

**ĐẠI HỌC QUỐC GIA THÀNH PHỐ HỒ CHÍ MINH**  
**TRƯỜNG ĐẠI HỌC KHOA HỌC TỰ NHIÊN**  
**KHOA: CÔNG NGHỆ THÔNG TIN**



**SEMINAR: MOBILE APP DEVELOPMENT**

**Giảng viên môn học** : TS. Đinh Bá Tiến

**Nhóm** : 9

**Học kì – Niên khóa** : HK1 – 2023-2024

**Thành viên nhóm**

**MSSV**

**Họ và tên**

21120066

Tô Tấn Hiệp

21120528

Nguyễn Tuấn Phong

21120478

Nguyễn Duy Khang

21120107

Nguyễn Minh Nhật

21120001

Nguyễn Ngọc Thiên Ân

## MỤC LỤC

<b>I.</b>	<b>GIỚI THIỆU .....</b>	<b>3</b>
1.1.	Định nghĩa Mobile App Development .....	3
1.2.	Lịch sử về Mobile App Development .....	3
<b>II.</b>	<b>QUÁ TRÌNH XÂY DỰNG MOBILE APP .....</b>	<b>3</b>
2.1.	Xác định mục tiêu chiến lược .....	3
2.2.	Phân tích và lên kế hoạch .....	3
2.3.	Thiết kế giao diện người dùng (UI/UX) .....	5
2.4.	Phát triển ứng dụng .....	5
2.5.	Kiểm thử ứng dụng .....	7
2.6.	Triển khai và bảo trì .....	10
<b>III.</b>	<b>PHÂN LOẠI .....</b>	<b>10</b>
<b>IV.</b>	<b>NGUỒN THAM KHẢO .....</b>	<b>11</b>

## I. Giới thiệu

### 1.1. Định nghĩa Moblie App Development.

Như chúng ta đã biết, tốc độ chuyển đổi số nhanh trong thời đại ngày nay đã dẫn đến sự ra đời và phát triển của nhiều công nghệ. Trong số đó đáng kể đến là “Mobile App Development” hay có tên gọi khác là “Phát triển Ứng dụng Di động”.

Là một quy trình tạo ra phần mềm cho thiết bị di động.

### 1.2. Lịch sử về Mobile App Development.

- Giai đoạn **Basic Phone**.
- Giai đoạn **Feature Phone**.
- Giai đoạn **Smart Phone**.

## II. Quá trình xây dựng Mobile App

### 2.1. Xác định mục tiêu chiến lược.

Quá trình phát triển ứng dụng di động bắt đầu bằng việc xây dựng một chiến lược để xác định cách doanh nghiệp của bạn có thể cải thiện thông qua ứng dụng di động.

Do mục tiêu của mỗi ứng dụng có thể khác nhau, vẫn có một ảnh hưởng cụ thể đối với chiến lược di động cần được xem xét trong quá trình phát triển. Một chiến lược được xác định rõ ràng nên giúp bạn giải quyết "tại sao" cho những sáng kiến phát triển di động của bạn.

### 2.2. Phân tích và lên kế hoạch.

Tại giai đoạn này, ý tưởng ứng dụng của bạn bắt đầu hình thành và trở thành một dự án thực tế. Quá trình phân tích và lập kế hoạch bắt đầu với việc xác định các trường hợp sử dụng và ghi lại các yêu cầu chức năng chi tiết.

Sau khi đã xác định các yêu cầu cho ứng dụng của mình, hãy chuẩn bị một lộ trình sản phẩm. Điều này bao gồm việc ưu tiên hóa các yêu cầu ứng dụng di động và nhóm chúng thành các cột mốc giao hàng. Bởi vì thời gian, tài nguyên và chi phí thường là mối quan tâm, hãy xác định sản phẩm tối thiểu có thể (MVP) và ưu tiên nó cho cuộc ra mắt ban đầu.

