目　　录

银行储蓄管理系统

一银行储蓄系统需求分析

系统功能简介：

信息系统：主要是在里面输入用户信息(户名，帐号，开户日期)

主要功能：记录用户所要进行的各种存取操作(存钱，取钱),并对操作数据做好记录

记录时间：主要是记录每个用户开户，存取，取钱的日期

相关金额：该用户的存款金额，取款金额，执行操作后账户余额

保存系统：可以以连接数据库模式保存查询过的内容，对于刚刚查询过的内容不必重复登陆

工作环境：该程序可用在各种银行性质的单位，能有效管理用户信息。方便，快捷，容易上手，安全保密，资料齐整

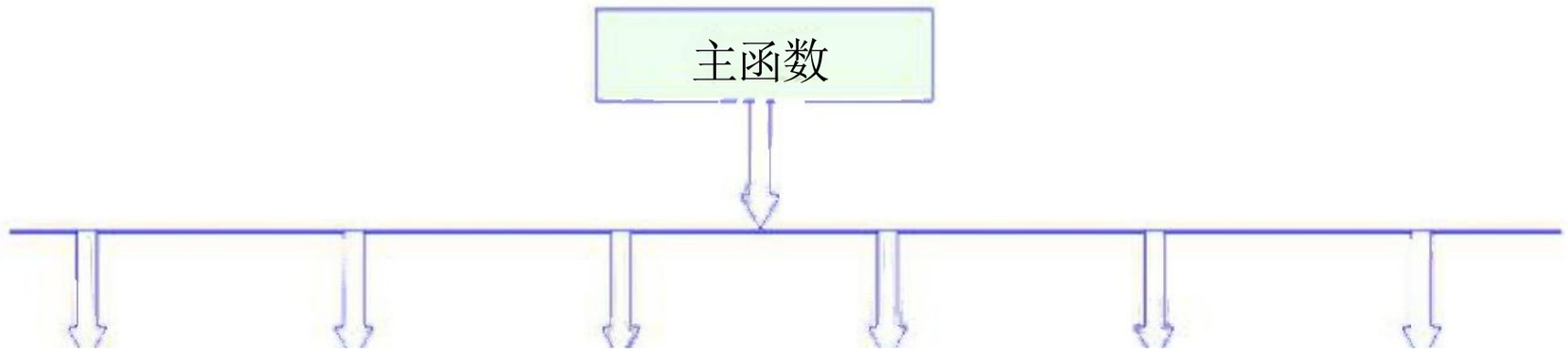
构造该程序，主要是使用C++&SQL　系统。在MS-DOS 以 及WINDOWS95 以上的操作系统上可以正常运行。

现今的社会，资金流动十分频繁。不单单是企业、厂商，连个人也不例外。银行作为一个金融机构，在现代人们的生活中扮演着及其重要的角色。为生活节奏飞快的现代人提供快速、便捷、高效理财服务，是每一个银行机构的共同职责。伴随着电脑技术的发展，各大银行储蓄管理软件也随之出现在这一舞台之上。

