

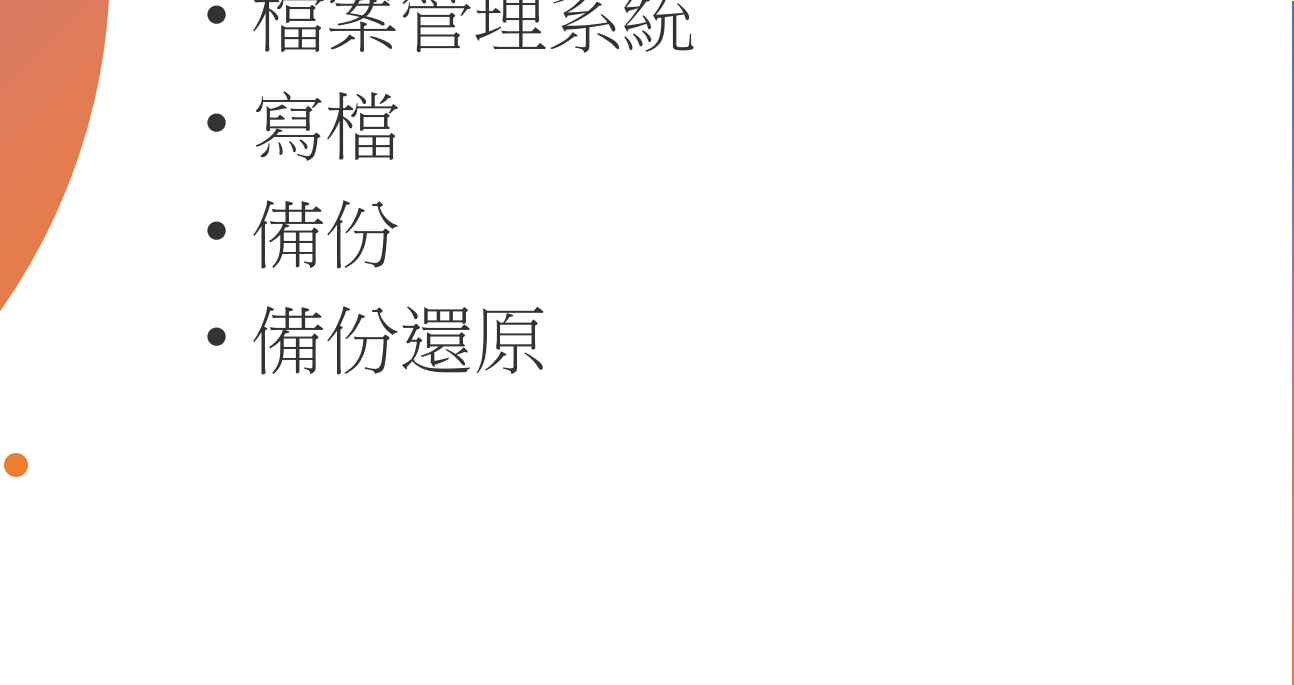
雲端備份系統

毛文佐

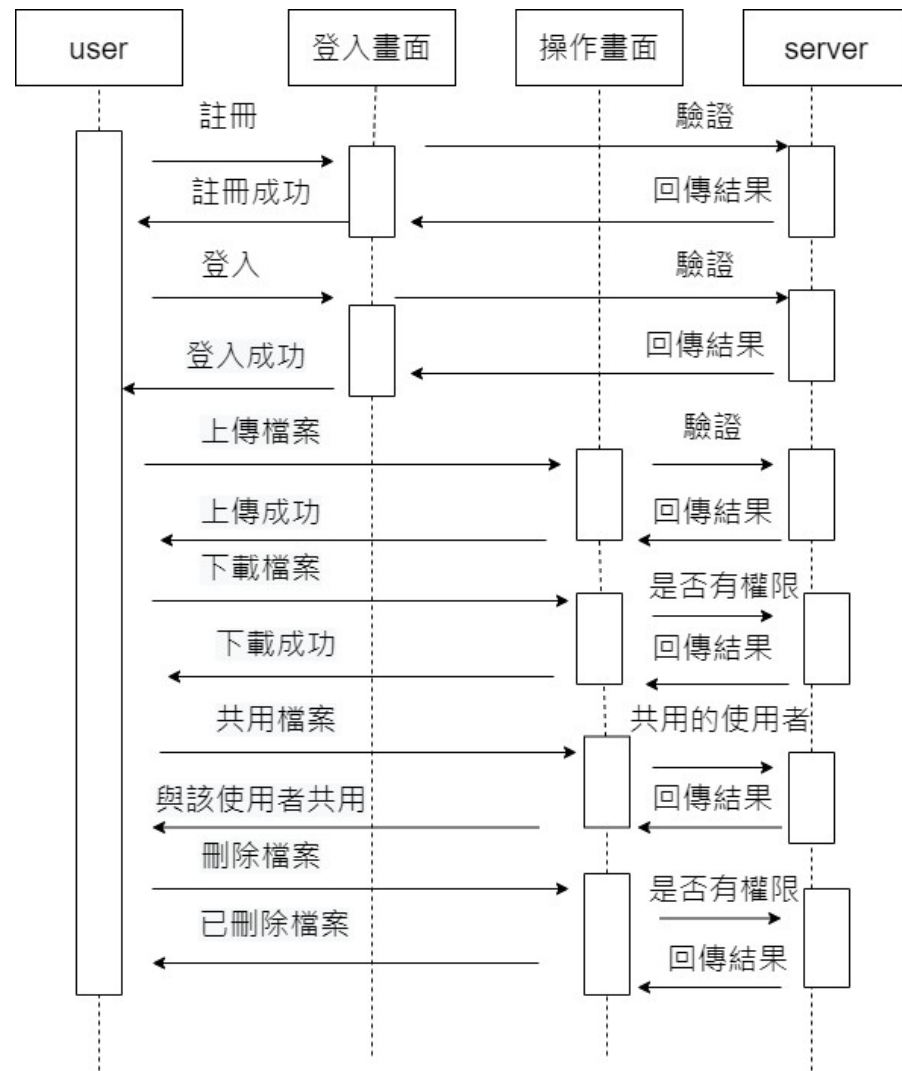
薛竣祐



主要功能

- 上傳
 - 下載
 - 共用
 - 刪除
 - 檔案管理系統
 - 寫檔
 - 備份
 - 備份還原
- 

循序圖



模組功能

- Client 端
 - 上傳模組
 - 下載模組
 - 檔案管理模組
 - 使用者模組
- Server端
 - 上傳API
 - 下載API
 - 檔案管理API
 - 存取限制管控

Json In Java Using Gson

同步JSON到JAVA LIST

```
private void updateList(String jsonFile){
    try {
        Scanner reader = new Scanner(new File(jsonFile));
        if(!reader.hasNextLine()) return;
        String data = reader.nextLine();
        reader.close();
        switch (jsonFile) {
            case USERJSONFILE -> this.users = new Gson().fromJson(data, new TypeToken<ArrayList<User>>() {}.getType());
            case FILEDETAILJSONFILE -> this.fileDetails = new Gson().fromJson(data, new TypeToken<ArrayList<FileDetail>>() {}.getType());
        }
    } catch (FileNotFoundException e) {
        System.out.println("An Error Occurred.");
        e.printStackTrace();
    }
}
```

Json In Java Using Gson

同步JAVA LIST到JSON

```
private void updateJson(String jsonFile) {
    try{
        FileWriter writer = new FileWriter(jsonFile);
        switch (jsonFile) {
            case USERJSONFILE -> writer.write(new Gson().toJson(users));
            case FILEDETAILJSONFILE -> writer.write(new Gson().toJson(fileDetails));
        }
        writer.close();
        System.out.println("Updated File.");
    } catch (IOException e) {
        System.out.println("An Error Occurred.");
        e.printStackTrace();
    }
}
```

