

Họ tên: Đinh Thanh Phong

MSSV: 2180607865

Câu 1:

Các nền tảng cho thiết bị di động thông minh hiện nay:

1. iOS

- **Đặc điểm:** Là hệ điều hành độc quyền của Apple, sử dụng trên các thiết bị như iPhone, iPad, và iPod Touch.
- **Ưu điểm:**
 - Bảo mật cao, ít bị tấn công bởi phần mềm độc hại.
 - Hệ sinh thái khép kín, đồng bộ giữa các thiết bị Apple, tạo trải nghiệm liền mạch.
 - Hiệu năng mượt mà và cập nhật nhanh chóng.
- **Khuyết điểm:**
 - Hạn chế trong việc tùy biến giao diện và tính năng.
 - Chỉ có thể sử dụng trên các thiết bị Apple, giá thành cao.

2. Android

- **Đặc điểm:** Là hệ điều hành mã nguồn mở do Google phát triển, sử dụng rộng rãi trên nhiều thiết bị của các hãng như Samsung, Xiaomi, Oppo.
- **Ưu điểm:**
 - Tính tùy biến cao, cho phép cá nhân hóa trải nghiệm người dùng.
 - Đa dạng thiết bị với các phân khúc giá khác nhau.
 - Hệ sinh thái Google phong phú.
- **Khuyết điểm:**
 - Vấn đề phân mảnh: Nhiều thiết bị và phiên bản hệ điều hành dẫn đến khó khăn trong việc tối ưu và cập nhật.
 - Ít an toàn hơn so với iOS do tính mở.

3. HarmonyOS

- **Đặc điểm:** Là hệ điều hành do Huawei phát triển nhằm thay thế Android, được dùng trên các thiết bị Huawei và Honor.
- **Ưu điểm:**
 - Khả năng tích hợp và kết nối đa thiết bị.
 - Hệ sinh thái Huawei cho phép đồng bộ và chia sẻ dữ liệu dễ dàng.
- **Khuyết điểm:**
 - Ít ứng dụng và chưa phổ biến ngoài hệ sinh thái Huawei.
 - Thiếu sự hỗ trợ từ các nhà phát triển ứng dụng quốc tế.

4. KaiOS

- **Đặc điểm:** Một hệ điều hành nhẹ, dùng trên các thiết bị feature phone (điện thoại thông minh cơ bản) và chủ yếu nhắm vào thị trường mới nổi.
- **Ưu điểm:**
 - Yêu cầu phần cứng thấp, giúp các thiết bị giá rẻ có thể truy cập Internet và các ứng dụng cơ bản.
 - Giao diện thân thiện và dễ sử dụng.
- **Khuyết điểm:**
 - Hạn chế về số lượng ứng dụng.
 - Khả năng mở rộng và phát triển tính năng không cao.

Câu 2:

Các nền tảng phát triển ứng dụng di động phổ biến hiện nay và so sánh sự khác biệt chính giữa chúng:

- **Native Development (Swift cho iOS, Kotlin cho Android)**
 - Phát triển ứng dụng trực tiếp trên nền tảng iOS hoặc Android với ngôn ngữ gốc.
 - **Ưu điểm:** Tối ưu hóa hiệu suất, truy cập đầy đủ các tính năng của hệ điều hành.
 - **Nhược điểm:** Tốn nhiều thời gian và chi phí vì phải phát triển riêng cho từng nền tảng.
- **React Native**

