

MSSV: 2180607674

Họ và tên: Võ Văn Kiệt

Lớp: 21DTHD5

Bài tập.

Câu 1: Hãy cho biết các nền tảng cho thiết bị di động thông minh hiện nay? Với mỗi nền tảng hãy cho biết đặc điểm, ưu và khuyết điểm.

Nền tảng	Đặc điểm	Ưu điểm	Khuyết điểm
Android	<ul style="list-style-type: none">– Mã nguồn mở: Cho phép các nhà phát triển tùy chỉnh và xây dựng các phiên bản Android riêng của họ.– Google Play Store: Nền tảng chính để phân phối và tải xuống các ứng dụng Android.– Đa dạng thiết bị: Hỗ trợ nhiều loại thiết bị từ nhiều nhà sản xuất khác nhau, dẫn đến sự phong phú về thiết kế và giá cả.– Tùy biến cao: Người dùng có thể tùy chỉnh giao diện, cài đặt các ứng dụng bên ngoài Google Play Store và thay đổi nhiều thiết lập hệ thống.	<ul style="list-style-type: none">– Sự đa dạng về thiết bị và giá cả, phù hợp với nhiều đối tượng người dùng.– Khả năng tùy biến và cài đặt ứng dụng linh hoạt.– Cộng đồng phát triển mạnh mẽ và phong phú	<ul style="list-style-type: none">– Sự phân mảnh: Nhiều phiên bản Android cùng tồn tại làm khó khăn trong việc cập nhật và hỗ trợ.– Bảo mật: Do tính mở và khả năng cài đặt ứng dụng từ nguồn không chính thống, nguy cơ bảo mật cao hơn.
IOS	<ul style="list-style-type: none">– Độc quyền: Chỉ có sẵn trên các thiết bị của Apple.	<ul style="list-style-type: none">– Tính ổn định và hiệu năng cao.	<ul style="list-style-type: none">– Độc quyền và giá cả cao: Giới hạn đối tượng người dùng.

	<ul style="list-style-type: none"> – AppStore: Nền tảng duy nhất để phân phối và tải xuống các ứng dụng cho iOS. – Giao diện người dùng: Được thiết kế thân thiện và dễ sử dụng, với các ứng dụng tích hợp chất lượng cao. – Bảo mật: Tính năng bảo mật mạnh mẽ, bao gồm mã hóa dữ liệu, bảo mật sinh trắc học (Face ID, Touch ID) 	<ul style="list-style-type: none"> – Bảo mật và bảo vệ quyền riêng tư mạnh mẽ. – Hệ sinh thái Apple đồng nhất và tương thích cao giữa các thiết bị. 	<ul style="list-style-type: none"> – Hạn chế tùy biến: Hạn chế trong việc tùy biến hệ điều hành và cài đặt ứng dụng từ bên ngoài AppStore.
Windows Phone	<ul style="list-style-type: none"> – Giao diện Metro: Giao diện người dùng với các ô vuông động (Live Tiles), cung cấp thông tin cập nhật trực tiếp trên màn hình chính. – Hệ sinh thái Microsoft: Tích hợp tốt với các dịch vụ của Microsoft như Office, OneDrive và Outlook. – Tính năng đồng bộ: Khả năng đồng bộ hóa mạnh mẽ với hệ điều hành 	<ul style="list-style-type: none"> – Giao diện người dùng trực quan và dễ sử dụng. – Tích hợp chặt chẽ với các dịch vụ của Microsoft. – Hiệu năng mượt mà trên các thiết bị cấu hình trung bình. 	<ul style="list-style-type: none"> – Số lượng ứng dụng hạn chế so với Android và iOS. – Thiếu sự hỗ trợ và cập nhật từ Microsoft sau khi ngừng phát triển. – Ít sự lựa chọn về thiết bị.

	Windows trên máy tính và các thiết bị khác của Microsoft.		
HarmonyOS	<ul style="list-style-type: none"> – Kiến trúc vi nhân: Sử dụng microkernel để tối ưu hiệu năng và bảo mật. – Super Device: Kết nối liền mạch giữa các thiết bị Huawei. – AppGallery: Kho ứng dụng thay thế cho Google Play Store. – Đa nền tảng: Thiết kế để chạy trên nhiều loại thiết bị khác nhau. 	<ul style="list-style-type: none"> – Hiệu năng tốt – Tích hợp mượt mà trong hệ sinh thái Huawei – Hỗ trợ đa nền tảng – Độc lập với Google Services 	<ul style="list-style-type: none"> – Thiếu các ứng dụng Google phổ biến – Giới hạn về địa lý và thị trường – Hệ sinh thái ứng dụng còn hạn chế – Phụ thuộc vào Huawei

Câu 2: Liệt kê các nền tảng phát triển ứng dụng di động phổ biến hiện nay và so sánh sự khác biệt chính giữa chúng.