银行储蓄管理程序的主要功能就是记录用户的账户信息，已经对用户的存取款操作作好记录及数据更新。

银行储蓄管理的特点是数据量大。数据更新频繁。因此便捷的操作，数据更新准确度，成了这一系统的主要指标。

二系统总框图

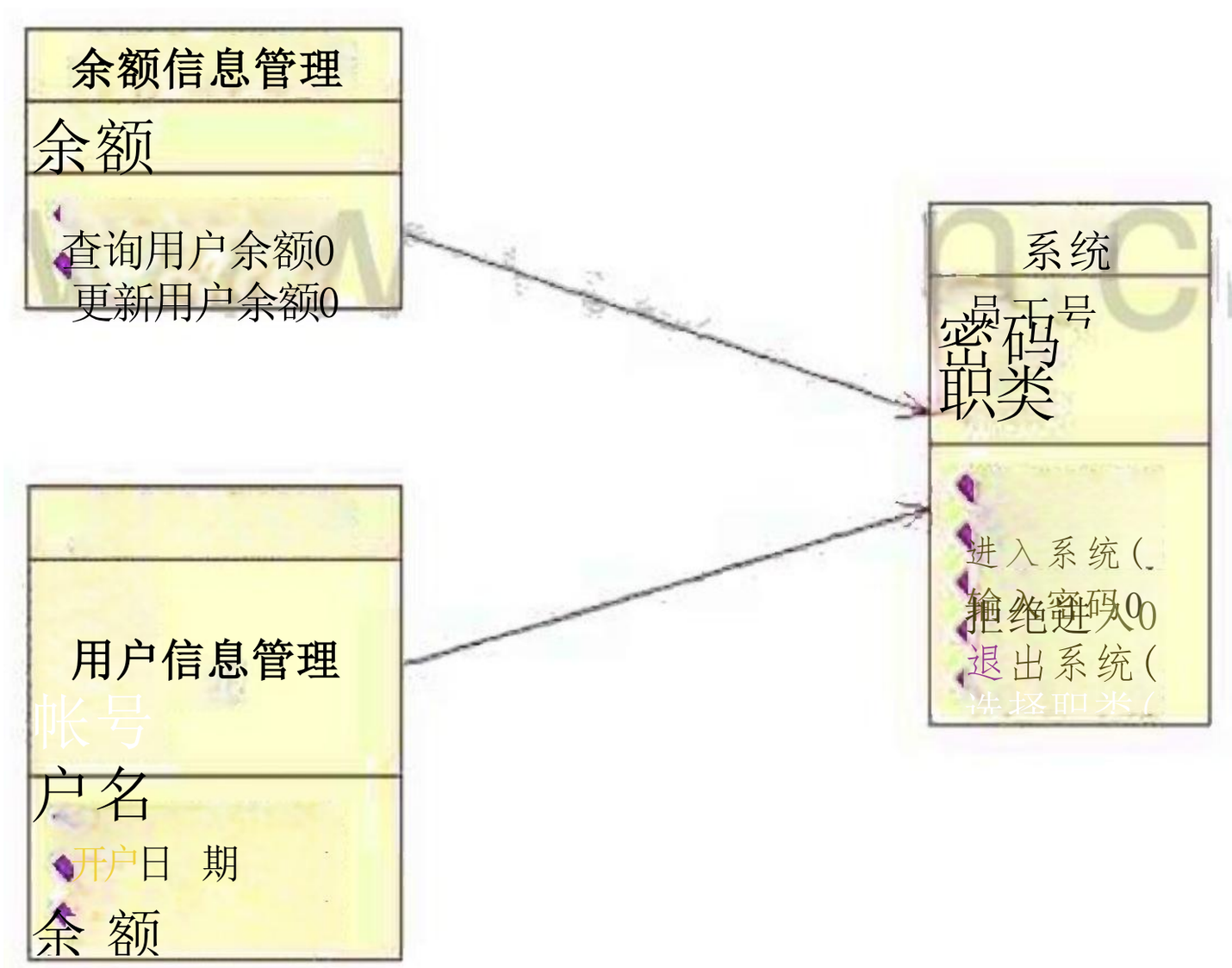


三逻辑设计

任何建模语言都以静态建模机制为基础，标准建模语言UML也不例外。所谓静态建模是指对象之间通过属性互相联系，而这些关系不随时间而转移。类和对象的建模，是UML建模的基础。

面向对象的开发方法的基本任务是建立对象模型，是软件系统开发的基础。UML中的对象类图表达了对象模型的静态结构，能够有效地建立专业领域的计算机系统对象模型。

图 1 系统类图



动态模型主要描述系统的动态行为和控制结构。如图中所示，三个序列图和两个协作图表现各用例与类的对象之间的动态合作关系以及合作过程中的行为次序，描述了一个用例的行为。

