổi lớn nằm trong “**danh sách 500 công ty**”.

1. **Tính năng đáng giá khiến Xamarin được quan tâm**

**- Các công cụ hỗ trợ Xamarin** được tích hợp trong trình phát triển [**IDE Visual Studio**](http://cameoplus.com/microsoft-visual-studio-la-gi-tim-hieu-diem-khac-biet-giua-cac-phien-ban/) và bạn có thể trực tiếp tạo các **ứng dụng Android** , **iOS** và **Windows** từ chính **Visual Studio**. Hầu hết các mã phổ biến được viết bằng **C#**. Vì vậy, bạn không cần phải học **Java**, **Objective-C** hoặc **Swift** để xây dựng ứng dụng nếu bạn đã biết **C#**. Nếu bạn là người mới bắt đầu, thì việc *đi theo con đường phát triển Xamarin* thay vì quá trình học tập thông thường thực sự có thể dạy bạn phát triển ứng dụng cho nhiều hơn một nền tảng. Sử dụng Xamarin đẻ phát triển ngay từ đầu mà không cần tìm hiểu cấu trúc của từng nền tảng có khiến cho kiến thức của bạn bị thiếu hụt?

- Câu trả lời cho câu hỏi đó chủ yếu là **Không**. Bộ tính năng tuyệt vời được cung cấp bởi toàn bộ bộ công cụ đảm bảo bạn không bỏ lỡ bất kỳ chức năng nào trên từng nền tảng. Nhưng tại một số điểm, nơi bạn có thể muốn đào sâu, tích hợp chức năng riêng biệt đều được hỗ trợ. *Xamarin cho phép bạn gọi mã hiện có được viết bằng các ngôn ngữ dành riêng cho nền tảng khác như Java trong Android*. Nhưng đó chỉ là khi bạn đang xây dựng một cái gì đó rất cụ thể mà không thể thực hiện trên các nền tảng khác nhau.

- Xamarin cũng hỗ trợ **các thiết bị đeo tay**. Bạn cũng có thể xây dựng các ứng dụng gốc cho **Android Wear** và **Apple Watch**. Hệ thống cung cấp, quản lý plugin nuget được tích hợp trong Xamarin cho phép bạn thêm nhiều chức năng hơn vào ứng dụng của mình bằng cách tải xuống các plugin đơn giản. Bạn có thể dễ dàng tích hợp ứng dụng của mình với hầu hết các dịch vụ phụ trợ phổ biến như Google Firebase, Amazone AWS, Microsoft Azure, Parse, v.v. Bạn cũng có thể thêm các phương thức xác thực phổ biến. Ngoài ra, các plugin để thêm hỗ trợ thanh toán và các tính năng khác cũng có sẵn. Hầu hết các plugin phổ biến là đa nền tảng nhưng các plugin cụ thể cho nền tảng cũng có sẵn, ví dụ: plugin hỗ trợ thanh toán Google Play.

### **IVXamarin hoạt động như thế nào**



Cơ chế chia sẽ mã của Xamarin – Nguồng: cameoplus.com

- Đây là câu hỏi được hỏi và đặc biệt nhất về nền tảng này. Vì vậy, về cơ bản Xamarin làm gì mà bạn có thể viết một mã chung cho các nền tảng khác nhau? Ở cấp độ gốc, Xamarin đã chuyển đổi toàn bộ SDK Android và iOS hiện có sang C# để bạn có thể viết mã bằng ngôn ngữ quen thuộc hơn. Và vì bạn có thể sử dụng C# để mã cho cả hai nền tảng, bạn cần nhớ ít cú pháp hơn. Bạn có thể truy cập hầu hết mọi API iOS hoặc Android trong C# bằng các công cụ Xamarin.

- Bây giờ đến với giao diện người dùng của nó, giao diện người dùng vẫn giữ nguyên khá nhiều. Bạn cần xây dựng UI riêng cho các nền tảng khác nhau và sau đó liên kết UI với cơ sở mã chung. Dưới đây là hình ảnh từ trang web Xamarin mà tôi muốn chia sẻ để bạn có thể hiểu rõ hơn những gì đang xảy ra dưới mui xe.

- Vì vậy, bạn có thể xây dựng giao diện người dùng ứng dụng gốc thực tế. Các giao diện người dùng như vậy không chỉ cung cấp cho người dùng cảm giác cần thiết mà cả các ứng dụng hoạt động như thể chúng là các ứng dụng gốc thông thường. Bên dưới mã UI cụ thể của nền tảng, là mã C# được chia sẻ gọi mã cơ sở chung.

- Thực tế có hai cách khác nhau để xây dựng giao diện người dùng. Bạn có thể sử dụng các phương thức gốc ban đầu để xây dựng giao diện người dùng hoặc bạn có thể sử dụng **Xamarin.Forms**. Biểu mẫu cho phép bạn xây dựng giao diện người dùng cho các nền tảng khác nhau cùng một lúc. Gần như có 100% chia sẻ mã nếu bạn quyết định chọn Biểu mẫu trên Công nghệ UI gốc.