Native Development (Phát triển nguyên bản)

- iOS (Swift/Objective-C)
 - Ưu điểm:
 - Hiệu năng cao nhất
 - Truy cập đầy đủ tính năng phần cứng
 - UX/UI mượt mà, đúng chuẩn Apple
 - Nhược điểm:
 - Chỉ chạy trên iOS
 - Chi phí phát triển cao
 - Cần học ngôn ngữ riêng
- Android (Kotlin/Java)

- Ưu điểm:
 - Hiệu năng tốt
 - Tích hợp sâu với hệ thống Android
 - Nhiều thư viện hỗ trợ
- Nhược điểm:
 - Chỉ chạy trên Android
 - Thời gian phát triển lâu
 - Phải bảo trì hai code base riêng biệt

Cross-platform Development (Phát triển đa nền tảng)

- React Native
 - Ưu điểm:
 - Một code base chạy được cả iOS/Android
 - Cộng đồng lớn, nhiều thư viện
 - Dễ học với dev web
 - Nhược điểm:
 - Hiệu năng thấp hơn native
 - Khó xử lý các tính năng phức tạp
 - Phụ thuộc bridge JavaScript
- Flutter
 - Ưu điểm:
 - Hiệu năng gần native
 - Hot reload giúp phát triển nhanh
 - Widget system linh hoạt
 - Nhược điểm:
 - Học curve khá cao với Dart
 - Kích thước ứng dụng lớn
 - Ecosystem còn non trẻ

Progressive Web Apps (PWA)

- Ưu điểm:

- Phát triển nhanh
- Dễ cập nhật
- Chi phí thấp
- Nhược điểm:
 - Hiệu năng hạn chế
 - Truy cập phần cứng bị giới hạn
 - Trải nghiệm không mượt mà bằng native

Hybrid Development (Phát triển lai)

- Ionic/Capacitor
 - Ưu điểm:
 - Sử dụng web technology quen thuộc
 - Triển khai nhanh
 - Chi phí thấp
 - Nhược điểm:
 - Hiệu năng kém
 - UX không tự nhiên
 - Khó tối ưu

So sánh tổng quan:

- Về hiệu năng: Native > Flutter > React Native > Hybrid > PWA
- Về chi phí và thời gian phát triển: PWA < Hybrid < React Native/Flutter < Native
- Về trải nghiệm người dùng: Native > Flutter > React Native > Hybrid > PWA
- Về khả năng bảo trì: Flutter/React Native > PWA > Hybrid > Native
- Về learning curve: PWA < React Native < Hybrid < Native < Flutter

Câu 3: Điều gì làm cho Flutter trở thành một lựa chọn phổ biến cho việc phát triển ứng dụng đa nền tảng? So sánh với các nền tảng khác như React Native và Xamarin.

Flutter phổ biến vì:

- Hot Reload giúp phát triển nhanh
- Widget system linh hoạt

- Hiệu năng cao nhờ Dart và Skia engine
- Cộng đồng lớn mạnh
- Được Google hỗ trợ

So sánh:

Tiêu chí	Flutter	React Native	Xamarin
Hiệu năng	<ul style="list-style-type: none"> – Hiệu năng gần native nhất – Skia engine render trực tiếp – Không cần bridge – Tối ưu memory management 	<ul style="list-style-type: none"> – Hiệu năng trung bình – JavaScript bridge overhead – Re-render không tối ưu – Memory leaks phổ biến 	<ul style="list-style-type: none"> – Hiệu năng khá tốt – Compile trực tiếp native – Quản lý bộ nhớ tốt – Startup time chậm
Phát triển	<ul style="list-style-type: none"> – Hot reload nhanh nhất – Developer tools tốt – Widget system trực quan – Testing framework đầy đủ 	<ul style="list-style-type: none"> – Hot reload tốt – Component reuse cao – NPM ecosystem phong phú – Debug tools đa dạng 	<ul style="list-style-type: none"> – Visual Studio IDE mạnh mẽ – .NET tools chuyên nghiệp – Testing framework tích hợp – Build time lâu
UI/UX	<ul style="list-style-type: none"> – UI nhất quán đa nền tảng – Custom widgets phong phú – Animation mượt mà 	<ul style="list-style-type: none"> – Native UI components – Platform-specific code – Custom components phức tạp 	<ul style="list-style-type: none"> – Native UI cho từng platform – XAML UI markup – Platform-specific features