图2银行储蓄管理系统序列图

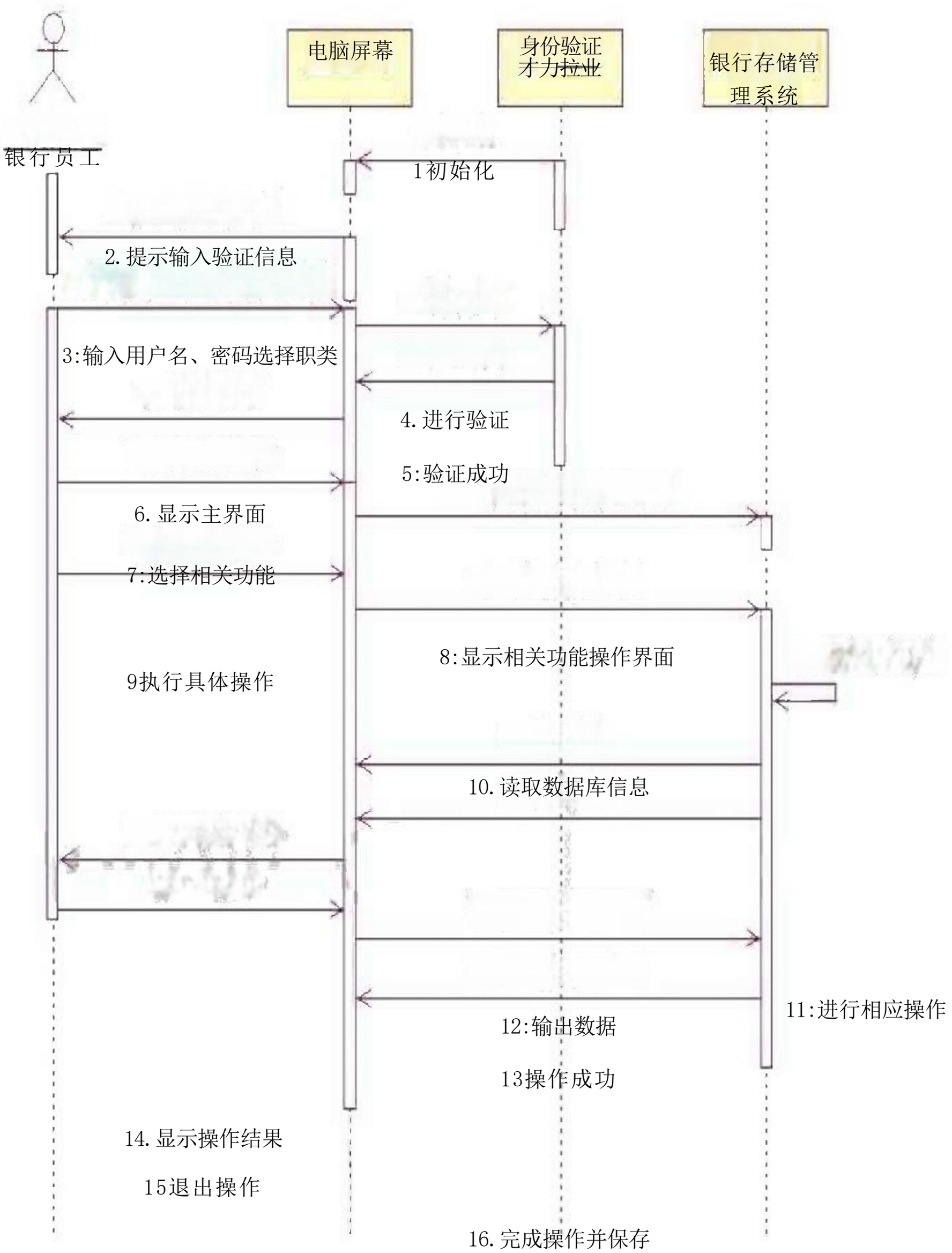
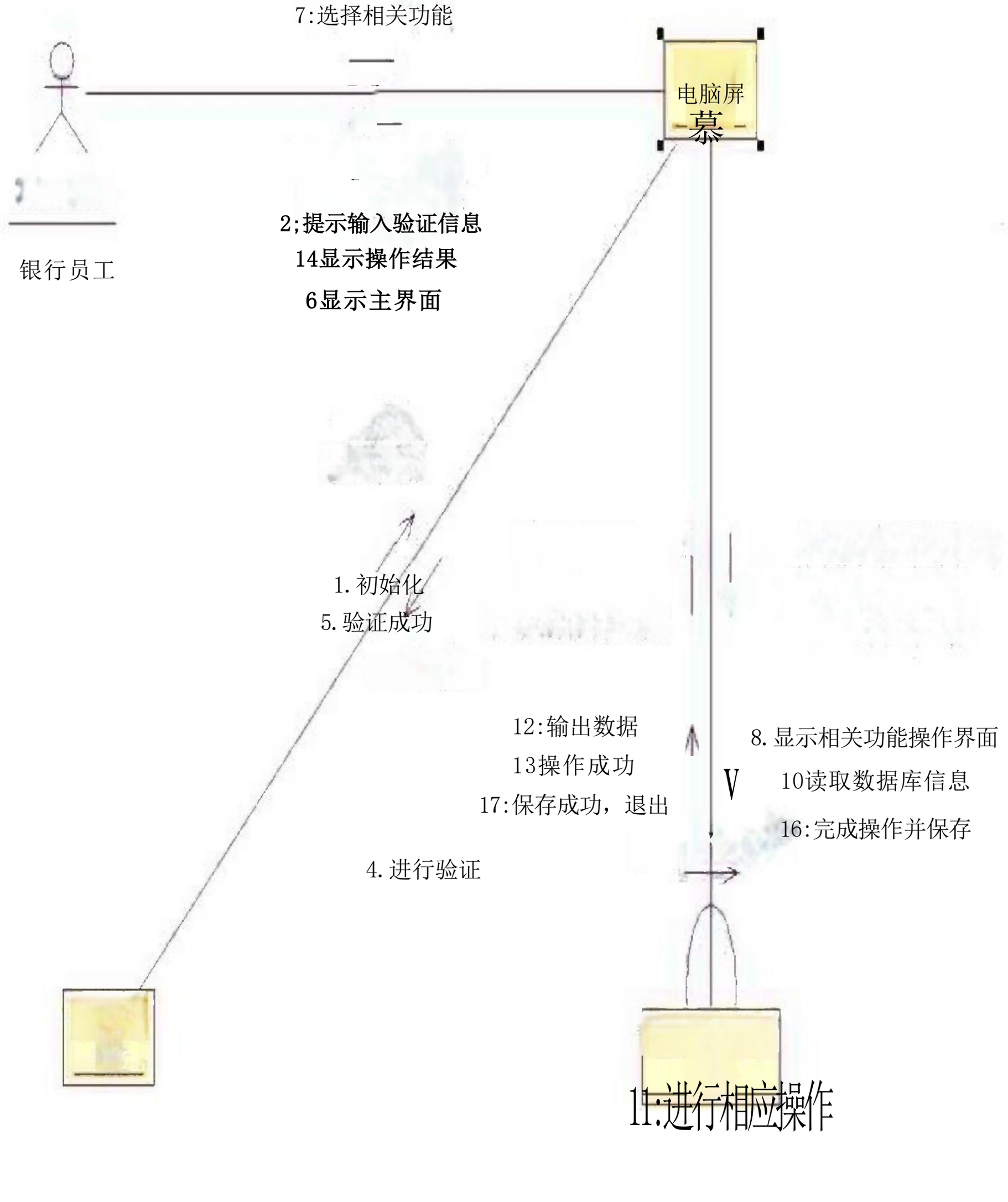