Một phần của giai đoạn lập kế hoạch trong quá trình phát triển ứng dụng bao gồm việc xác định kỹ năng cần thiết cho sáng kiến phát triển ứng dụng của bạn. Ví dụ, nền tảng di động iOS và Android sử dụng các công nghệ phát triển khác nhau. Nếu mục tiêu của bạn là xây dựng ứng dụng di động cho cả hai nền tảng iOS và Android, thì đội ngũ phát triển di động của bạn nên bao

gồm các nhà phát triển iOS và Android. Nếu phạm vi của ứng dụng di động của bạn ít phức tạp hơn và không yêu cầu các điều khiển cụ thể cho từng nền tảng, bạn có thể xem xét việc tạo ứng dụng di động bằng các công nghệ chung như React Native và Flutter.

### 2.3. Thiết kế giao diện người dùng.

Mục đích của thiết kế ứng dụng là tạo ra trải nghiệm người dùng mượt mà và dễ dàng với một giao diện được hoàn thiện.

Sự thành công của ứng dụng di động của bạn sẽ phụ thuộc vào việc người dùng áp dụng và hưởng lợi từ các tính năng của nó như thế nào. Mục tiêu của thiết kế UI và UX cho ứng dụng di động là tạo ra trải nghiệm người dùng xuất sắc, làm cho ứng dụng của bạn trở nên tương tác, dễ hiểu và thân thiện với người dùng. Trong khi thiết kế UI đẹp có thể giúp thu hút sự chú ý ban đầu, ứng dụng của bạn cần có trải nghiệm người dùng một cách tự nhiên để giữ cho người dùng ứng dụng tham gia.

**Information Architecture & Workflows** Bước đầu tiên của quá trình thiết kế ứng dụng di động của bạn là xác định dữ liệu mà ứng dụng sẽ hiển thị cho người dùng, dữ liệu nó sẽ thu thập, cách người dùng tương tác với sản phẩm hoàn thành và các hành trình của họ trong ứng dụng.

Đối với các công ty quan tâm đến phát triển ứng dụng di động doanh nghiệp, các giải pháp di động thường hỗ trợ người dùng với các vai trò và quyền khác nhau. Do đó, quan trọng để tích hợp những quy tắc này vào kiến trúc thông tin của ứng dụng của bạn. Sơ đồ quy trình giúp xác định mọi tương tác có thể của người dùng với ứng dụng và cấu trúc điều hướng của ứng dụng.

**Wireframes** Nhà thiết kế ứng dụng di động thường bắt đầu thiết kế ứng dụng với các bản vẽ trên giấy. Khung dây là phiên bản số của các bản vẽ đó. Khung dây là các bố cục khái niệm - còn được gọi là mô phỏng thấp - giúp tạo cấu trúc hình ảnh cho yêu cầu chức năng của ứng dụng của bạn.

Với khung dây, sự tập trung chủ yếu vào thẩm mỹ và trải nghiệm người dùng, chứ không phải là màu sắc và kiểu dáng. Việc tạo khung dây là một cách nhanh chóng và hiệu quả chi phí trong thiết kế ứng dụng di động. Trong khi tạo khung dây, bạn nên xem xét thiết kế UI cụ thể cho từng thiết bị. Do đó, không có ý nghĩa là ứng dụng của bạn có thể được sử dụng trên iPhone, iPad, hoặc điện thoại và máy tính bảng Android mà không gặp vấn đề.

**Style Guide** Hướng dẫn kiểu ghi chép các tiêu chuẩn thiết kế của ứng dụng của bạn, bao gồm bảng màu, kiểu chữ và đồ họa, theo hướng dẫn thương hiệu của công ty bạn.

