**Diagram**

**1. Tại sao lại có bảng Guest trong conceptual diagram? Bảng Guest không bao giờ có ở đây?**

**2. Tại sao lại bảng Guest lại có trong conceptual diagram mà không có trong entity diagram?**

🡪 **Conceptual diagram để xây dựng các đối tượng xuất hiện trong user requirement.** Do vậy trong requirement của nhóm trước đó có đối tượng Guest là người sử dụng ở mobile client nên nhóm sẽ xây dựng đối tượng Guest trong conceptual diagram.

**3. Tại sao conceptual diagram ở trang 73 không có thực thể Inode nhưng trong bản description lại có?**

**\*Note: theo quan điểm thầy Tài, trong ERD, khi A kế thừa X, B kế thừa X, thì A và B không thể có quan hệ một nhiều.**

🡪 Ban đầu nhóm thiết kế để segment là is-a với INode và Path cũng là is-a với INode, và do vậy trong mô hình conceptual diagram, segment không thể một nhiều với Path, do vậy sẽ bỏ đi thực thể INode như trang 73, tuy nhiên do sai sót trong quá trình biên tập tài liệu nên thực thể Inode vẫn còn xuất hiện trong description. Nhóm chúng em xin lỗi vì sai sót này.

**Tại sao segment và path kế thừa từ thực thể INode lại sai? (cho câu trả lời trên)**

**Tại sao ở conceptual diagram segment và path không có kế thừa mà ở class diagram có kế thừa ?**

🡪 Ở bảng conceptual diagram, segment có quan hệ is a với Inode và Path cũng có quan hệ is a với Inode, do vậy segment và path không thể có quan hệ 1 nhiều với nhau cho nên nhóm chúng em quyết định xóa đi thực thể Inode trong conceptual.

(đối với Class diagram để xây dựng các class trong quá trình hiện thực thuật toán và business, cho nên nhóm chúng em thấy cách xây dựng 1 interface chung Inode là thuận tiện)

**5. Tại sao những thực thể như Path, Segment có ở conceptual diagram nhưng không có ở entity diagram hoặc physical database design.**

🡪 **Conceptual diagram để xây dựng các đối tượng xuất hiện trong user requirement** ở đây các đối tượng như Path, Segment, Result, Journey, Leg, Step dung để hiển thị dữ liệu cho mobile. Cho nên nhóm quyết định đưa vào conceptual diagram. Entity diagram, physical database được thiết kế dung để lưu trữ dữ liệu , tuy nhiên các thông tin trên hiện tại không được lưu trữ cho nên không đưa vào entity, physical database.

**6. Ở conceptual (web, mobile) và entity diagram, route có 2 quan hệ với station, tại sao bên physical database design route không còn quan hệ với station nữa?**

**(ở điểm này nhóm em thay đổi theo thầy Tài)**

🡪 Khi xây dựng conceptual diagram nhóm chúng em xây dựng dựa trên thực tế đó là ở một chuyến xe bus chúng ta cần biết trạm đầu là gì và trạm kết thúc là gì do vậy ở conceptual và entity diagram có 2 quan hệ là trạm đầu và trạm kết thúc. Tuy nhiên khi thiết kế physical database design nhóm thấy rằng quan hệ đó không giúp ích cho việc truy vấn và xử lý thuật toán. Có thể biết được trạm đầu và trạm kết thúc qua pathinfo cho nên nhóm quyết định xóa đi 2 quạn hệ đó.

**7. Ở conceptual (web, mobile) và entity diagram, route có 2 quan hệ với station, bên class diagram route không còn quan hệ với station nữa ?**

**(cách trả lời tương tự như trên, dựa vào thay đổi từ thầy Tài)**

🡪 Khi xây dựng conceptual diagram nhóm chúng em xây dựng dựa trên thực tế đó là ở một chuyến xe bus trạm đầu là gì và trạm kết thúc là gì do vậy ở conceptual và entity diagram có 2 quan hệ là trạm đầu và trạm kết thúc. Tuy nhiên khi thiết kế class diagram nhóm thấy rằng quan hệ đó không giúp ích cho việc truy vấn và xử lý thuật toán. Có thể biết được trạm đầu và trạm kết thúc qua pathinfo cho nên nhóm quyết định xóa đi 2 quạn hệ đó.

