Ngô Văn Huy Hoàng

2180608513

Câu 1:

Nền tảng cho thiết bị di động thông minh hiện nay: IOS, Android, Window Phone.

- iOS: iOS
  - Đặc điểm: Là nền tảng độc quyền do Apple phát triển, chỉ hoạt động trên các thiết bị của Apple như iPhone và iPad.
  - Ưu điểm: Tính bảo mật cao, khả năng tối ưu phần mềm tốt, hiệu năng ổn định mà không cần đòi hỏi nhiều về cấu hình so với Android.
  - Khuyết điểm: Hệ điều hành chỉ độc quyền cho các dòng điện thoại của Apple và không thể sử dụng trên các điện thoại khác, kho ứng dụng ít hơn so với Android.
- Android: 
  - Đặc điểm: Là nền tảng mã nguồn mở do Google phát triển, được nhiều nhà sản xuất sử dụng như Samsung, Xiaomi,... Không chỉ sử dụng trên các thiết bị di động, Android còn được tùy biến cho TV, máy chơi game và các thiết bị điện tử khác.
  - Ưu điểm: Hệ điều hành mở, vì hầu hết cách thiết bị di động điều sử dụng nên Android sở hữu kho ứng dụng khổng lồ, khả năng tùy biến cao, dễ dàng đặt lại thiết bị nếu như quên mật khẩu.
  - Khuyết điểm: Hiện tại độ bảo mật của Android là khá cao nhưng sẽ không bằng nếu so sánh với iOS.
- Windows Phone (hiện đã ngừng phát triển): 
  - Đặc điểm: Được phát triển bởi Microsoft, tích hợp tốt với hệ sinh thái Windows.
  - Ưu điểm: Tích hợp liền mạch với máy tính Windows, phù hợp với các giải pháp doanh nghiệp.
  - Khuyết điểm: Thị phần hạn chế, thiếu hỗ trợ ứng dụng, và đã bị ngừng hỗ trợ từ Microsoft.

Câu 2:

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

1. **Flutter:**

   - Ngôn ngữ sử dụng: Dart
   - Đặc điểm nổi bật: Flutter cho phép lập trình viên xây dựng giao diện người dùng mượt mà với các hiệu ứng chuyển động phong phú. Do sử dụng cùng một mã nguồn cho cả Android và iOS, Flutter giúp giảm thời gian và chi phí phát triển.
   - Ưu điểm: Hiệu năng tốt gần như ứng dụng gốc, giao diện đẹp, hỗ trợ mạnh mẽ từ Google.
   - Nhược điểm: Cộng đồng còn mới và chưa phát triển rộng rãi bằng các nền tảng khác, hỗ trợ thư viện bên thứ ba còn hạn chế.

2. **React Native:**

   - Ngôn ngữ sử dụng: JavaScript
   - Đặc điểm nổi bật: Là nền tảng mã nguồn mở do Facebook phát triển, React Native cho phép sử dụng một mã nguồn để phát triển ứng dụng cho cả Android và iOS, tiết kiệm chi phí và thời gian.

- Ưu điểm: Cộng đồng lớn, nhiều thư viện hỗ trợ, dễ dàng tích hợp với các framework JavaScript khác, có thể tái sử dụng một số thành phần UI.
- Nhược điểm: Hiệu năng không cao bằng ứng dụng gốc, cần mã gốc (native code) cho các tính năng phức tạp.

3. **Xamarin:**

- Ngôn ngữ sử dụng: C#
- Đặc điểm nổi bật: Xamarin được phát triển bởi Microsoft, cho phép viết một mã nguồn và chạy trên nhiều nền tảng, đặc biệt phù hợp cho các ứng dụng doanh nghiệp.
- Ưu điểm: Tích hợp tốt với Visual Studio, khả năng truy cập các API gốc của Android và iOS, phù hợp với các dự án cần sự bảo mật và hiệu suất cao.
- Nhược điểm: Kích thước ứng dụng lớn, cấu hình ban đầu phức tạp, ít được sử dụng cho các dự án nhỏ hoặc ứng dụng cá nhân.

4. **Native Android và iOS:**

- Ngôn ngữ sử dụng: Java/Kotlin cho Android và Swift/Objective-C cho iOS.
- Đặc điểm nổi bật: Phát triển trực tiếp trên nền tảng Android hoặc iOS, tận dụng tối đa khả năng của thiết bị.
- Ưu điểm: Hiệu suất tối ưu, truy cập đầy đủ các tính năng phần cứng của thiết bị, tương thích cao.
- Nhược điểm: Chi phí và thời gian phát triển cao do phải viết mã riêng cho từng nền tảng, khó bảo trì khi cần phát hành ứng dụng cho cả hai nền tảng.

