**National College of Ireland**

Higher Diploma in Science in Computing

**Distributed Systems**

**Terminal Based Assignment Assessment**

**(TABA)**

Saturday 4th May, 09:00 am to Tuesday 8th May, 23:55 pm

**Lecture: Mark Cudden**

Repository:

<https://github.com/sergio-oliveira-br/distsys>

Sergio Vinicio da Silva Oliveira

[**x23170981@student.ncirl.ie**](mailto:x23170981@student.ncirl.ie)

**1. Consider that you are designing a "Product Management" system that supports at least four operations.**

* **Explain and differentiate between the four kinds of service methods supported in gRPC.**
* **Include appropriate proto file definitions and sample code in your explanations.**
* **How would each type of method be used?**
* **How would these methods be defined in the proto file?**
* **What would the corresponding code look like?**

Just a little introduction.

In gRPC, it is possible for a client application to directly call a method in a server application on a different machine as if it were a local object, which simplifies the creation of distributed applications and services.

Is based around the idea of defining a service, specifying the methods, where I will explore a lit bit more, that can be called remotely with their parameters and return types.

The image below is a representation of the concept of connection.

A diagram of a software process

Description automatically generated

gRPC (2023) *Introduction to gRPC*. Available at: <https://grpc.io/docs/what-is-grpc/introduction/> [Accessed 7 May 2024].

Implementing the gRPC concept.

Keep in mind that the theme will be built around a bookstore, and these are the attributes of the object of study.

A close-up of a book

Description automatically generated

The image below represents the implementation of the SalesServices service using the Protocol Buffer version proto3, line 16.

To define a service in gRPC, we proceed as described in line 19. Then, the rpc methods can be defined in our service definition by specifying their request and response types, as we can see in line 22.

This service, in particular, addresses the idea of Unary RPC, aiming to provide the user with details about a particular book.

As previously mentioned, the proto file also contains protocol buffer message type settings for all request and response types used in our service methods. And through line 26 represented the client side, and line 29 represented the server side, we establish these definitions.

The user request the information about one specific book using ID. Then, the server responds returning a information type String.

A computer screen shot of text

Description automatically generated

Introducing another service implementation, this one focused on Client-Side Stream and Bidirectional Stream.

Through the image below, we can observe a structure similar to that previously mentioned. But this time the service definition has two different methods.

Line 23 illustrates the implementation of the Client-Side Stream, and in line 27 Bidirectional Stream implementation.

In both instances, the methods' definitions contain the keyword “stream”, which means the existence of transmission.

The method addBookImages( ) proposes the idea that the user can send a collection of images related to a specific book to the server, and the server will respond with a single response. This concept is related to the Client-Side Stream approach.

Nonetheless, the method insertBulkDiscounts( ), allows the user to make numerous adjustments to the individual prices on many books, the server then responds individually to each request.

It's important to point out that the proposal is designed to accommodate a large number of individuals.

To illustrate the idea, consider that a 10% reduction will be implemented in all books in a certain segment.

In contrast to the Client-Side Stream, Bidirectional approach, allows the server responds with a single response. This time, we will be given individual confirmation for each adjustment made, as well as the freedom of individual adjustments.

A computer screen shot of text

Description automatically generated

Start the last PROTO

A screen shot of a computer program

Description automatically generated

~~To apply the concepts presented in the classroom, using the idea of a smart home as a basis, I chose to approach the four gRPC services. Also, in an attempt to improve and reinforce my concepts in gRPC services, and assuming that practice leads to perfection, more than one version was built to explore the Java Swing and JavaFX topics.~~

~~The first two versions differ only in the graphical presentation, where one addresses command line concepts, and the second addresses the GUI graphical user interface through the Java Swing features.~~

**~~Please consider the third version as the main one to be evaluated. It includes GUI addressing JavaFX.~~**