**8. Tại sao nhóm không có installation guide trang 172.**

🡪[Mở tài liệu đã chỉnh sửa] nhóm chúng em xin xin lỗi vì thiếu sót này, nhóm chúng em chưa kịp đưa phần này vào tài liệu gửi hội đồng. tuy nhiên nhóm đã bỏ xung thiếu sót này vào tài liệu

9**. Trong conceptual và entity diagram, giữa route và pathinfo có quan hệ nhiều nhiều, đồng thời lại có thực thể trip ở giữa. ?**

**(thêm vào quan hệ nhiều nhiều dựa trên thay đổi thầy Tài)**

Quan hệ giữa Route và PathInfo là quan hệ nhiều-nhiều. Tuy nhiên, do giữa mỗi đoạn, có nhiều chuyến đi trong ngày đi ngang qua, nên chúng em them thực thể Trip**.**

**Các câu hỏi khác.**

**1. Tại sao lại không sử dụng google maps API để tìm kiếm đường đi bằng xe bus mà phải sử dụng thuật toán riếng với server riêng?**

🡪 Lý do 1: với cách xây dựng server riêng, tụi em có thể tùy chỉnh thuật toán dễ dàng hơn ví dụ đưa ra hệ thống thông báo kẹt xe, hệ thống lưu ý thông tin đường cao tốc đền đỏ…vv

Lý do 2: google maps api thiếu một số các chức năng ví dụ thời gian di chuyển, số lần chuyển tuyến tối đa và tìm kiếm nhiều hơn điểm, cụ thể từ 2 – 4 điểm.

**2. Nhóm chúng em sử dụng google map api cho motor có khác gì so với sử dụng google map.**

🡪 Hệ thống chúng em hơn google map là tìm kiếm nhiều hơn 2 điểm, hổ trợ map offline, hổ trợ thống báo khi tham gia giao thông, hổ trợ thông báo khi sai đường và tìm lại đường đi đúng, hổ trợ trên các thiết bị đồng hồ thông minh. Google map không có những chức năng này.

**3. Tại sao bản đồ trên wear lại sử dụng google map mà trên mobile lại dung nutieq.**

🡪Đây là nhược điểm trong hệ thống của nhóm, hệ thống của nhóm sử dụng nutti map offline cho mobile tuy nhiên, map này vẫn chưa hổ trơ trên đồng hồ thông minh nên nhóm phải sử dụng google map. Do vậy khi tham gia giao thông sử dụng đồng hồ, đồng hồ hoặc điện thoại phải có 3g.

**Câu hỏi thầy khánh**

**1) Khi giao thong trên đường đang có 3G, hệ thong đang sử dung và người dung xem bản đồ, rồi 3G mất, hệ thong có hỗ trợ cơ chế tự chuyển đổi sang bản đồ offline và hỗ trợ người dung tiếp hay ko?**

- Khi bản đồ chưa được tải, hệ thống sẽ dùng bản đồ online (nếu có 3G), nếu đã tải bản đồ rồi thì hệ thống luôn dùng bản đồ offline (dù 3G có đang được bật hay không).

**2) Khi demo di chuyển simulator xe máy, đồng hồ chạy theo, điện thoại có hiển thi đi thằng hay rẽ trái phải nhưng đồng hồ chỉ có bản đồ di chuyển không có thong báo gì cả???**

- Đây là do cơ chế truyền dữ liệu của wear và mobile. Việc thông báo là dùng chức năng notification của hệ thống. API sẽ tự handle notification để gửi qua wear. Delay notification luôn cao hơn dữ liệu gửi trực tiếp từ app mobile sang app wear. (nên người dùng sẽ luôn marker di chuyển trước khi thấy thông báo).