Hướng dẫn kiểu đóng góp vào chiến lược thiết kế của ứng dụng. Việc thiết lập một hướng dẫn kiểu từ sớm như một phần của quá trình phát triển ứng dụng di động của bạn cải thiện năng suất của các nhà phát triển ứng dụng di động của bạn. Tuân thủ hướng dẫn kiểu sẽ giúp đảm bảo tính nhất quán trong cảm nhận và cảm nhận của ứng dụng của bạn. Là một phần của thiết kế ứng dụng của bạn, bạn cũng nên xem xét hướng dẫn thiết kế ứng dụng từ Apple cho ứng dụng iOS và từ Google cho ứng dụng Android.

**Mockups** Mô hình ảo, hay thiết kế chất lượng cao, là bản vẽ cuối cùng của thiết kế hình ảnh của ứng dụng của bạn. Mô hình ảo được tạo ra bằng cách áp dụng hướng dẫn kiểu của bạn vào khung dây ứng dụng. Khi thiết kế ứng dụng của bạn bắt đầu hoàn thiện, hãy mong đợi sự điều chỉnh thêm vào kiến trúc thông tin, quy trình làm việc và thẩm mỹ của nó. Figma, Invision và Adobe XD là những công cụ phổ biến nhất để tạo mô hình ảo chất lượng cao. Những công cụ này giúp việc hợp tác giữa đội ngũ thiết kế và đội ngũ kỹ thuật của ứng dụng của bạn trở nên mượt mà.

**Interactive Prototype** Trong khi mô hình ảo hiển thị chức năng của ứng dụng di động của bạn bằng các thiết kế tĩnh, bạn có thể tạo ra một nguyên mẫu tương tác từ thiết kế của mình bằng các công cụ như Invision và Figma. Nguyên mẫu mô phỏng trải nghiệm người dùng và cho thấy cách ứng dụng sẽ hoạt động như một sản phẩm hoàn thiện. Mặc dù việc phát triển nguyên mẫu có thể tốn thời gian, nhưng nỗ lực đó xứng đáng, vì chúng cung cấp kiểm tra ở giai đoạn đầu của thiết kế và chức năng của ứng dụng của bạn. Thường xuyên, nguyên mẫu giúp xác định các điều chỉnh cần thiết cho chức năng được đề xuất của ứng dụng - tiết kiệm thời gian và chi phí trong suốt quá trình phát triển ứng dụng di động của bạn.

Một số công ty ưa thích việc tạo ra nguyên mẫu tương tác ở giai đoạn khung dây, đặc biệt là khi các yêu cầu chức năng của ứng dụng phụ thuộc nhiều vào UI. Hoặc nếu có nhu cầu xem xét chức năng được đề xuất của ứng dụng với một nhóm tập trung.

### 2.4. Phát triển ứng dụng.

Lập kế hoạch vẫn là một phần quan trọng trong quá trình phát triển ứng dụng di động. Trước khi bắt đầu các công việc phát triển/lập trình thực tế, bạn sẽ cần phải:

1. Xác định kiến trúc kỹ thuật,
2. Chọn một ngăn xếp công nghệ và
3. Xác định các cột mốc phát triển.

Ngoài các điểm trên, nếu phạm vi của quá trình phát triển ứng dụng di động của bạn bao gồm các công nghệ như Ứng dụng Tăng Cường (AR) và Trí Tuệ Nhân Tạo (AI), bạn sẽ cần xác nhận rằng đội ngũ phát triển ứng dụng của bạn có khả năng thiết kế và triển khai chúng. Tiếp cận công nghệ như vậy sẽ giúp ứng dụng của bạn mang lại trải nghiệm người dùng độc đáo, phong phú, đặc biệt là làm cho ứng dụng của bạn nổi bật so với những ứng dụng khác trên thị trường và công ty của bạn so với đối thủ.

Một dự án ứng dụng di động điển hình bao gồm ba phần chính: công nghệ phía sau/server, API(s), và giao diện người dùng phía trước của ứng dụng di động.

**Công Nghệ Phía Sau/Server:** Bao gồm cơ sở dữ liệu và các đối tượng phía máy chủ cần thiết để hỗ trợ chức năng của ứng dụng di động. Nếu bạn đang sử dụng nền tảng phía sau hiện tại, bạn có thể cần sửa đổi để phù hợp với sản phẩm di động mới của bạn.

