Project Proposal

Rachel Roche

Student Number: 23309814

OVERVIEW

This document is submitted as an interim submission of CA project of the Distributed Systems module on the Higher Diploma in Computing specializing in Software Development delivered by Yasantha Samarawickrama.

Domain description

You should best describe the overall purpose of the service, explain the functionalities within each service and overall contribution of the service to the application
[200 words]

4. Retail (Personalized shopping recommendations, automated checkout, inventory tracking)

Rachel's Shoe Shop

The application will provide an online shopping experience for shoes. It will be made up of a number of services to provide an end to end shopping experience:

- ListShoes – Returns details on all shoes currently stocked by Rachel's Shoe shop.
- GetPrice – Returns the price of a specific shoe.
- ShoppingCart – Will add a specific shoe to the user's shopping cart.
- ViewCart – Will return the list of shoes in the user's shopping cart.
- Purchase – Will purchase the shoes in the user's shopping cart.
- Chat – Help facility if the user has any questions.

Each user of the application will need a unique identifier to keep track of items added to their personal shopping cart. When the user wishes to purchase a pair or multiple pairs of shoes, the add each pair to the shopping cart. The user can view the contents of their shopping cart. When the user chooses to Purchase, the items in their shopping cart are processed, and the shopping cart contents is reverted to empty.

Service definition and RPC

You should explain in detail, with example the request and response for each functionality within the service. Explain in detail the parameters
[300 words]

ListShoes

This is a server rpc service. No parameters are passed in the request. It will return a stream of ProductResponse objects, which will represent the shoes stocked by the shop. Each ProductResponse object will return the brand and the price of the shoe.

GetPrice

This is a unary rpc service. The user will provide the brand of the shoe which will be passed as the only parameter in the ProductRequest. The service will return a single ProductResponse object which will contain the brand and the price of the shoe.

ShoppingCart

This is a client rpc service which will pass in a stream of ShoppingCartRequest objects. The user will add one or more items to their shopping cart, and the content of the cart will be streamed to the server. Each ShoppingCartRequest object will contain the brand, size, color and quantity added to the cart. Once all items have been added to the cart, a message will be returned to indicate success, along with a total item count of objects added to the shopping cart.

ViewCart

This is a server rpc call. The cartId will be passed in from the client in the ViewCartRequest, and the server will return a stream of all items in the cart in the ViewCartResponse.

Purchase

This is unary rpc call. The cartId will be passed in from the client in the PurchaseRequest, and the server will return the PurchaseResponse which will contain a message with the purchase details. The user's cart will also be cleared after the contents have been purchased.

Chat

This is a bidirectional rpc call which will enable the user to engage with the customer service team if any issues are encountered while shopping.

Shoe Shop Proto Definition

```
syntax = "proto3";

package shop;

service ShoeShop{

  //Unary Rpc
  rpc GetPrice(ProductRequest) returns (ProductResponse);
  rpc Purchase(PurchaseRequest) returns (PurchaseResponse);

  //Server Rpc
  rpc ListShoes(Empty) returns (stream ProductResponse);
  rpc ViewCart(Empty) returns (stream ViewCartResponse);

  //Client rpc
  rpc      ShoppingCart(stream      ShoppingCartRequest)       returns
(ShoppingCartResponse);

  //Bidirectional
  rpc Chat(stream ChatMessage) returns (stream ChatMessage);
}
```

```
message ProductRequest{
  string brand = 1;
}

message ProductResponse{
  string brand = 1;
  uint32 price = 2;
}

message ViewCartRequest {
  string cartId = 1;
}

message ViewCartResponse {
  string brand = 1;
  uint32 size = 2;
  string color = 3;
  uint32 quantity = 4;
}

message PurchaseRequest {
  string cartId = 1;
}

message PurchaseResponse {
  string message = 1;
}

message Empty{}

message ShoppingCartRequest {
  string brand = 1;
  string color = 2;
  string size = 3;
  uint32 quantity = 4;
}

message ShoppingCartResponse {
  string message = 1;
}

message ChatMessage{
  string user = 1;
  string message = 2;
}
```