- Sau khi thực hiện UI là phần khó nhất mà bạn cần kết nối UI với codebase. Và điều đó một lần nữa có thể được thực hiện bằng hai chiến lược chia sẻ mã khác nhau được gọi là ‘**Shared Project**‘ hoặc ‘**Portable Class Libraries**‘.

**B. Ứng dụng**

**I. Nguyên nhan hướng phát triển:**

- Hệ thống sổ liên lạc đã có trên web nhưng

+ Việc truy cập vào web có thể gây khó khăn cho 1 số nhỏ PHHS.

+ Với xu hướng smartphone đang phát triển và đón đầu việc này USTeam quyết định phát triển hệ thống lên nền tảng mobile. (Theo công bố của World Advertising Research Center (WARC), vào năm 2025 sẽ có khoảng 72,6 % người dùng trên thế giới sẽ truy cập Internet qua smartphone).

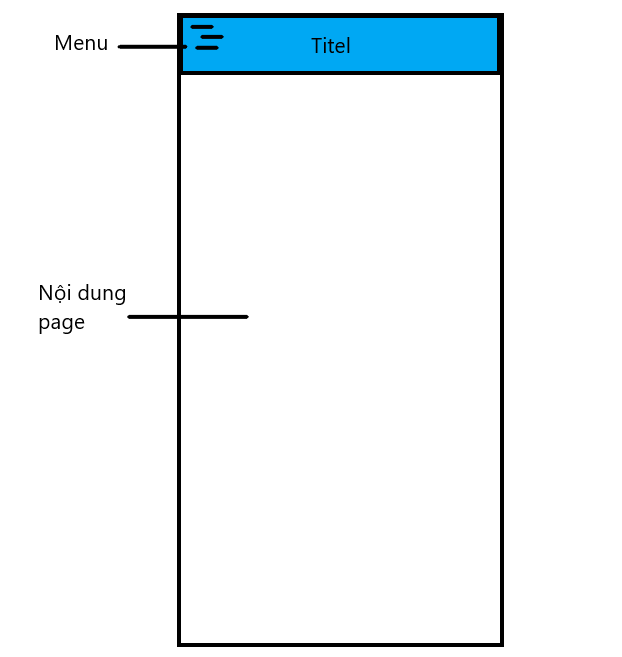
**II. Mục đích:**

- Với App mobile người dùng sẽ sử dụng dễ dàng hơn, linh hoạt hơn.

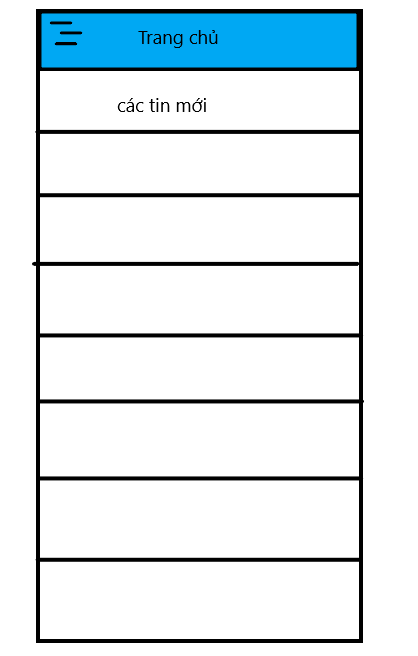
- Dựa vào thông báo trên mobile để thông báo về điểm chuyên cần hay thông báo mới của nhà trường giúp gia đình nắm bắt một cách tốt nhất về tình hình trường lớp của con mình.

III. Giao diện (ý tưởng)

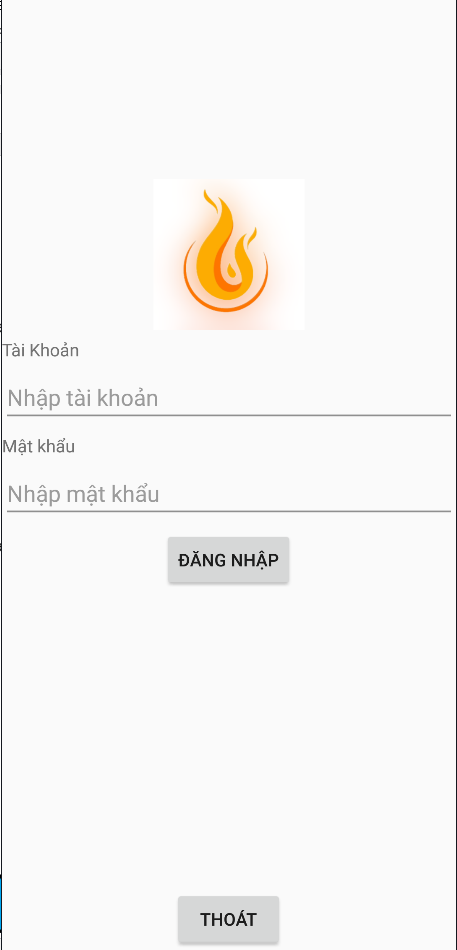
1. Giao diện tổng – trang chủ



1. Giao diện trang chủ



1. Dăng nhập



1. Đổi tarng chủ



1. Đổi pass