So sánh sự khác biệt chính:

- **Hiệu suất:** Native (Android, iOS) có hiệu suất tốt nhất, sau đó là Flutter, React Native, và cuối cùng là Xamarin.
- **Thời gian phát triển:** Flutter, React Native và Xamarin tiết kiệm thời gian hơn so với Native vì sử dụng chung mã nguồn.
- **Tính năng:** Native có khả năng truy cập tất cả các tính năng của hệ điều hành. Các nền tảng khác như Flutter và React Native có thể truy cập hầu hết các tính năng nhưng đôi khi cần viết mã gốc cho những tính năng đặc biệt.
- **Cộng đồng hỗ trợ:** React Native có cộng đồng lớn nhất, tiếp theo là Flutter, Xamarin, và cuối cùng là Native khi phải phụ thuộc vào từng nền tảng cụ thể (Android hoặc iOS).

Câu 3:

Flutter trở thành một lựa chọn phổ biến cho việc phát triển ứng dụng đa nền tảng vì:

- **Giao diện đẹp và nhất quán trên cả Android và iOS:** Flutter sử dụng công cụ đồ họa riêng, không phụ thuộc vào các thành phần giao diện gốc của hệ điều hành, giúp ứng dụng trông giống nhau trên mọi thiết bị.
- **Hiệu suất cao:** Flutter được xây dựng bằng ngôn ngữ Dart và biên dịch trực tiếp thành mã gốc (native code), giúp tăng tốc độ và giảm độ trễ khi sử dụng.
- **Hot Reload:** Cho phép lập trình viên xem kết quả ngay sau khi thay đổi mã, giúp đẩy
- **Hỗ trợ widget phong phú:** Flutter cung cấp một loạt các widget tùy chỉnh, cho phép
- **Hỗ trợ mạnh từ Google:** Là sản phẩm của Google, Flutter được hỗ trợ phát triển mạnh mẽ và liên tục cải tiến.

Bảng so sánh Flutter với React Native và Xamarin:

Tiêu chí	**Flutter**	**React Native**	**Xamarin**
Ngôn ngữ	Dart	JavaScript	C#
Hiệu suất	Tốt nhờ biên dịch trực tiếp thành mã gốc	Tương đối tốt, nhưng gặp hạn chế khi dùng cầu nối (bridge)	Tốt, nhưng kích thước ứng dụng lớn hơn
Tính năng Hot Reload	Có, giúp tăng tốc phát triển	Có, nhưng không mạnh bằng Flutter	Có, nhưng không hiệu quả như Flutter hoặc React Native
Giao diện (UI)	Sử dụng widget riêng, đảm bảo nhất quán trên các nền tảng	Sử dụng các thành phần UI gốc của hệ điều hành	Phụ thuộc vào UI gốc của hệ điều hành
Cộng đồng	Đang phát triển nhanh, được hỗ trợ mạnh từ Google	Cộng đồng lớn, do ra đời sớm và dựa trên JavaScript	Cộng đồng nhỏ hơn, chủ yếu là lập trình viên C# và .NET
Tài nguyên thư viện	Nhiều thư viện, nhưng chưa phong phú như React Native	Phong phú, nhiều thư viện từ cộng đồng	Ít thư viện hơn, phụ thuộc vào hệ sinh thái của Microsoft
Khả năng tích hợp	Hỗ trợ tốt các API hệ thống và thiết bị phần cứng	Cần viết mã gốc cho các tính năng phức tạp	Tích hợp tốt với hệ sinh thái của Microsoft, phù hợp cho doanh nghiệp

Mức độ phổ biến	Đang tăng nhanh, đặc biệt trong các ứng dụng yêu cầu UI đẹp	Phổ biến nhất trong các nền tảng đa nền tảng	Ít phổ biến hơn, chủ yếu dùng trong các dự án doanh nghiệp
Đối tượng phù hợp	Phát triển ứng dụng có giao diện đẹp và mượt mà	Phát triển nhanh, tiết kiệm chi phí	Các ứng dụng doanh nghiệp, cần bảo mật và ổn định cao

Câu 4:

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

1. **Java:**
   - Lý do sử dụng: Java là ngôn ngữ chính thức và lâu đời nhất cho phát triển Android. Google đã phát triển Android SDK dựa trên Java, do đó có rất nhiều thư viện và tài liệu hỗ trợ.
   - Ưu điểm: Java có cộng đồng lớn, khả năng bảo mật tốt và nhiều thư viện có sẵn để hỗ trợ phát triển ứng dụng.
   - Nhược điểm: Cú pháp Java khá dài dòng và không linh hoạt bằng các ngôn ngữ mới hơn, dễ dẫn đến lỗi nếu không cẩn thận.
2. **Kotlin:**
   - Lý do sử dụng: Kotlin là ngôn ngữ lập trình hiện đại do JetBrains phát triển và được Google chính thức hỗ trợ cho Android vào năm 2017. Kotlin được thiết kế để khắc phục các hạn chế của Java và mang lại trải nghiệm lập trình tốt hơn.
   - Ưu điểm: Cú pháp ngắn gọn, dễ đọc, giảm thiểu lỗi Null Pointer Exception, tương thích tốt với Java, và có hiệu năng cao.
   - Nhược điểm: Vì mới được phát triển nên cộng đồng và tài nguyên hỗ trợ không phong phú bằng Java, nhưng đang phát triển nhanh chóng.

3. **C++:**
    - Lý do sử dụng: C++ được sử dụng chủ yếu để phát triển các thành phần yêu cầu hiệu suất cao, như phần mềm đồ họa, xử lý dữ liệu hoặc các trò chơi phức tạp.
    - Ưu điểm: Hiệu suất cao, giúp xử lý tốt các tác vụ phức tạp và nặng về CPU.
    - Nhược điểm: C++ khó học hơn, dễ gặp lỗi nếu không quản lý tốt bộ nhớ, và không thân thiện với lập trình viên mới bắt đầu.
4. **JavaScript (với React Native):**
    - Lý do sử dụng: JavaScript là ngôn ngữ phổ biến trong phát triển ứng dụng đa nền tảng, nhờ các framework như React Native cho phép phát triển ứng dụng Android và iOS chỉ với một mã nguồn duy nhất.
    - Ưu điểm: Phát triển đa nền tảng, tiết kiệm chi phí và thời gian, cộng đồng lớn và nhiều thư viện hỗ trợ.
    - Nhược điểm: Hiệu suất không cao bằng các ngôn ngữ gốc, gặp giới hạn khi cần tích hợp các tính năng phức tạp.

<u>Câu 5:</u>

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

1. **Objective-C:**
    - Lý do sử dụng: Objective-C là ngôn ngữ lập trình chính thức đầu tiên được Apple sử dụng để phát triển ứng dụng iOS. Nó có nguồn gốc từ ngôn ngữ C và được bổ sung thêm các tính năng lập trình hướng đối tượng.
    - Ưu điểm: Tích hợp tốt với các API của Apple, đã được thử nghiệm và sử dụng lâu đời, cộng đồng lớn và có nhiều tài liệu hỗ trợ.
    - Nhược điểm: Cú pháp phức tạp, khó đọc hơn so với các ngôn ngữ hiện đại, dễ gặp lỗi nếu không cẩn thận, và ngày càng ít được sử dụng vì đã có ngôn ngữ thay thế là Swift.

2. **Swift:**

- Lý do sử dụng: Swift được Apple phát triển và giới thiệu vào năm 2014 như một ngôn ngữ thay thế cho Objective-C, với mục tiêu giúp lập trình viên dễ dàng tiếp cận và viết mã ngắn gọn, an toàn hơn.
- Ưu điểm: Cú pháp đơn giản, dễ đọc và dễ bảo trì; hỗ trợ mạnh mẽ các tính năng an toàn như ngăn ngừa lỗi Null Pointer; hiệu năng cao và tối ưu cho iOS; tích hợp tốt với Objective-C.
- Nhược điểm: Mặc dù đã phát triển nhanh chóng, Swift vẫn là ngôn ngữ mới hơn nên có ít tài liệu hơn so với Objective-C và có thể không tương thích hoàn toàn với các dự án cũ.

3. **JavaScript (với React Native):**

- Lý do sử dụng: JavaScript qua các framework như React Native cho phép phát triển ứng dụng iOS đa nền tảng, tiết kiệm thời gian và chi phí bằng cách sử dụng một mã nguồn duy nhất cho cả Android và iOS.
- Ưu điểm: Phát triển đa nền tảng, cộng đồng lớn, nhiều thư viện hỗ trợ, và thân thiện với lập trình viên JavaScript.
- Nhược điểm: Hiệu suất không tốt bằng ngôn ngữ gốc như Swift và Objective-C, khó tích hợp các tính năng phức tạp của hệ điều hành.