	<ul style="list-style-type: none"> – Material & Cupertino sẵn có 	<ul style="list-style-type: none"> – Animation đôi khi lag 	<ul style="list-style-type: none"> – Custom controls hạn chế
Cộng đồng & Hỗ trợ	<ul style="list-style-type: none"> – Cộng đồng tăng trưởng nhanh – Google hỗ trợ mạnh – Documentation chi tiết – Package ecosystem đang phát triển 	<ul style="list-style-type: none"> – Cộng đồng lớn nhất – Nhiều thư viện third-party – Stack Overflow active – Facebook support 	<ul style="list-style-type: none"> – Cộng đồng nhỏ hơn – Microsoft enterprise support – Documentation chuyên nghiệp – Ít thư viện third-party
Chi phí & Thời gian	<ul style="list-style-type: none"> – Free và open source – Phát triển nhanh – Learning curve trung bình – CI/CD dễ setup 	<ul style="list-style-type: none"> – Free và open source – Developers JS dễ tìm – Phát triển khá nhanh – CI/CD nhiều tools 	<ul style="list-style-type: none"> – Chi phí license cao – Developers C# khó tìm – Phát triển chậm hơn – CI/CD tích hợp Visual Studio

Câu 4: Liệt kê các ngôn ngữ lập trình chính được sử dụng để phát triển ứng dụng trên Android và giải thích tại sao chúng lại được chọn.

Java

- Ngôn ngữ truyền thống cho Android
- Cộng đồng lớn, nhiều tài liệu
- OOP mạnh mẽ
- Được chọn vì độ ổn định và phổ biến

Kotlin

- Ngôn ngữ chính thức cho Android từ 2019
- Tương thích hoàn toàn với Java
- Cú pháp hiện đại, an toàn null
- Được chọn vì tính năng hiện đại và hiệu quả

Câu 5: Liệt kê các ngôn ngữ lập trình chính được sử dụng để phát triển ứng dụng trên iOS.

Swift

- Ngôn ngữ chính thức của Apple từ 2014
- Hiện đại, an toàn
- Hiệu năng cao
- Dễ học và sử dụng

Objective-C

- Ngôn ngữ truyền thống
- Vẫn được dùng trong các dự án legacy
- Tương thích tốt với C/C++
- Cú pháp phức tạp hơn Swift

Câu 6: Hãy thảo luận về những thách thức mà Windows Phone đã phải đối mặt và nguyên nhân dẫn đến sự sụt giảm thị phần của nó.

Thách thức chính:

- Thiếu ứng dụng phổ biến
- Thị phần nhỏ không hấp dẫn nhà phát triển
- Ra mắt muộn sau Android và iOS
- Chiến lược marketing không hiệu quả
- Hạn chế về phần cứng

Câu 7: Khám phá các ngôn ngữ và công cụ để phát triển ứng dụng web trên thiết bị di động.

Ngôn ngữ:

- HTML5
- CSS3 + Media Queries
- JavaScript/TypeScript
- PHP, Python, Ruby (Backend)

Công cụ:

- Framework Frontend:
 - React
 - Vue.js
 - Angular
- Framework Mobile-first:
 - Bootstrap
 - Foundation
 - Tailwind CSS
- Testing Tools:
 - Chrome DevTools
 - Browser Stack
 - Mobile-Friendly Test

Câu 8: Nghiên cứu về 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:

- Tăng trưởng mạnh cho mobile developer
- Lương cao hơn web developer
- Nhiều cơ hội việc làm quốc tế

Kỹ năng được yêu cầu:

1. Technical Skills:

- Native development (Android/iOS)
- Cross-platform frameworks
- Version control (Git)

- API integration
- Database management
- Testing & debugging

2. Soft Skills:

- Problem solving
- Team work
- Communication
- Time management
- Continuous learning

3. Business Skills:

- UI/UX understanding
- App monetization
- Analytics
- Security awareness
- Performance optimization