- Dùng JavaScript và framework React để tạo ứng dụng chạy trên cả iOS và Android.
- **Ưu điểm:** Tốc độ phát triển nhanh, có thể dùng lại code cho nhiều nền tảng.
- **Nhược điểm:** Hiệu suất không bằng ứng dụng native, một số tính năng cần viết code native bổ sung.
- **Flutter**
 - Dùng ngôn ngữ Dart và framework Flutter để phát triển ứng dụng đa nền tảng.
 - **Ưu điểm:** Hiệu suất gần như native, giao diện linh hoạt, có nhiều công cụ UI.
 - **Nhược điểm:** Thư viện chưa phong phú bằng các nền tảng khác, cộng đồng còn nhỏ hơn React Native.
- **Xamarin**
 - Sử dụng ngôn ngữ C# và .NET để phát triển ứng dụng đa nền tảng cho iOS, Android, và Windows.
 - **Ưu điểm:** Tích hợp tốt với hệ sinh thái Microsoft, hiệu suất cao hơn so với các framework JavaScript.
 - **Nhược điểm:** Ứng dụng có dung lượng lớn, cộng đồng ít phổ biến hơn Flutter và React Native.
- **Ionic**
 - Dựa trên HTML, CSS và JavaScript (kết hợp với Angular hoặc React) để xây dựng ứng dụng di động đa nền tảng.
 - **Ưu điểm:** Phát triển nhanh, giao diện đa dạng, có thể tích hợp với các web app.
 - **Nhược điểm:** Hiệu suất kém hơn các nền tảng khác, thiếu các tính năng native mạnh.

Câu 3:

Điều gì làm cho Flutter trở thành một lựa chọn phổ biến cho việc phát triển ứng dụng đa nền tảng?

- **Hiệu suất cao:** Flutter sử dụng công cụ đồ họa riêng (Skia), cho phép render giao diện trực tiếp và đạt hiệu suất gần với ứng dụng native.

- **Khả năng tạo UI tùy chỉnh:** Flutter cung cấp nhiều widget giúp tạo ra giao diện đồng nhất trên cả iOS và Android với độ tùy chỉnh cao.
- **Cộng đồng đang phát triển:** Flutter được Google hỗ trợ và phát triển mạnh mẽ, với nhiều công cụ và thư viện liên tục cập nhật.

So sánh với các nền tảng khác:

- **React Native:** React Native có cộng đồng lớn và dễ học đối với những người biết JavaScript. Tuy nhiên, React Native có hiệu suất kém hơn Flutter và phụ thuộc vào các bridge để tương tác với thành phần native, làm giảm hiệu năng.
- **Xamarin:** Xamarin hỗ trợ tốt hơn cho những người sử dụng hệ sinh thái Microsoft. Nhưng ứng dụng tạo bằng Xamarin thường có dung lượng lớn và cộng đồng không đa dạng bằng Flutter hay React Native.

Câu 4:

Các ngôn ngữ lập trình chính được sử dụng để phát triển ứng dụng trên Android và giải thích lý do chọn chúng:

- **Kotlin:**
 - Được Google chính thức hỗ trợ cho Android, Kotlin có cú pháp đơn giản, giúp giảm thiểu lỗi và tối ưu hóa mã nguồn.
 - Tương thích ngược với Java, nên dễ dàng kết hợp với các dự án Java cũ.
- **Java:**
 - Ngôn ngữ truyền thống cho Android, với cộng đồng và tài liệu phong phú.
 - Khả năng bảo trì tốt và tương thích với các thư viện, framework lâu đời.
- **Dart (Flutter):**
 - Được sử dụng khi phát triển ứng dụng Android bằng Flutter, giúp viết code một lần và chạy trên nhiều nền tảng.
 - Hiệu suất cao và dễ tùy chỉnh giao diện người dùng.
- **C++:**
 - Dùng trong các phần yêu cầu hiệu suất cao, ví dụ như xử lý hình ảnh, âm thanh, hoặc các tác vụ phức tạp.
 - Tích hợp với Android thông qua Native Development Kit (NDK).

Câu 5:

Các ngôn ngữ lập trình chính được sử dụng để phát triển ứng dụng trên iOS gồm:

- **Swift:** Ngôn ngữ chính và phổ biến nhất để phát triển ứng dụng iOS. Swift có cú pháp hiện đại, dễ học, nhanh chóng và mạnh mẽ, giúp tối ưu hóa hiệu suất ứng dụng trên iOS.
- **Objective-C:** Ngôn ngữ lâu đời của Apple, hiện vẫn được sử dụng nhưng dần bị thay thế bởi Swift. Một số dự án hoặc thư viện cũ vẫn duy trì và phát triển bằng Objective-C.
- **JavaScript** (kết hợp với **React Native**): Dùng để xây dựng các ứng dụng đa nền tảng, có thể chạy trên cả iOS và Android. React Native cho phép phát triển ứng dụng iOS bằng JavaScript.

