

ĐỀ CƯƠNG ĐỀ TÀI NGHIÊN CỨU KHOA HỌC

A. THÔNG TIN CHUNG

Sinh viên (nhóm sinh viên) - MSSV:	Lớp:
Lê Quốc Hải - 2108110118	K15DCPM05
Nguyễn Thị Thanh Hiếu - 2108110393	K15DCPM05
Nguyễn Hoàng Khuyển - 2108110356	K15DCPM05
Nguyễn Vinh Quang - 2108110353	K15DCPM05
Trần Mạnh Tiến - 2108110131	K15DCPM05

Người hướng dẫn: Ths. Lê Huỳnh Phước

Đơn vị công tác: Khoa Công Nghệ Thông Tin Trường Đại học Gia Định

Đề tài: Phân tích và đánh giá các tính năng, hiệu suất và sự phù hợp của các thư viện lập trình Reactive.

B. ĐỀ CƯƠNG ĐỀ TÀI NGHIÊN CỨU KHOA HỌC

1. Tính cấp thiết và ý nghĩa thực tiễn của đề tài

- Tính cấp thiết của đề tài thực hiện:

+ Đối với các mô hình lập trình cũ hơn trước khi Reactive Programming ra đời gặp phải các vấn đề: Khó kiểm soát mã nguồn, lập trình theo cơ chế đồng bộ dẫn tới việc hệ thống phải tạm dừng để xử lý một khối công việc lớn trước khi tiếp tục thực thi, không có sự co giãn hoặc sự co giãn không linh hoạt khiến thời gian xử lý phụ thuộc vào kích thước dữ liệu cần xử lý và bên cạnh đó cũng có thể dẫn tới lãng phí tài nguyên, không có khả năng phục hồi sau khi xảy ra lỗi dẫn đến cả hệ thống ngừng hoạt động, sự liên kết chặt chẽ giữa các thành phần gây ra nhiều khó khăn trong quá trình bảo trì hệ thống và khả năng xảy ra lỗi cao.

+ Mỗi thư viện lập trình Reactive Programming có những ưu, khuyết điểm nhất định và sự phù hợp nhất định. Vì vậy, việc làm rõ sự phù hợp của mỗi thư

viện Reactive Programming và xác định một dự án phần mềm nên triển khai thư viện Reactive Programming nào là cần thiết và quan trọng trong quá trình phát triển một hệ thống phần mềm Reactive.

- Tính mới của đề tài đối với sinh viên (nhóm sinh viên): Khái niệm Reactive Programming và các khái niệm liên quan trong lập trình phần mềm, kiến trúc Reactive Systems, kiến trúc các thư viện Reactive Programming, làm việc thực nghiệm với các thư viện lập trình Reactive Programming.

- Nguyên nhân tìm hiểu và ý nghĩa của đề tài được thực hiện:

- + Nguyên nhân: Khái niệm về Reactive Programming vẫn còn mơ hồ và trừu tượng, chưa được làm rõ, việc ứng dụng thư viện Reactive Programming phù hợp cho một hệ thống phần mềm Reactive là cần thiết và quan trọng.

- + Ý nghĩa: Làm rõ được khái niệm về Reactive Programming, so sánh ưu, khuyết điểm, đánh giá được sự phù hợp của mỗi thư viện Reactive Programming trong các ngữ cảnh khác nhau.

2. Nhiệm vụ thực hiện của đề tài

Các nhiệm vụ đặt ra khi thực hiện đề tài bao gồm:

- Nhiệm vụ 1: Tìm hiểu và giải thích được Reactive Programming là gì ?

- Nhiệm vụ 2: Phân tích và đánh giá các ưu nhược điểm của Reactive Programmng so với các công nghệ khác.

- Nhiệm vụ 3: Tìm hiểu về các thư viện lập trình Reactive. (Ít nhất giải thích được nó là gì ?, mục đích của nó ?, kiến trúc của nó ? và cơ chế hoạt động của nó ?)

- Nhiệm vụ 4: Từ những gì đã tìm hiểu được từ các thư viện Reactive kể trên, tiến hành phân tích, đánh giá ưu, nhược điểm của mỗi thư viện Reactive đã tìm hiểu.