图3银行储蓄管理系统协作图


3.输入用户名、密码选择职类
9:执行具体操作
15退出操作
7:选择相关功能
电脑屏
幕
2;提示输入验证信息
14显示操作结果
6显示主界面
银行员工
1.初始化
5.验证成功
12:输出数据
13操作成功
17:保存成功，退出
8.显示相关功能操作界面
10读取数据库信息
16:完成操作并保存
4.进行验证
11:进行相应操作
身 份验
证
银行存储管理
系统

四类的设计与分析

为了使得程序有较高的易读性，我做了多个界面并为每个界面设了类。由多个类来把所有的功能函数包括起来。功能分的比较细，条理清楚明确。下面对这些类作一下简单的分析：

1 功能类(包括denglu, jiemian, jiemianx, tianjia, shanchu, xianshi, gengxin)

```
class  denglu:public  CDialog
{
public:
    denglu(CWnd*pParent  =NULL);

    enum{IDD=IDD_DIALOG_denglu    };
    CComboBox      m  status;
    CEdit    m_pwd;
    CEdit    m_num;

    protected:
    virtual void DoDataExchange(CDataExchange*pDX);

protected:
    HICON m_hIcon;
    virtual  BOOLOnInitDialog();
    afx_msg void OnSysCommand(UINT nID,LPARAM IParam);
    afx_msg void OnPaint();
    afx_msg HCURSOR OnQueryDragIcon();
    afx_msg void OnButton1();
    afx_msg void OnButton2();
    DECLARE_MESSAGE_MAP()
};
```