資料格式：使用者

```
public class User {  
    private int uid;  
    private String account;  
    private String password;  
    private boolean active;  
  
    public User(int uid, String account, String password, boolean active) {  
        this.uid = uid;  
        this.account = account;  
        this.password = getSHA256(password);  
        this.active = active;  
    }  
}
```

使用者的密碼加密 & 登入

```
public boolean checkLogin(String account, String password){
    return this.isActive() && getAccount().equals(account) && getSHA256(password).equals(getPassword());
}

public static String getSHA256(String input){
    String toReturn = null;
    try {
        MessageDigest digest = MessageDigest.getInstance("SHA-256");
        digest.reset();
        digest.update(input.getBytes(StandardCharsets.UTF_8));
        toReturn = String.format("%064x", new BigInteger(1, digest.digest()));
    } catch (Exception e) {
        e.printStackTrace();
    }
    return toReturn;
}
```


實際JSON檔

```
[
  {
    "uid":0,
    "account":"alvin",
    "password":"8e35c2cd3bf6641bdb0e2050b76932cbb2e6034a0ddacc1d9bea82a6ba57f7cf",
    "active":false
  },
  {
    "uid":1,
    "account":"kevin",
    "password":"8fekofeqd3bf6feijanykob76932c28d28fnhfkoec1d9bea129c0jffjiewdjnc",
    "active":true
  }
]
```

資料格式：檔案

```
public class FileDetail {  
    private String creator;  
    private String[] accessor;  
    private String fileName;  
    private String date;//yyyy-MM-dd HH:mm:ss  
  
    public FileDetail(String creator, String[] accessor, String fileName, String date){  
        this.creator = creator;  
        this.accessor = accessor;  
        this.fileName = fileName;  
        this.date = date;  
    }  
}
```

實際JSON檔

```
[
  {
    "creator": "alvin",
    "accessor": [
      "alvin",
      "kevin",
      "david"
    ],
    "fileName": "PDF_IS_OK.pdf",
    "date": "2021-01-01 at 00:00:01"
  },
  {
    "creator": "kevin",
    "accessor": [
      "kevin"
    ],
    "fileName": "IMAGE_IS_OK.png",
    "date": "2021-01-01 at 01:12:53"
  }
]
```

檔案儲存方式：File Manager

名稱	修改日期	大小	種類
▼  a	今天 上午 5:21	--	檔案夾
 KEYNOTE_IS_OK.key	今天 上午 5:21	104.4 MB	Keynote 簡報
▼  alvin	今天 上午 5:21	--	檔案夾
 PDF_IS_OK.pdf	今天 上午 5:15	77 KB	PDF 文件
▼  kevin	今天 上午 5:20	--	檔案夾
 IMAGE_IS_OK.png	2020年12月15日 下午 10:29	101 KB	PNG 影像
▼  s	今天 上午 5:20	--	檔案夾
 JSON_IS_OK.json	2020年12月15日 下午 10:28	22 byte	JSON Document
 TXT_IS_OK.txt	今天 上午 4:08	84 byte	純文字文件

路徑上dir1支持Linux和Windows系統

```
//TODO System.getProperty() and FILE.Separator
File dir1 = new File( pathname: System.getProperty("user.dir")+File.separator+"cloud"+File.separator+account);
File dir2 = new File( pathname: "./cloud/"+account);
```

ActionListener

ACTIONLISTENER 介面
進行按鈕事件的處理

需要實作
ACTIONPERFORMED()

ActionListener的兩種方式

TODO Button Listener Way1

```
 JButton login,signup;  
public void actionPerformed(ActionEvent e) {  
    if(e.getSource() == login) {...}else if(e.getSource() == signup) {...}  
}
```

TODO Button Listener Way2


```
private final JButton buttonUpload = new JButton( text: "Upload");  
buttonUpload.addActionListener(event -> buttonUploadActionPerformed());  
void buttonUploadActionPerformed(){  
    //do something  
}
```

