**Tên nhóm: Nhóm 3**

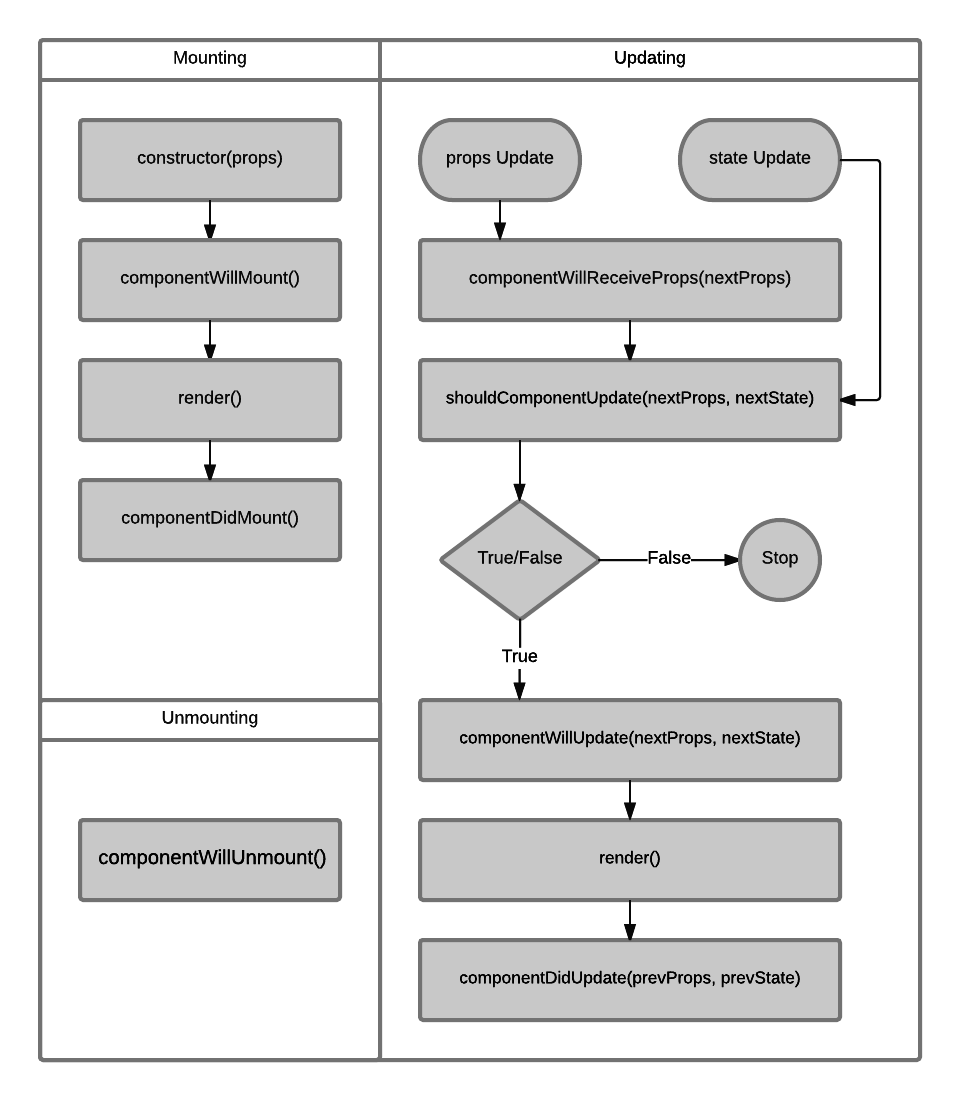
Thành viên

|  |  |  |  |
| --- | --- | --- | --- |
| **STT** | **Họ và tên** | **Email** | **SDT** |
| 1 | Trần Thị Xuân Hoàng | xuanhoang2937@gmail.com | 0978532509 |
| 2 | Ngô Xuân Chiến | xuanchienhut@gmail.com | 0355062707 |

Yêu cầu tuần 2.2

1. Vòng đời của Component

Một component khi bắt đầu được khởi tạo cho đến lúc bị xoá khỏi DOM sẽ trải qua 3 giai đoạn, đó là mounting (gắn), updating (cập nhật) và unmounting (gỡ bỏ). Mounting diễn tả quá trình một component được khởi tạo và vẽ lên trong DOM. Unmounting là khi một component bị xoá khỏi DOM. Giữa mount và unmount thì thường chúng ta sẽ cần cập nhật dữ liệu mới, đó là giai đoạn updating. React cung cấp cho chúng ta một danh sách các hàm tương ứng với từng giai đoạn



1. Props và State

* Props

props là viết tắt của Properties.

Các component nhận props từ component cha. Bạn không được thay đổi giá trị của props trong các component này mà chỉ được phép đọc giá trị ra thôi. Trong React thì dữ liệu sẽ đi theo một chiều, có nghĩa là từ component cha => các component con.

* State

state thì hoạt động khác với props. state là dữ liệu nội bộ của một Component, trong khi propslà dữ liệu được truyền cho Component. Chính vì vậy chúng ta hoàn toàn có thể thay đổi state

dùng hàm setState để cập nhật giá trị.

1. Style trong React Native

* Style: giống CSS
* Import:

**import** {  
 **StyleSheet**  
} **from 'react-native'**;

* Khai báo biến và định nghĩa giá trị

**const** styles = **StyleSheet**.create({  
 **container**: {  
 **flexDirection**: **'row'**,  
 **height**: 130,  
 **padding**: 10,  
 **marginBottom**: 10,  
 **backgroundColor**: **'lightblue'**,  
 },  
});

* Sử dụng

<**View style=**{styles.**container**}></**View**>

1. Width, Height và Flex

Là các thành phần trong style

* Width, height xác định chiều rộng, chiều cao của các khối trong giao diện (Kích thước cố định)

style={{width: 50, height: 50}}

* Flex: xác định kích thước dựa theo khối chứa nó,

<View style={{flex: 1}}> Khối cha chia làm 6 phần

<View style={{flex: 1, backgroundColor: 'powderblue'}} /> Khối con chứa 1 phần của khối cha

<View style={{flex: 2, backgroundColor: 'skyblue'}} /> Khối con chứa 2 phần của khối cha

<View style={{flex: 3, backgroundColor: 'steelblue'}} /> Khối con chứa 3 phần của khối cha

</View>

1. FlexBox

Là một thuộc tính giúp căn chỉnh vị trí tương đối của các thành phần con trong thành phần cha.

* flexDirection

Sắp xếp các đối tượng con theo column (chiều dọc) hoặc row (chiều ngang). 2 giá trị của nó là : rowhoặc column. Mặc định giá trị là column

* justifyContent

Căn vị trí tương đối của đối tượng con theo trục chính của flex. Gồm các giá trị: flex-start, center, flex-end, space-around, space-between

* alignItems

Căn vị trí tương đối của các đối tượng con theo trục còn lại. Gồm các giá trị: flex-start, flex-end , center, strecht.

* Flex-wrap : wrap | no-wrap

Thuộc tính này cho phép container bọc các items con mặc dù kích thước của chúng vượt qua kích thước của container. Măc định là no-wrap.

* Oder

Sắp xếp vị trí các items theo giá trị nhập

* Flex-grow

Phân định kích thước của các đối tượng con tương đối bằng tỉ lện giữa các giá trị flex-grow.

* Flext-shrinlk

Giá trị này sẽ quyết định tỉ lệ co giãn giữa các items khi co giãn màn hình.