



National
College^{of}
Ireland

Distributed Systems
March 2022

Joanna Goch
ID 21137641

Contents

1	Introduction	4
2	Service 1: Patient Health Data Service.....	4
2.1	Methods	4
2.1.1	RPC Method 1: Add Prescription	4
2.1.2	RPC Method 2: Get Prescriptions	4
2.1.3	RPC Method 3: Add Lab Results.....	4
2.1.4	RPC Method 3: Add Doctor Notes	5
3	Service 2: Permission & Authentication	5
3.1	Methods	5
3.1.1	RPC Method 1: Give Data Access.....	5
3.1.2	RPC Method 2: Remove Data Access.....	5
3.1.3	RPC Method 3: Get Access History	5
4	Service 3: Technical Support Service	5
4.1	Methods	6
4.1.1	RPC Method 1: Request Phone Call	6
4.1.2	RPC Method 2: Queue Up for Chat	6
4.1.3	RPC Method 3: Interactive Chat.....	6

1 Introduction

This proposal outlines a part of a service in Smart health/medical environment.

A central service is storing patient data allowing any medical team as well as the patient to access the data easily using apps. A program that implements the service client interfaces correctly would be able to interact with the central patient data storage service.

A patient can have a health card that stores the patient ID. A doctor can scan the card which allows the app to find the patient in the central database quickly and easily.

2 Service 1: Patient Health Data Service

Storing and retrieving health data of a specific patient.

2.1 Methods

2.1.1 RPC Method 1: Add Prescription

Adds one prescription for a patient to the database.

Input message contains doctor ID, patient ID, date, prescription text.

Return message contains Boolean to confirm successful completion.

Type: Unary

2.1.2 RPC Method 2: Get Prescriptions

Gets open prescriptions for a patient from the database.

Input message contains patient ID.

Returns multiple messages containing information such as doctor ID, patient ID, date, prescription text for each open prescription.

Type: Server Streaming (StreamingFromServer)

2.1.3 RPC Method 3: Add Lab Results

Adds one or multiple results from patient's laboratory health check to the database.

Input is a stream of messages containing information such as doctor ID, patient ID, date, lab type, results, diagnosis.

Return message contains Boolean to confirm successful completion.

Type: Client Streaming (StreamingFromClient)

2.1.4 RPC Method 3: Add Doctor Notes

Adds a note from health professional to the database.

Input message contains doctor ID, patient ID, date, diagnosis, recommendations, notes.

Return message contains Boolean to confirm successful completion.

Type: Unary

3 Service 2: Permission & Authentication

This service controls who can access what data in the database. It allows the patient to grant or revoke access.

3.1 Methods

3.1.1 RPC Method 1: Give Data Access

Gives read and write access to an account.

Input message contains patient ID and doctor ID.

Return message contains Boolean to confirm successful completion.

Type: Unary

3.1.2 RPC Method 2: Remove Data Access

Removes read and write access to an account.

Input fields are patient ID and doctor ID.

Return message contains Boolean to confirm successful completion.

Type: Unary

3.1.3 RPC Method 3: Get Access History

Returns a list of access log. Input fields are patient ID.

Returns a list of log access items each with date, doctor ID, access data type, operation type.

Type: Unary

4 Service 3: Technical Support Service

Service that allows client applications to expose the technical support service given by the team maintaining the central health database solution.

4.1 Methods

4.1.1 RPC Method 1: Request Phone Call

Request to get a phone call from the Technical Support Service.

Input is a message that contains: user id, subject, description, phone number, preferred time.

Return message contains Boolean to confirm successful completion.

Type: Unary

4.1.2 RPC Method 2: Queue Up for Chat

This puts the client app into the queue for the interactive technical support chat. The queue information is updated with how many support requests are ahead of the user.

Input is a message containing user id, subject.

Return messages contain the number of users ahead in the queue.

Type: Server Streaming (StreamingFromServer)

4.1.3 RPC Method 3: Interactive Chat

Streams chat messages from the client to the server and from the server to the client.

Input is a stream of chat messages and return is also a stream of chat messages.

Type: bidirectional streaming (StreamingBothWays)