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 ứng dụng và hệ sinh thái yếu:**

- Một trong những thách thức lớn nhất của Windows Phone là thiếu hụt ứng dụng so với Android và iOS. Các nhà phát triển ứng dụng thường ưu tiên phát triển cho hai nền tảng có thị phần lớn này, dẫn đến việc các ứng dụng phổ biến không có trên Windows Phone hoặc ra mắt muộn hơn nhiều.
- Hệ sinh thái Windows Phone kém phong phú so với đối thủ, gây khó khăn cho người dùng khi không thể tiếp cận với các ứng dụng mà họ cần.

2. Thiếu sự quan tâm từ các nhà phát triển:

- Các nhà phát triển không mặn mà với việc phát triển ứng dụng trên Windows Phone vì thị phần thấp và lợi nhuận không cao. Việc này tạo ra một vòng luẩn quẩn: ít ứng dụng dẫn đến ít người dùng, và ít người dùng lại càng làm cho các nhà phát triển không muốn đầu tư vào nền tảng này.

3. Giao diện và tính năng không đột phá:

- Mặc dù giao diện Metro UI của Windows Phone khá độc đáo với các ô Live Tiles, nhưng nó không thực sự hấp dẫn đối với phần lớn người dùng. Giao diện này không đủ để thuyết phục người dùng chuyển từ Android hoặc iOS.
- Các tính năng của Windows Phone cũng không có gì nổi bật hay đột phá, khiến người dùng cảm thấy không có lý do để lựa chọn nền tảng này.

4. Vấn đề trong chiến lược marketing và đối tác phần cứng:

- Microsoft không đầu tư đủ vào chiến lược marketing để cạnh tranh với Apple và Google. Điều này làm cho người dùng ít biết đến hoặc không có động lực để chuyển sang Windows Phone.
- Windows Phone chủ yếu được hỗ trợ bởi Nokia và một vài nhà sản xuất nhỏ. Điều này làm cho người dùng có ít lựa chọn thiết bị hơn so với Android, vốn có rất nhiều mẫu mã đa dạng từ các hãng khác nhau.

5. Phát triển chậm và chiến lược không nhất quán:

- Microsoft không kịp thời cải thiện và phát triển hệ điều hành Windows Phone, khiến nó không theo kịp tốc độ đổi mới của Android và iOS. Bên cạnh đó, việc thay đổi chiến lược phát triển nhiều lần khiến cho cả người dùng và các nhà phát triển cảm thấy thiếu tin tưởng vào sự ổn định của nền tảng.
- Sự chuyển đổi từ Windows Phone sang Windows 10 Mobile và sau đó là từ bỏ hoàn toàn Windows Phone đã làm giảm niềm tin của người dùng và các đối tác.

6. Cạnh tranh gay gắt từ Android và iOS:

- Android và iOS đã chiếm lĩnh thị trường di động từ sớm và có sự cải tiến không ngừng, khiến Windows Phone không đủ sức để cạnh tranh. Android chiếm ưu thế với sự đa dạng và giá cả hợp lý, trong khi iOS có hệ sinh thái mạnh và người dùng trung thành.

Câu 7:

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

-Ngôn ngữ lập trình phổ biến để phát triển ứng dụng web trên di động

1. **HTML, CSS và JavaScript**
    - **HTML (HyperText Markup Language):** Dùng để tạo cấu trúc cơ bản cho trang web, như tiêu đề, đoạn văn, hình ảnh và liên kết.
    - **CSS (Cascading Style Sheets):** Được sử dụng để thiết kế giao diện và bố cục, tạo nên màu sắc, kiểu chữ, và các yếu tố đồ họa cho trang web.
    - **JavaScript:** Ngôn ngữ lập trình chính cho các trang web động, cho phép xử lý sự kiện và tương tác người dùng trên trang web.
    - **Ưu điểm:** Ba ngôn ngữ này là nền tảng cốt lõi của mọi trang web và được hỗ trợ bởi tất cả các trình duyệt di động. Kết hợp HTML, CSS và JavaScript giúp tạo ra các trang web có giao diện thân thiện và chức năng phong phú.
2. **TypeScript**
    - **Mô tả:** TypeScript là một phiên bản nâng cao của JavaScript, phát triển bởi Microsoft, với tính năng hỗ trợ kiểu dữ liệu tĩnh (static typing).
    - **Ưu điểm:** Giúp phát hiện lỗi sớm, dễ bảo trì và phát triển ứng dụng lớn, đặc biệt là khi làm việc với các framework lớn như Angular.
    - **Nhược điểm:** Yêu cầu kiến thức về JavaScript và có thể tăng độ phức tạp cho dự án.