- Nhiệm vụ 5: Từ những ưu nhược điểm cũng như những gì đã tìm hiểu được về các thư viện Reactive kể trên, tiến hành phân tích sự phù hợp của mỗi thư viện Reactive đã tìm hiểu trong nhiều ngữ cảnh.

3. Phương pháp thực hiện đề tài

- Đối tượng tìm hiểu của đề tài:

+ Đối tượng chính: Các thư viện Reactive Programming.

+ Đối tượng liên quan: Reactive Programming

- Thời gian thực hiện đề tài dự kiến: 60 ngày

- Kinh phí thực hiện dự kiến: 1.500.000 VNĐ

- Dự kiến kết quả đạt được: Xây dựng được các ứng dụng để thực nghiệm về các thư viện Reactive Programming để phân tích và đánh giá được hiệu suất, ưu, nhược điểm của mỗi thư viện Reactive Programming và từ đó đánh giá được sự phù hợp của mỗi thư viện Reactive Programming trong các ngữ cảnh khác nhau.

4. Dự kiến các phần, các mục

Sau các mục “Lời cảm ơn”, “Lời cam đoan”, “Tóm tắt đề tài”, “Mục lục”, “Bảng tra cứu thuật ngữ, từ viết tắt”, “Danh mục hình ảnh”, “Danh mục các bảng” và “Mở đầu” là nội dung chính của đề tài với kết cấu như sau:

CHƯƠNG 1: TỔNG QUAN VỀ ĐỀ TÀI NGHIÊN CỨU KHOA HỌC

1.1 Giới thiệu tổng quan đề tài

1.2 Nhiệm vụ đề tài

1.3 Cấu trúc đề tài

CHƯƠNG 2: CÁC CƠ SỞ LÝ THUYẾT

2.1 Tổng quan về reactive programming

2.1.1 Giới thiệu về Reactive Programming

2.1.2 Tình hình phát triển của Reactive Programming

2.2 Lý thuyết về reactive programming

2.2.1 Định nghĩa

2.2.1.1. Reactive là gì ?

2.2.1.2. Cách tiếp cận của Reactive Programming ?

2.2.1.3. Phương pháp triển khai mô hình Reactive Programming ?

2.2.1.4. Sẽ như thế nào nếu không có Reactive Programming ?

2.2.2 Đánh giá ưu nhược điểm của Reactive Programming ?

2.2.2.1. Ưu điểm của Reactive Programming ?

2.2.2.2. Nhược điểm của Reactive Programming ?

2.2.3 So sánh giữa Reactive Programming với các phương pháp lập trình khác

2.2.3.1. Imperative Programming

2.2.3.2. Functional Programming

2.2.4 Ứng dụng thực tế

2.3 Các thư viện của RP

2.3.1 Spring Web Flux

2.3.1.1 Giới thiệu

2.3.1.2 Kiến trúc

2.3.1.3 Cơ chế hoạt động

2.3.1.4 Hiệu suất

2.3.2 RxJava

2.3.2.1 Giới thiệu

2.3.2.2 Kiến trúc

2.3.2.3 Cơ chế hoạt động

2.3.2.4 Hiệu suất

2.3.3 So sánh giữa các thư viện đã liệt kê

2.3.4 Sự phù hợp của các thư viện trong các ngữ cảnh khác nhau.

CHƯƠNG 3: MÔ HÌNH THỰC NGHIỆM

3.1 Giải thích mô hình thực nghiệm

3.2 Mục tiêu thực nghiệm

3.3 Các bước thực nghiệm

3.4 Kết quả đạt được

CHƯƠNG 4: KẾT LUẬN VÀ HƯỚNG PHÁT TRIỂN

4.1 Kết luận

4.2 Hướng phát triển của đề tài

Tài liệu tham khảo để xây dựng đề cương

[1] ...

[2] ...

[3] ...

Tp.Hồ Chí Minh , ngày ... tháng ... năm ...

**XÁC NHẬN CỦA
NGƯỜI HƯỚNG DẪN**
(Ký, ghi rõ họ tên)

**SINH VIÊN
THỰC HIỆN ĐỀ TÀI**
(Ký, ghi rõ họ tên)