**3) Khi download âm thanh với cơ chế cache, luôn ghi đè hay ghi tiếp đến hết 100 MB thì ghi đè? Có hỏi người dung có muốn ghi đè hay xóa gì ko hay tự tiện xóa vì đôi lúc họ có những còn đường favourite?**

- File download về chỉ là những thông báo dẫn đường. Thuật toán sẽ tự động ghi đè khi ở đoạn cuối 100MB để đủ ghi file. Thường một tuyến đường đi qua 4 điểm mất tối đa khoảng 5MB, nên dung lượng 100MB luôn đủ cho một hành trình di chuyển.

4) **Hệ thong có cho phép người dung lưu favourite cho user khoảng 2 đến 3 lộ trình tối đa ko? Nếu hệ thong chưa có thì hệ thong có thể làm điều đó được hay không? Bằng cách nào, bổ sung thực thể hay class nào ...?**

🡪 hiện tại hệ thống chỉ lưu các địa chỉ tìm kiếm gần đây chưa lưu favorite cho các lộ trình. Tuy nhiên nhóm đã nghĩ tới nhược điểm này và đã xây dựng 1 bảng các thực thể để làm được điều này. Nhóm xin mở slide để rõ hơn

[bật slide khác].

5) Khoảng thời gian delay giữa điện thoại và wear là bao nhiêu?

- trung bình 2-3s

**Component**

**1. Tại sao lại tạo ra một lớp application context?**

**2. Tại sao application context lại sử dung facede pattern đóng gói request và response và đưa ra các hàm cần thiết.**

🡪 Kiến trúc sử dụng facede or application context đóng gói lại request và response đưa ra các hàm cần thiết để đưa ra các hàm đơn giản cho người sử dung (lập trình viên). Lập trình viên không cần biết rõ về request or response, chỉ cần biết thông tìn các hàm làm gì.

**3. Tại sao ActitionFactory lại kết thừa từ IActionFactory mà không sử dụng trực tiếp?**

Để giảm đi sự phụ thuộc giữa dispatcherServlet và ActionFactory, trong tương lai nếu hệ thống sử dụng các url mới thì chỉ cần tạo một lớp actionFactory khác kế thừa từ IActionFatory code sẽ k thây đổi DispatcherServlet

**4. Dispatcher là của hệ thống hay của ứng dung**

🡪 do mình code.

**5. Tại sao chỉ có 1 dispatcherServlet, không tạo ra nhiều dispatcherServlet để đón nhận request?**

🡪 Việc đưa ra một dispatcher để dễ dàng quản lý hết tất các request gửi lên, vì khi đó xử lý các request sẽ do actionFactory đảm nhận. Khi hệ thống càng lớn, số lượng URL ngày càng nhiều, việc sử dụng nhiều servelet sẽ khó quản lý hơn.

**6. ActionFactory để làm gì ?, method factory design để làm gì?**

🡪 ActionFactory (method Factory ) là nơi tập trung để tạo ra các action. Việc tập trung một nơi để khởi tạo đối tượng giúp việc dễ dàng quản lí việc khởi tạo và quản lí khi lập trình.

**7. Tại sao các action lại hiện thực từ IAction**

🡪 Việc tạo ra 1 interface chung giúp cho dispatcherservlet có thể thực thi các action mà không cần biết các action đó cụ thể là gì.

**Wear**

1. Giao tiếp giữa mobile và wear sử dụng Bluetooth hay Bluetooth BLE?

🡪Giao tiếp với wear hệ thống sử dụng Bluetooth or Bluetooth BLE đều được.

**2. Wear không thể chuyển dữ liệu qua lại mobile? System architecture chỉ thấy mũi tên một chiều từ mobile sang wear.**

🡪 Kết nối giữa mobile và wear là 2 chiều, wear có thể chuyển dữ lieu qua lại mobile tuy nhiên hệ thống nhóm em chỉ đưa dữ liệu 1 chiều từ mobile qua wear.

**Web server**

**1. Khác biệt giữa business object và entity là gì?**

Entity là để mapping giữa từng đồi tượng và record của database, 1 attribute tương ứng với 1 cột trong db.

Business object là các đối tượng trên bộ nhớ để sử lý business của hệ thống.

Business component là nơi để tập trung sử lý business của hệ thống.