-Công cụ và framework phổ biến

1. **React Native**
    - **Mô tả:** Được phát triển bởi Facebook, React Native cho phép xây dựng ứng dụng di động bằng JavaScript và React, có thể chạy trên cả Android và iOS.
    - **Ưu điểm:** Sử dụng lại nhiều mã nguồn từ ứng dụng web React, giảm chi phí và thời gian phát triển cho đa nền tảng.
    - **Nhược điểm:** Hiệu suất không tốt bằng ứng dụng gốc (native), cần kiến thức về JavaScript và React.

2. **Flutter**
    - **Mô tả:** Là framework do Google phát triển, sử dụng ngôn ngữ Dart để tạo ứng dụng di động. Flutter cho phép tạo ứng dụng có giao diện đẹp và mượt mà.
    - **Ưu điểm:** Hiệu suất tốt gần như ứng dụng gốc, khả năng tùy chỉnh giao diện cao.
    - **Nhược điểm:** Dart là ngôn ngữ ít phổ biến, cộng đồng còn hạn chế so với JavaScript.
3. **Angular và Ionic**
    - **Mô tả:** Angular là framework JavaScript mạnh mẽ do Google phát triển, hỗ trợ tốt cho việc phát triển ứng dụng web trên di động. Ionic là framework kết hợp với Angular để tạo ứng dụng di động đa nền tảng.
    - **Ưu điểm:** Angular có cấu trúc mạnh mẽ, phù hợp cho các ứng dụng lớn. Ionic cung cấp các thành phần UI phong phú, giúp tạo ứng dụng có giao diện như ứng dụng gốc.
    - **Nhược điểm:** Hiệu suất không tốt bằng ứng dụng gốc, cần kiến thức sâu về Angular.
4. **Vue.js và Quasar Framework**
    - **Mô tả:** Vue.js là một framework JavaScript linh hoạt và dễ học, phù hợp với các ứng dụng đơn giản đến phức tạp. Quasar là framework tích hợp với Vue để tạo ứng dụng di động và web đa nền tảng.
    - **Ưu điểm:** Dễ học và sử dụng, giúp phát triển nhanh chóng ứng dụng di động và web.
    - **Nhược điểm:** Không mạnh mẽ bằng Angular trong các dự án lớn, yêu cầu kiến thức về Vue.js.
5. **Progressive Web Apps (PWA)**
    - **Mô tả:** PWA là một loại ứng dụng web đặc biệt có thể cài đặt và hoạt động như một ứng dụng di động mà không cần qua kho ứng dụng (App Store hoặc Google Play).
    - **Ưu điểm:** Dễ phát triển và triển khai, có thể truy cập từ trình duyệt và không cần tải về từ App Store, có khả năng hoạt động offline.

- **Nhược điểm:** Không có quyền truy cập đầy đủ vào các tính năng phần cứng của thiết bị, không phù hợp với các ứng dụng đòi hỏi hiệu suất cao.

-So sánh các công cụ phát triển ứng dụng web di động:

Công cụ / Ngôn ngữ	Ưu điểm	Nhược điểm
React Native	Phát triển đa nền tảng, tái sử dụng mã nguồn	Hiệu suất kém hơn ứng dụng gốc
Flutter	Hiệu suất cao, giao diện mượt mà	Ngôn ngữ Dart chưa phổ biến
Angular & Ionic	Cấu trúc mạnh mẽ, nhiều thành phần UI	Hiệu suất thấp, yêu cầu kiến thức Angular
Vue.js & Quasar	Dễ học, phát triển nhanh	Không mạnh cho dự án lớn

Câu 8:

Nhu cầu nhân lực lập trình viên di động hiện nay:

- Đa dạng hóa công nghệ

- Yêu cầu tích hợp và bảo mật nâng cao

- Cập nhật kỹ năng liên tục

- Kiểm thử đa nền tảng

- Có khả năng phát triển ứng dụng cho cả smartphone và tablet

Hiện nay, những kỹ năng được yêu cầu nhiều nhất cho lập trình viên trên thiết bị di động bao gồm:

- Ngôn ngữ lập trình: Kiến thức vững về Java và Kotlin cho phát triển ứng dụng Android, cũng như Swift cho iOS.

- Thiết kế UI/UX: Khả năng thiết kế giao diện người dùng và trải nghiệm người dùng hấp dẫn và dễ sử dụng.

- Kiểm thử tự động: Kỹ năng viết test tự động để đảm bảo chất lượng ứng dụng.

- Kỹ năng mềm: Kỹ năng giao tiếp, xử lý tình huống và khả năng làm việc nhóm cũng rất quan trọng trong môi trường phát triển phần mềm.

- Machine Learning và AI trên di động , Internet of Things (IoT),…