**API:** Giao Diện Lập Trình Ứng Dụng (API) là một phương thức giao tiếp giữa ứng dụng và máy chủ/cơ sở dữ liệu phía sau. Sử dụng kiến trúc dựa trên microservices với các tiêu chuẩn mã hóa phù hợp cho phép các công ty tạo ra một giao diện có thể mở rộng và an toàn để trao đổi dữ liệu giữa phía trước và phía sau của ứng dụng di động.

**Giao Diện Người Dùng Phía Trước của Ứng Dụng Di Động:** Phía trước là ứng dụng di động native mà người dùng cuối cài đặt và tương tác trên thiết bị di động của họ. Trong hầu hết các trường hợp, ứng dụng di động bao gồm trải nghiệm người dùng tương tác dựa trên dữ liệu thời gian thực và yêu cầu kết nối mạng giữa phía trước và phía sau của ứng dụng. Trong một số trường hợp, ứng dụng có thể yêu cầu hoạt động offline và sử dụng lưu trữ dữ liệu trên thiết bị di động.

Có bốn phương pháp phát triển chính khi xây dựng ứng dụng di động.

1. **Ứng dụng di động native:** được viết bằng ngôn ngữ và framework của nền tảng và chạy trực tiếp trên hệ điều hành của thiết bị như iOS và Android. Cung cấp hiệu suất và trải nghiệm người dùng tốt nhất.
2. **Ứng dụng di động native đa nền tảng:** có thể được viết bằng nhiều ngôn ngữ lập trình và framework, sau đó biên dịch thành một ứng dụng native chạy trên hệ điều hành của thiết bị. Phù hợp cho các ứng dụng di động đơn giản không yêu cầu các tính năng cụ thể của thiết bị và không cần phải cập nhật ngay khi các framework hệ điều hành mới được phát hành.
3. **Ứng dụng di động kết hợp:** được xây dựng bằng các công nghệ web tiêu chuẩn, bao gồm JavaScript, CSS và HTML5, sau đó được đóng gói dưới dạng gói cài đặt ứng dụng. Một container web cung cấp môi trường chạy trình duyệt và một cầu nối cho các API thiết bị native bằng Apache Cordova. Phù hợp cho các công ty muốn sử dụng lại các ứng dụng web hiện tại và có ngân sách vừa phải.
4. **Ứng dụng web tiến bộ:** cung cấp một cách tiếp cận thay thế cho phát triển ứng dụng di động truyền thống, bỏ qua cửa hàng ứng dụng và cài đặt ứng dụng. Đây là các ứng dụng web sử dụng khả năng của trình duyệt bao gồm làm việc ngoại tuyến, chạy quy trình nền và thêm liên kết vào màn hình chính của thiết bị. Tạo ra trải nghiệm người dùng giống như 'ứng dụng' và là lựa chọn tốt cho các ứng dụng có chức năng hạn chế và yêu cầu làm việc ngoại tuyến.

Quá trình phát triển ứng dụng di động có thể yêu cầu tích hợp với các API bên ngoài, cho phép họ nhanh chóng tích hợp, sử dụng và tiêu thụ các khả năng phổ biến vào ứng dụng của họ.

Các khả năng này bao gồm:

1. Dịch vụ xác thực đơn và đa yếu tố cho việc đăng ký/đăng nhập người dùng.
2. Thông báo đẩy.
3. Xử lý thanh toán.
4. Theo dõi vị trí.
5. Phân tích để theo dõi sự tương tác của người dùng.
6. Tích hợp mạng xã hội.
7. Truy cập đa phương tiện.