ActionListener

name :

password :

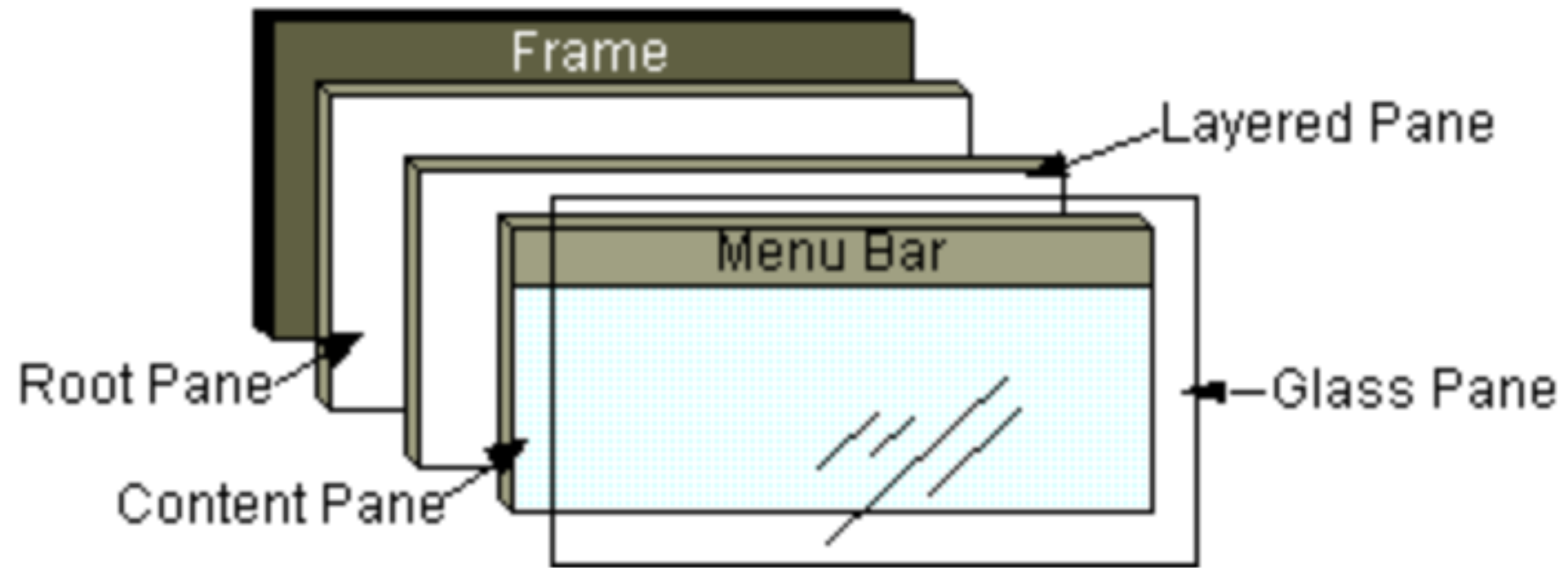
Message ×

 **success**

```
mirror_mod = modifier_ob.  
set mirror object to mirror.  
mirror_mod.mirror_object  
operation == "MIRROR_X":  
mirror_mod.use_x = True  
mirror_mod.use_y = False  
mirror_mod.use_z = False  
operation == "MIRROR_Y":  
mirror_mod.use_x = False  
mirror_mod.use_y = True  
mirror_mod.use_z = False  
operation == "MIRROR_Z":  
mirror_mod.use_x = False  
mirror_mod.use_y = False  
mirror_mod.use_z = True  
  
selection at the end -add  
mirror_ob.select= 1  
modifier_ob.select=1  
context.scene.objects.active  
("Selected" + str(modifier_ob.  
mirror_ob.select = 0  
= bpy.context.selected_object  
data.objects[one.name].select  
  
print("please select exactly  
  
-- OPERATOR CLASSES ----  
  
types.Operator):  
X mirror to the selected  
object.mirror_mirror_x"  
mirror X"  
  
context):  
context.active_object is not
```