Ngoài ra, một số ngôn ngữ và công cụ khác như **Dart** (Flutter), **C#** (Xamarin) cũng có thể được sử dụng cho các ứng dụng đa nền tảng trên iOS.

Câu 6:

Những thách thức mà Windows Phone đã phải đối mặt và nguyên nhân dẫn đến sự sụt giảm thị phần của nó:

1. **Thiếu sự hỗ trợ từ nhà phát triển ứng dụng:** Windows Phone không thu hút được đủ lượng nhà phát triển ứng dụng vì thị phần thấp và chi phí duy trì ứng dụng cao. Kho ứng dụng của Windows Phone thiếu rất nhiều ứng dụng phổ biến so với iOS và Android.
2. **Cạnh tranh khốc liệt với iOS và Android:** iOS và Android đã chiếm lĩnh thị trường với hệ sinh thái ứng dụng phong phú, nền tảng ổn định và có sự gắn kết cao với người dùng.
3. **Hạn chế trong việc tiếp cận người dùng:** Chiến lược tiếp thị không đủ mạnh và Windows Phone chỉ hỗ trợ một số dòng thiết bị nhất định, khiến phạm vi người dùng bị thu hẹp so với đối thủ.
4. **Chậm trễ trong cải tiến và cập nhật tính năng:** So với iOS và Android, Windows Phone thường chậm trễ trong việc cập nhật và cải tiến nền tảng, làm người dùng không có nhiều động lực để gắn bó.

Câu 7:

Các ngôn ngữ và công cụ để phát triển ứng dụng web trên thiết bị di động:

- **HTML, CSS và JavaScript:** Các ngôn ngữ cơ bản để xây dựng web, thường kết hợp với các framework hoặc thư viện JavaScript.

- **React Native:** Một framework dùng JavaScript (React) để phát triển ứng dụng di động đa nền tảng (iOS và Android).
- **Flutter:** Sử dụng ngôn ngữ Dart, giúp phát triển ứng dụng di động đa nền tảng với hiệu suất cao, có thể chạy trên cả iOS, Android và web.
- **Ionic:** Framework đa nền tảng dựa trên HTML, CSS và JavaScript (kết hợp Angular hoặc React) để xây dựng ứng dụng di động và web.
- **Progressive Web Apps (PWAs):** Ứng dụng web tiên tiến, có thể hoạt động như ứng dụng gốc trên thiết bị di động và có thể cài đặt từ trình duyệt.

Câu 8:

Nhu cầu nguồn nhân lực lập trình viên trên thiết bị di động hiện nay và những kỹ năng được yêu cầu nhiều nhất:

- **Nhu cầu nhân lực:** Hiện nay, các công ty công nghệ đều cần lập trình viên di động để phát triển các ứng dụng phục vụ người dùng trên smartphone. Lập trình viên di động có nhu cầu cao trong các lĩnh vực fintech, thương mại điện tử, giáo dục và chăm sóc sức khỏe.
- **Kỹ năng được yêu cầu:**
 - **Kỹ năng ngôn ngữ lập trình di động:** Năm vững Swift, Objective-C (cho iOS) và Kotlin, Java (cho Android) là một lợi thế lớn.
 - **Kinh nghiệm với các công nghệ đa nền tảng:** React Native, Flutter và Xamarin đang ngày càng được ưa chuộng để tối ưu hóa chi phí và thời gian phát triển ứng dụng.
 - **Kiến thức về UI/UX:** Thiết kế giao diện người dùng và trải nghiệm người dùng là kỹ năng cần thiết, vì các ứng dụng di động yêu cầu giao diện trực quan và thân thiện.
 - **Bảo mật di động:** Am hiểu các phương pháp bảo mật, mã hóa, và bảo vệ dữ liệu người dùng trên thiết bị di động.
 - **Kỹ năng làm việc nhóm và sử dụng công cụ quản lý dự án:** Kỹ năng làm việc với nhóm và sử dụng các công cụ như Git, Jira, hoặc Asana giúp tối ưu hóa quy trình phát triển.