8. Lưu trữ dữ liệu trên đám mây.
9. Tích hợp chat.
10. Kết nối CRM.
11. Lỗi thoại và bot trò chuyện.

Công nghệ di động phát triển nhanh chóng với các phiên bản mới của các nền tảng di động. Hơn nữa, các thiết bị di động mới được phát hành mỗi vài tháng. Với các nền tảng và thiết bị thay đổi nhanh chóng, tính linh hoạt là quan trọng để ra mắt ứng dụng di động trong khung thời gian và ngân sách. Xác định các cột mốc phát triển như là một phần của kế hoạch phát triển linh hoạt hỗ trợ việc phát triển ứng dụng di động của bạn theo từng đợt. Nếu thời gian đưa sản phẩm ra thị trường là mối quan tâm hàng đầu, hãy sử dụng phương pháp phát triển linh hoạt.

Trong giai đoạn phát triển, các nhà phát triển ứng dụng di động có thể sử dụng giải pháp quản lý phiên bản như GitHub để quản lý và chia sẻ mã nguồn với các nhà phát triển ứng dụng khác trong đội. Trước khi bắt đầu phát triển di động, đội của bạn có thể xác định một quy trình quản lý mã nguồn và tạo các bản xây dựng ứng dụng để kiểm thử QA.

Sau mỗi cột mốc phát triển, ứng dụng di động được chuyển cho đội kiểm thử ứng dụng để xác thực. Bao gồm theo dõi sự cố và báo cáo lỗi qua các chương trình như Sentry và Crashlytics. Điều này sẽ giúp giải quyết sự cố kỹ thuật phát sinh trong quá trình sử dụng ứng dụng của bạn.

### 2.5. Kiểm thử ứng dụng.

Thực hiện quá trình kiểm thử chất lượng (QA) cẩn thận trong quá trình phát triển ứng dụng di động giúp ứng dụng trở nên ổn định, sử dụng được và an toàn. Để đảm bảo kiểm thử QA toàn diện cho ứng dụng di động của bạn, bạn cần chuẩn bị các trường hợp kiểm thử mà đề cập đến tất cả các khía cạnh của kiểm thử ứng dụng.

Tương tự như cách các trường hợp sử dụng định hình quá trình phát triển ứng dụng di động, các trường hợp kiểm thử định hình quá trình kiểm thử ứng dụng di động. Các trường hợp kiểm thử hướng dẫn đội của bạn thực hiện các bước kiểm thử, ghi lại kết quả kiểm thử để đánh giá chất lượng phần mềm và theo dõi sự sửa chữa để kiểm tra lại. Một phương pháp tốt là hòa nhập đội QA của bạn vào các giai đoạn Phân Tích và Thiết Kế. Sự hiểu biết của họ về các yêu cầu chức năng và mục tiêu của ứng dụng của bạn sẽ giúp tạo ra các trường hợp kiểm thử chính xác. Nhiều công ty cũng ưa thích phát triển kiểu dẫn động bằng kiểm thử và tự động hóa kiểm thử. Điều này đòi hỏi nỗ lực bổ sung trong việc phát triển và duy trì kiểm thử, và đội kỹ thuật của bạn sẽ phải xem xét ưu nhược điểm của việc triển khai tự động hóa kiểm thử.

Ứng dụng của bạn nên trải qua các phương pháp kiểm thử dưới đây để mang lại một giải pháp chất lượng.

**Kiểm Thử Trải Nghiệm Người Dùng:** Bước quan trọng trong kiểm thử ứng dụng di động là xác nhận rằng bản triển khai cuối cùng khớp với trải nghiệm người dùng được tạo ra bởi nhóm thiết kế ứng dụng. Hình ảnh, quy trình làm việc và tương tác là những gì sẽ tạo ấn tượng trực tiếp đối với người dùng cuối về ứng dụng của bạn. Hãy đảm bảo rằng ứng dụng của bạn sử dụng các



font nhất quán, bố cục phong cách, bảng màu và khoảng trắng giữa dữ liệu, thiết kế biểu tượng và điều hướng. Đảm bảo rằng ứng dụng của bạn tương thích với các hướng dẫn thiết kế ban đầu sẽ ảnh hưởng trực tiếp đến sự chấp nhận của người dùng.