SOME TECH

JAVA Swing



Swing Frame 2 ways

```
public static void main(CloudInterface o, String account) {  
    try {  
        UIManager.setLookAndFeel(UIManager.getSystemLookAndFeelClassName());  
    } catch (Exception ex) {  
        ex.printStackTrace();  
    }  
  
    SwingUtilities.invokeLater(() -> new Page_Home(o, account).setVisible(true));  
}
```

呼叫繼承Frame的class中的main Func

```
Page_Home.main(o, account);
```

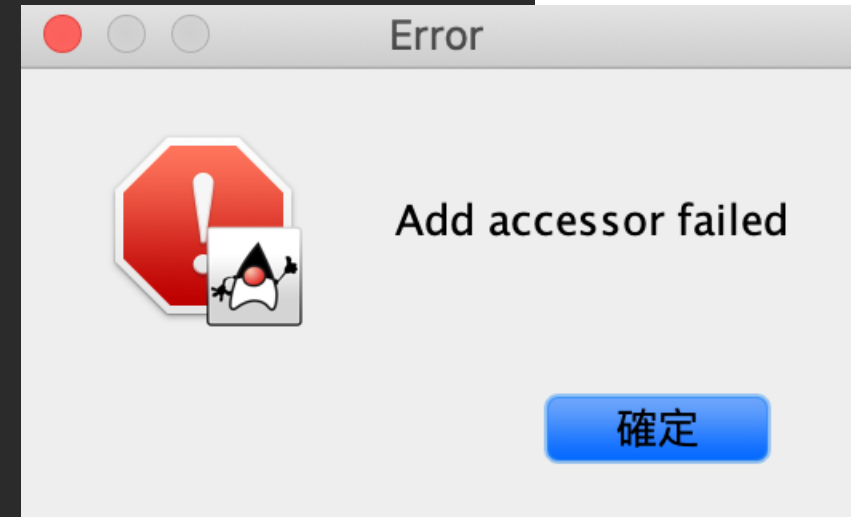
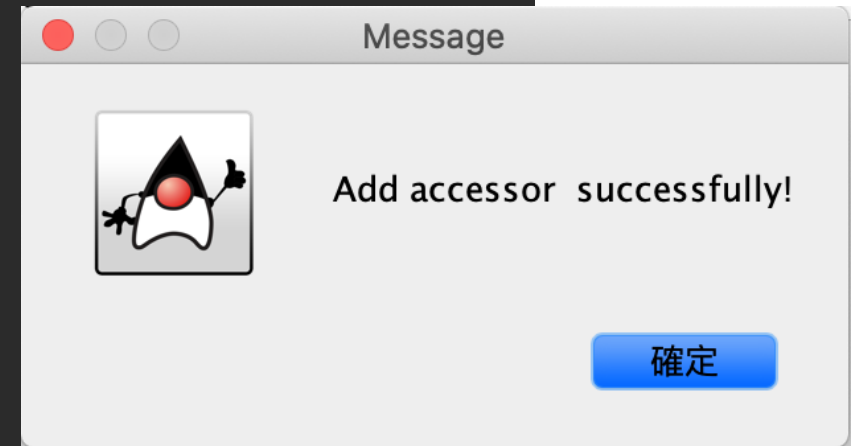
```
new Page_Login(o);
```

呼叫繼承Frame的class中的init Func

```
public Page_Login(CloudInterface o) {  
    this.o = o;  
    this.setResizable(false);  
    this.setLayout(new FlowLayout());  
    this.add(jp1); this.add(jp2); this.add(jp3);  
    this.setSize( width: 400, height: 300);  
    this.setLocationRelativeTo(null);  
    this.setVisible(true);  
    this.setDefaultCloseOperation(JFrame.EXIT_ON_CLOSE);  
}
```

