```go
package main

import (
        "fmt"
        "github.com/google/uuid"
        "log"
        "math/rand"
        "net/rpc"
        "time"
)

// Structs dùng chung giữa Client, LoadBalancer và Server
type GetRequest struct {
        Bucket string
        Key    int
}
type GetReply struct {
        Success bool
        Err     error
        Data    []byte
}

type SetRequest struct {
        Bucket string
        Key    int
        Data   []byte
}
type SetReply struct {
        Success bool
        Err     error
}

type DeleteRequest struct {
        Bucket string
        Key    int
}
type DeleteReply struct {
        Success bool
        Err     error
}

type GetAllRequest struct {
        Bucket string
}
type GetAllReply struct {
        Success bool
        Data    map[int][]byte
        Err     error
}

type RemoveServerRequest struct {
        Address string
}
```

```
54
55  type RemoveServerReply struct {
56          Success bool
57          Message string
58  }
59
60  // LoadBalancerClient giúp giao tiếp với LoadBalancer
61  type LoadBalancerClient struct {
62          client *rpc.Client
63  }
64
65  // NewLoadBalancerClient tạo kết nối với LoadBalancer
66  func NewLoadBalancerClient(address string) (*LoadBalancerClient, error) {
67          client, err := rpc.DialHTTP("tcp", address)
68          if err ≠ nil {
69                  return nil, err
70          }
71          return &LoadBalancerClient{client: client}, nil
72  }
73
74  // Set gửi dữ liệu đến LoadBalancer
75  func (lb *LoadBalancerClient) Set(bucket string, key int, value []byte) error
    {
76          log.Printf("Client sending Set request - Bucket: %s, Key: %d",
    bucket, key)
77
78          req := &SetRequest{Bucket: bucket, Key: key, Data: value}
79          reply := &SetReply{}
80          err := lb.client.Call("LoadBalancer.Set", req, reply)
81
82          if err ≠ nil {
83                  log.Printf("Client failed to send Set request: %v", err)
84                  return err
85          }
86
87          log.Printf("Client received response: Success = %v", reply.Success)
88          return nil
89  }
90
91  // Get lấy dữ liệu từ LoadBalancer
92  func (lb *LoadBalancerClient) Get(bucket string, key int) ([]byte, error) {
93          req := &GetRequest{Bucket: bucket, Key: key}
94          reply := &GetReply{}
95          err := lb.client.Call("LoadBalancer.Get", req, reply)
96          if err ≠ nil {
97                  return nil, err
98          }
99          return reply.Data, nil
100 }
101
102 func (lb *LoadBalancerClient) SetMultipleData(randomId bool) {
103
104         for i := 1; i ⩽ 10; i++ {
105                 var id int
106                 if randomId {
```

```
107                         id = rand.Intn(1000) + 1 // ID từ 1 đến 1000
108                 } else {
109                         id = i
110                 }
111                 uuidStr := uuid.New().String()
112                 email := fmt.Sprintf("user%d_%d@example.com", id, id)
113
114                 data := fmt.Sprintf(`{"ID":%d,"UUID":"%s","Email":"%s"}`, id,
    uuidStr, email)
115                 log.Printf("Client sending Set request - Bucket: user, Key:
    %d", i)
116
117                 req := &SetRequest{Bucket: "user", Key: id, Data:
    []byte(data)}
118                 reply := &SetReply{}
119                 err := lb.client.Call("LoadBalancer.Set", req, reply)
120
121                 if err ≠ nil {
122                         log.Printf("Client failed to send Set request: %v",
    err)
123                         continue
124                 }
125
126                 log.Printf("Client received response for Key %d: Success =
    %v", i, reply.Success)
127
128                 // Random sleep từ 200ms đến 800ms để mô phỏng request không
    đồng đều
129                 time.Sleep(time.Duration(rand.Intn(600)+200) *
    time.Millisecond)
130         }
131 }
132
133 func (lb *LoadBalancerClient) GetMultipleData() {
134         for i := 1; i ⩽ 10; i++ {
135                 log.Printf("Client sending Get request - Bucket: user, Key:
    %d", i)
136
137                 req := &GetRequest{Bucket: "user", Key: i}
138                 reply := &GetReply{}
139                 err := lb.client.Call("LoadBalancer.Get", req, reply)
140
141                 if err ≠ nil {
142                         log.Printf("Client failed to get data for Key %d:
    %v", i, err)
143                         continue
144                 }
145
146                 if reply.Success {
147                         log.Printf("Client received response for Key %d: %s",
    i, string(reply.Data))
148                 } else {
149                         log.Printf("Client received error for Key %d: %v", i,
    reply.Err)
150                 }
```

```
151
152                 time.Sleep(time.Millisecond * 500)
153         }
154 }
155
156 // RemoveServer yêu cầu LoadBalancer loại bỏ một server
157 func (lb *LoadBalancerClient) RemoveServer(serverID string) error {
158         log.Printf("Client sending RemoveServer request - ServerID: %s",
    serverID)
159
160         req := &RemoveServerRequest{Address: serverID}
161         reply := &RemoveServerReply{}
162         err := lb.client.Call("LoadBalancer.RemoveServer", req, reply)
163
164         if err ≠ nil {
165                 log.Printf("Client failed to send RemoveServer request: %v",
    err)
166                 return err
167         }
168
169         log.Printf("Client received response: Success = %v", reply.Success)
170         return nil
171 }
172
173 func main() {
174         // Kết nối với LoadBalancer
175         lbClient, err := NewLoadBalancerClient("localhost:9000")
176         if err ≠ nil {
177                 log.Fatalf("Error connecting to LoadBalancer: %v", err)
178         }
179         defer lbClient.client.Close()
180
181         // Uncomment to set 10 new data entries
182         //lbClient.SetMultipleData(false)
183         //lbClient.SetMultipleData(true)
184
185         // Uncomment to get 10 data entries
186         //lbClient.GetMultipleData()
187
188         // Uncomment to request remove server
189         serverToRemoveID := "localhost:2"
190         lbClient.RemoveServer(serverToRemoveID)
191 }
192
```