**Kiểm Thử Chức Năng:** Độ chính xác của chức năng của ứng dụng di động là quan trọng đối với sự thành công của nó. Mặc dù khó có thể dự đoán được hành vi và kịch bản sử dụng của mỗi người dùng cuối, kiểm thử chức năng cho phép bạn xác định rằng gần như mọi hành vi có thể đã được chứng minh là hoạt động.

Chức năng của ứng dụng của bạn nên được kiểm thử bởi nhiều người dùng để bao phủ nhiều điều kiện kiểm thử tiềm năng nhất có thể. Bạn có thể ngạc nhiên khi phát hiện lỗi khi hai người dùng khác nhau kiểm thử cùng một tính năng, nhưng lại có kết quả khác nhau. Ví dụ, cả hai người dùng có thể điền vào cùng một biểu mẫu, nhưng cả hai đều có thể nhập dữ liệu khác nhau - điều này có thể dẫn đến việc phát hiện một lỗi.

Mục tiêu của kiểm thử chức năng là đảm bảo rằng người dùng có thể sử dụng các tính năng và chức năng của ứng dụng của bạn mà không gặp vấn đề nào. Nó có thể được chia nhỏ thành kiểm thử hệ thống (ứng dụng hoạt động như một toàn bộ) và kiểm thử đơn vị (các chức năng riêng lẻ của ứng dụng hoạt động đúng).

Nếu bạn đang xây dựng ứng dụng cho cả hai nền tảng di động iOS và Android, thì kiểm thử chức năng của bạn nên bao gồm so sánh tính năng giữa cả hai phiên bản của ứng dụng di động của bạn.

**Kiểm Thử Hiệu Năng:** Có nhiều tiêu chí định lượng để đo lường hiệu suất của ứng dụng của bạn.

- Làm thế nào ứng dụng của bạn phản ứng với yêu cầu của người dùng?
- Màn hình của ứng dụng load nhanh như thế nào?
- Ứng dụng của bạn có làm giảm pin điện thoại, hoặc gây rò rỉ bộ nhớ không?
- Ứng dụng của bạn có tận dụng băng thông mạng hiệu quả không?
- Kích thước của ứng dụng của bạn có lớn hơn cần thiết không?

Ngay cả khi ứng dụng của bạn đáp ứng các tiêu chí hiệu suất cơ bản, hãy kiểm tra ứng dụng, API và backend để xem xét tải bằng cách mô phỏng số lượng người dùng đồng thời tối đa. Ứng dụng của bạn nên có thể xử lý tải và hoạt động tốt ngay cả khi có đột ngột tăng sử dụng.

**Kiểm Thử Bảo Mật:** Bảo mật là mối quan tâm tối cao đối với các ứng dụng doanh nghiệp. Bất kỳ lỗ hổng tiềm ẩn nào có thể dẫn đến tình trạng tấn công. Nhiều công ty thuê các đơn vị bên ngoài để thực hiện kiểm thử bảo mật cẩn thận trên ứng dụng của họ. Tuy nhiên, đội QA và phát triển của bạn có thể thực hiện một số biện pháp đơn giản để giúp bảo vệ ứng dụng của bạn.

Bất kỳ API nào được ứng dụng của bạn sử dụng để tương tác với backend đều phải sử dụng giao thức "HTTPS" và bao gồm xác thực bổ sung cho quyền truy cập API. Điều này là một cách tiêu chuẩn để bảo vệ giao tiếp giữa ứng dụng và backend của nó. Lưu ý: bạn phải có một chứng chỉ SSL được cài đặt trên backend của bạn để công việc này hoạt động đúng cách.