Swing Dialog : Message

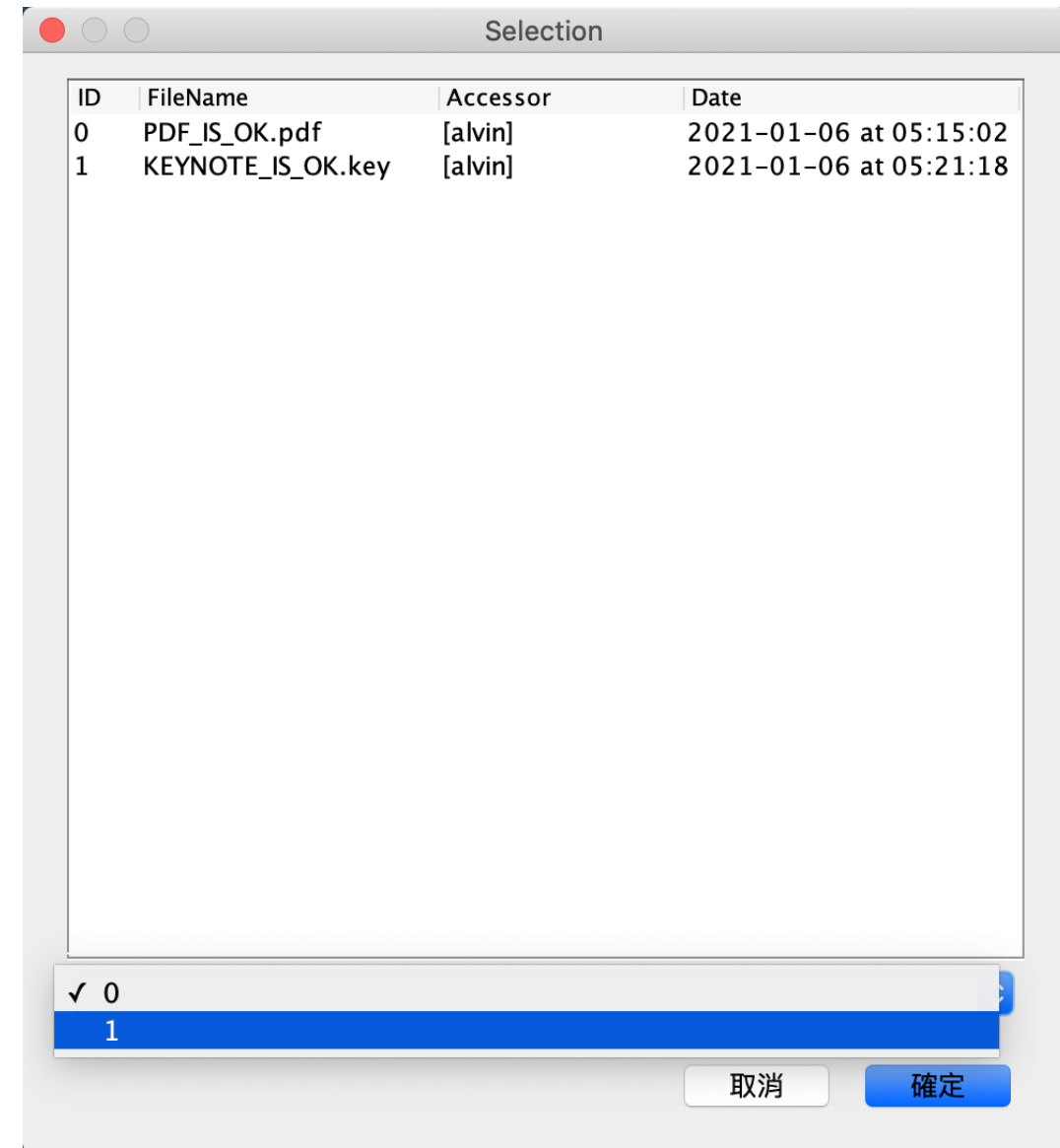
```
if (o.addAccessor(account, myFiles[choose][1].toString(), addAccessor)) {  
    //TODO MessageDialog1  
    JOptionPane.showMessageDialog(  
        parentComponent: this,  
        message: "Add accessor successfully!",  
        title: "Message",  
        JOptionPane.INFORMATION_MESSAGE  
    );  
} else {  
    //TODO MessageDialog2  
    JOptionPane.showMessageDialog(  
        parentComponent: this,  
        message: "Add accessor failed",  
        title: "Error",  
        JOptionPane.ERROR_MESSAGE  
    );  
}
```



Swing Dialog : Input1

TODO InputDialog1

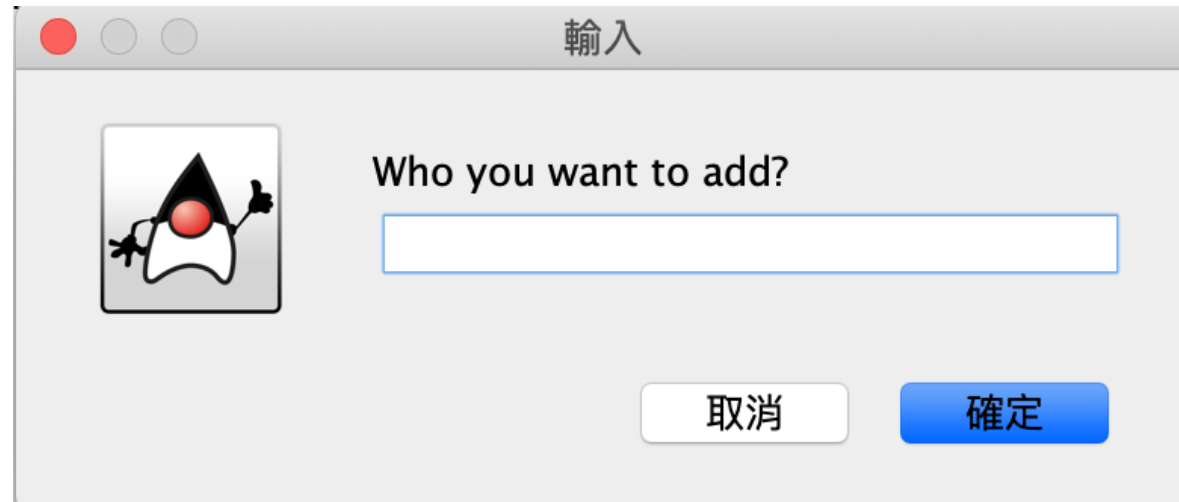
```
Object selected = JOptionPane.showInputDialog(  
    parentComponent: null,  
    new JScrollPane(table),  
    title: "Selection",  
    JOptionPane.PLAIN_MESSAGE,  
    icon: null,  
    values,  
    initialSelectionValue: "0"  
);
```