Nếu ứng dụng của bạn yêu cầu người dùng đăng nhập, các phiên đăng nhập này nên được theo dõi trên thiết bị và backend. Các phiên người dùng nên bị chấm dứt bởi hệ thống khi người dùng không hoạt động trong một khoảng thời gian dài (thường là mười phút hoặc ít hơn trên một ứng dụng di động). Nếu ứng dụng của bạn lưu trữ thông tin đăng nhập của người dùng trên thiết bị để làm cho việc đăng nhập lại thuận tiện, bạn phải đảm bảo rằng nó đang sử dụng một dịch vụ đăng tin cậy. Ví dụ, iOS cung cấp tính năng Keychain có thể được sử dụng để lưu trữ chi tiết tài khoản của người dùng cho một ứng dụng cụ thể.

Biểu mẫu nhập dữ liệu trong ứng dụng di động của bạn nên được kiểm thử để đảm bảo không có rò rỉ dữ liệu.

**Kiểm Thử Thiết Bị và Nền Tảng:** Trung bình, các thiết bị di động mới nhập thị trường mỗi 12 tháng với phần cứng, firmware và thiết kế mới. Hệ điều hành di động được cập nhật mỗi vài tháng. Nhiều nhà sản xuất thiết bị di động như Samsung, LG, HTC, Motorola sử dụng nền tảng Android mã nguồn mở, nhưng họ tùy chỉnh nền tảng cho các thiết bị di động của họ, có kích thước và hình thức khác nhau.

So sánh với Apple, có môi trường kiểm soát hơn, vì họ sở hữu và quản lý cả phần cứng và hệ điều hành. Tuy nhiên, có nhiều thiết bị iPhone & iPad (Apple iOS) khác nhau trên thị trường.

Đây là nơi kiểm thử trong quá trình phát triển ứng dụng di động khác biệt đáng kể so với kiểm thử ứng dụng web. Bạn có thể kiểm thử ứng dụng web của mình chỉ trên trình duyệt Chrome trong môi trường Windows hoặc sử dụng dịch vụ như BrowserStack để kiểm thử trên nhiều trình duyệt và độ phân giải khác nhau. Nhưng ứng dụng di động của bạn phải được kiểm thử trên nhiều thiết bị di động hoặc bộ mô phỏng thiết bị để đảm bảo ứng dụng của bạn hoạt động mượt mà cho tất cả người dùng. Bạn cũng có thể tận dụng dịch vụ test farm để kiểm thử ứng dụng của bạn trên nhiều thiết bị, phiên bản hệ điều hành và mô phỏng người dùng đồng thời.

Nếu bạn đang xây dựng ứng dụng di động với các tính năng yêu cầu khả năng thực tế ảo và trí tuệ nhân tạo, bạn sẽ cần thiết bị hỗ trợ các công nghệ này. Điều này đòi hỏi điều kiện và môi trường kiểm thử cụ thể để xác nhận các chức năng này.

Kiểm thử ứng dụng di động trên tất cả các thiết bị di động, chi phí hỗ trợ liên tục và vấn đề quản lý thiết bị di động làm tăng phức tạp của ứng dụng di động. Đó là lý do chính tại sao nhiều công ty xây dựng ứng dụng doanh nghiệp của họ cho một nền tảng di động duy nhất và thường cung cấp thiết bị di động cho nhân viên của họ. Theo trải nghiệm của chúng tôi, hầu hết các công ty thường phát triển ứng dụng di động doanh nghiệp của họ trước với nền tảng di động iOS của Apple. Phiên bản Android thường là quan tâm phụ.

Kiểm thử là quan trọng đối với sự thành công của một ứng dụng. Nó chiếm một phần đáng kể của quá trình phát triển ứng dụng di động tổng thể của chúng tôi. Việc sở hữu và triển khai một chiến lược kiểm thử di động toàn diện là điều bắt buộc để cung cấp ứng dụng di động chất lượng.

Trong giai đoạn kiểm thử, có nhiều cách để phân phối bản phát triển của bạn cho các nhà kiểm thử. Phương pháp phổ biến nhất với ứng dụng iOS là sử dụng Testflight và đối với ứng dụng Android là cài đặt qua không dây (OTA) hoặc thông qua Google Play.

### 2.6. Triển khai và bảo trì.

Để phát hành ứng dụng di động native của bạn, bạn cần gửi nó đến cửa hàng ứng dụng phù hợp, Apple App Store cho ứng dụng iOS và Google Play cho ứng dụng Android. Tuy nhiên, trước khi ra mắt ứng dụng di động của bạn, bạn sẽ cần một tài khoản nhà phát triển với Apple App Store và Google Play Store.

Khi đóng gói ứng dụng của bạn để gửi đến cửa hàng ứng dụng, hãy sử dụng một tên và định danh gói khác nhau so với phiên bản QA của ứng dụng của bạn. Điều này sẽ giúp bạn phân biệt giữa các phiên bản QA và sản xuất của ứng dụng. Như vậy, bạn có thể tiếp tục kiểm thử các bản nâng cấp cho ứng dụng của mình trước khi phát hành chúng.

Việc phát hành ứng dụng trong cửa hàng ứng dụng đòi hỏi việc chuẩn bị siêu dữ liệu bao gồm:

- Tiêu đề của ứng dụng của bạn
- Mô tả ngắn
- Mô tả chi tiết
- Thẻ loại
- Từ khoá
- Biểu tượng khởi chạy
- Ảnh chụp màn hình của cửa hàng ứng dụng
- Đồ họa banner
- Video quảng cáo

Nếu ứng dụng của bạn yêu cầu người dùng đăng nhập, bạn sẽ cần cung cấp một tài khoản người dùng thử nghiệm làm phần của quy trình gửi. Sau khi gửi, ứng dụng sẽ trải qua quy trình kiểm tra của cửa hàng ứng dụng. Quá trình này có thể mất vài ngày, tùy thuộc vào chất lượng của ứng dụng của bạn và mức độ tuân thủ các hướng dẫn của cửa hàng ứng dụng. Nếu bị từ chối, bạn sẽ nhận được thông báo với lý do từ chối. Sau khi sửa lỗi, bạn có thể gửi lại ứng dụng và trải qua kiểm tra bổ sung cho đến khi ứng dụng được chấp nhận và xuất bản trong cửa hàng ứng dụng.

## III. Phân loại.

- Native Application.
- Web Application.
- Hybrid Application.
- Cross-Platform Frameworks.

## IV. Nguồn tham khảo.

1. <https://www.shutterstock.com/search/smartphone-addiction-infographic>
2. <https://www.youtube.com/watch?v=DsIviEKZad0>
3. <https://skynet-software.com/phat-trien-ung-dung-di-dong-co-hoi-de-cho-doanh-nghiep-phat-trien-hoat-dong-kinh-doanh/>
4. <https://www.invonto.com/insights/mobile-app-development-process/>
5. <https://amela.vn/phan-biet-native-app-web-app-hybrid-app-nen-su-dung-app-nao/>
6. <https://www.geeksforgeeks.org/top-mobile-app-development-platforms/>
7. <https://viblo.asia/p/phan-loai-ung-dung-di-dong-YWOZrE2YKQ0>
8. <https://viblo.asia/p/tong-quan-ve-react-native-gwd43kOr4X9>
9. <https://www.altexsoft.com/blog/pros-and-cons-of-flutter-app-development/>
10. <https://aws.amazon.com/vi/what-is/flutter/>
11. <https://ionic.io/resources/articles/what-is-ionic>
12. <https://www.altexsoft.com/blog/pros-and-cons-of-xamarin-vs-native/>
13. <https://www.javatpoint.com/what-is-xamarin>
14. <https://cordova.apache.org/>
15. [https://www.tutorialspoint.com/cordova/cordova\\_overview.htm](https://www.tutorialspoint.com/cordova/cordova_overview.htm)

HẾT