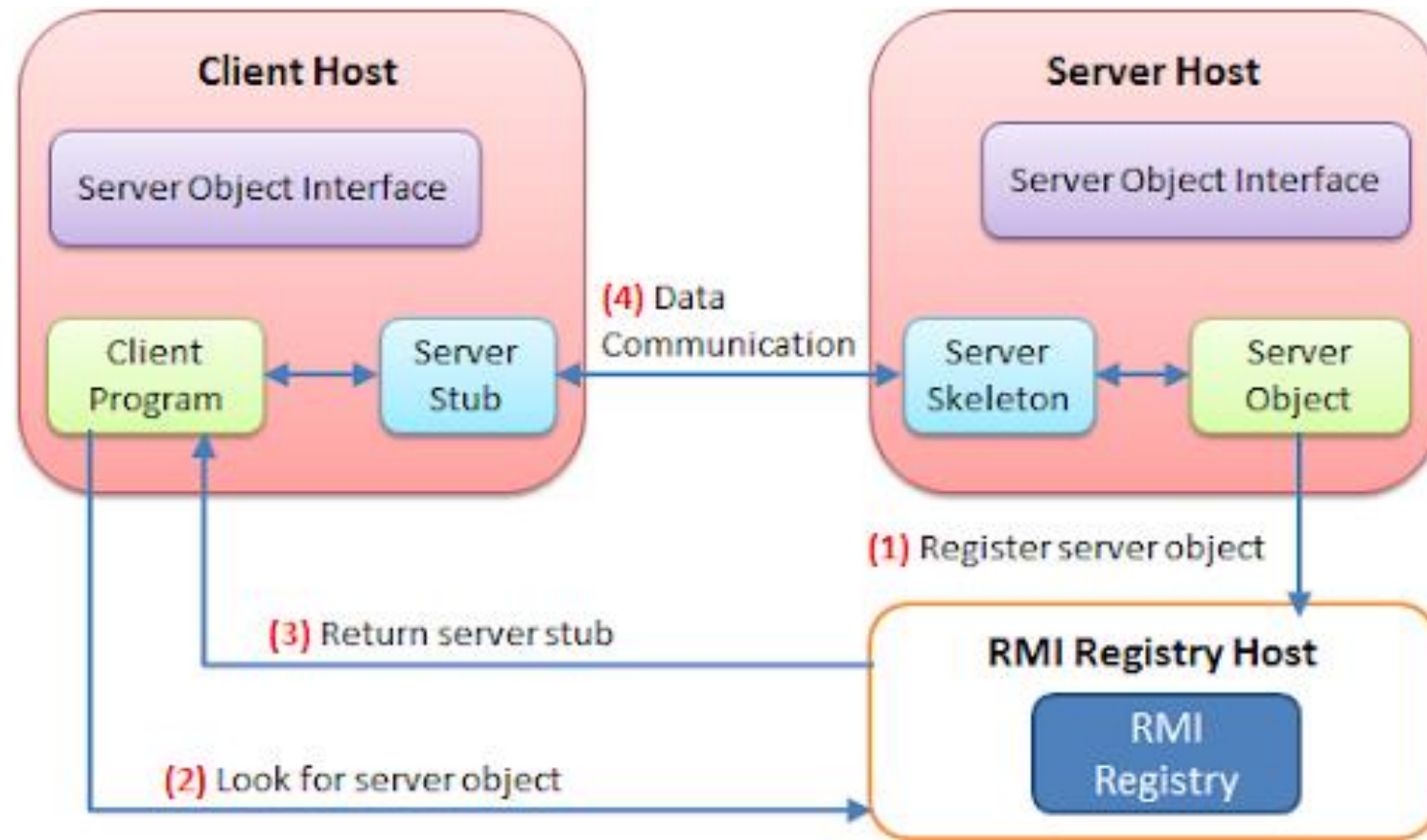
Swing Dialog : Input2

TODO InputDialog2

```
String addAccessor = JOptionPane.showInputDialog( parentComponent: this, message: "Who you want to add?");
```



RMI



RMI (Server Host)

TODO old way

```
CloudInterface name = new RMIImpl();  
System.out.println("Registering ...");  
Naming.rebind(name: "arithmetic", name);  
System.out.println("Register success");
```

TODO rmi on server

```
CloudInterface name = new RMIImpl();  
LocateRegistry.getRegistry(1099); //connect to exist registry  
Registry registry = LocateRegistry.createRegistry(port: 1099); //create registry on server  
System.out.println("Registering ...");  
registry.bind(name: "CloudServer", name);  
System.err.println("Register success");
```

同一主機上，可省去多開一個服務的過程

RMI (Client Host)

```
public class Client {  
    public static void main (String []args) {  
        CloudInterface    o = null;  
        try {  
            Registry registry = LocateRegistry.getRegistry( host: null);  
            o = (CloudInterface) registry.lookup( name: "CloudServer");  
            System.out.println("RMI server connected");  
        } catch (Exception e) {  
            System.out.println("Server lookup exception: " + e.getMessage());  
        }  
        new Page_Login(o);  
    }  
}
```

建立完連線，直接開啟Login畫面

DEMO