MFC 的功能类所包含的主要都是界面按键的成员方法

2 Resource类

```
//{{NO_DEPENDENCIES}}
//Microsoft  Developer  Studio  generated  include  file.
//Used  by  zhi.rc
//
#define  IDM  ABOUTBOX                    0x0010
#define      IDD_ABOUTBOX                100
#define      IDS_ABOUTBOX                101
#define  IDD  ZHI  DIALOG                102
#define  IDD_DIALOG_tianjia              102
#define      IDR_MAINFRAME               128
#define IDD_DIALOG_cunkuan               131
#define IDD_DIALOG_gengxin               131
#define IDD_DIALOG_qukuan                132
#define IDD_DIALOG_xianshi               133
#define IDD_DIALOG_denglu                134
#define IDD_DIALOG_jiemian               135
#define IDD DIALOG shanchu               136
```

```
#define IDD_DIALOG_jiemianx                137
#define    IDC    EDITname                 1003
#define   IDC_EDITriqi                     1005
#define IDC_EDITyu                         1007
#define       IDC_LIST1                    1026
#define    IDC_COMB01                      1031
#define      IDC      BUTTON1              1032
#define    IDC    EDITnum                  1033
#define      IDC_EDITpwd                   1034
#define          IDC_BUTTONqu              1037
#define   IDC   BUTTONadd                  1039
#define   IDC   BUTTONdel                  1040
#define   IDC   BUTTONshow                 1041
#define    IDC_BUTTONcun                   1042
#define      IDC_BUTTONupdate              1042
#define   IDC   BUTTONout                  1043
#define IDC_EDITid                         1044
#define    IDC     EDITcun                 1045
#define          IDC_BUTTONok              1046
#define    IDC_BUTTONback                  1047
#define      IDC_EDITqu                    1048
#define          IDC_BUTTON2               1049
#define    IDC_BUTTONcheck                 1050

//Next default values for new objects
//
#ifdef  APSTUDI0_INVOKED
#ifndef APSTUDI0 READONLY SYMBOLS
#define   APS    NEXT   RESOURCE    VALUE          138
#define       _APS_NEXT_COMMAND_VALUE            32771
#define   APS    NEXT   CONTROL    VALUE          1051
#define _APS_NEXT_SYMED_VALUE                     101
#endif
#endif
```

定义各个按键

3　zhi和 StdAfx 类

```
class  CZhiApp:public   CWinApp
 {
public:
      CZhiApp();


//0verrides
      //ClassWizard    generated    virtual    function    overrides
```

//{{AFX_VIRTUAL(CZhiApp)
public:
virtual B00L InitInstance();
//}}AFX_VIRTUAL

//Implementation

//{{AFX_MSG(CZhiApp)
//NOTE -the ClassWizard will add and remove member functions here.
DO NOT EDIT what you see in these blocks of generated code !
//}}AFX_MSG
DECLARE MESSAGE MAP()
};

#if !defined(AFX STDAFX H E9923E7F D8824080 BAFO 8699A6C6D790 INCLUDED_)
#define AFX STDAFX H E9923E7F D8824080 BAFO 8699A6C6D790 INCLUDED

#if _MSC_VER>1000
#pragma once
#endif//_MSC_VER>1000

#define VC EXTRALEAN //Exclude rarely-used stuff from Windows headers

#include <afxwin.h> //MFC core and standard components
#include <afxext.h> //MFC extensions
#include <afxdisp.h> //MFC Automation classes
#include <afxdtctl.h> //MFC support for Internet Explorer 4 Common Controls
#ifndef _AFX_NO_AFXCMN_SUPPORT
#include <afxcmn.h> //MFC support for Windows Common Controls
#endif //_AFX_NO_AFXCMN_SUPPORT
创建MFC文件便会自动生成的定义类

4 SQL类
#include"afxdb.h"
class SQL
{
public:
CDatabase m_db;

```
    CRecordset     rs;
};
```

数据库连接类，包含了与数据库连接相关的成员方法

五数据库中表结构设计

本系统所需连接的表有两个，包括：admin 和 users

admin 包含a_num,a_pwd,a_right ,关于操作员的员工号，密码和职类，用于存储操作员信息

users 包含u_id,u_name,u_riqi,u_yu, 关于用户的帐号，户名，开户日期和余额，用于存储用户信息

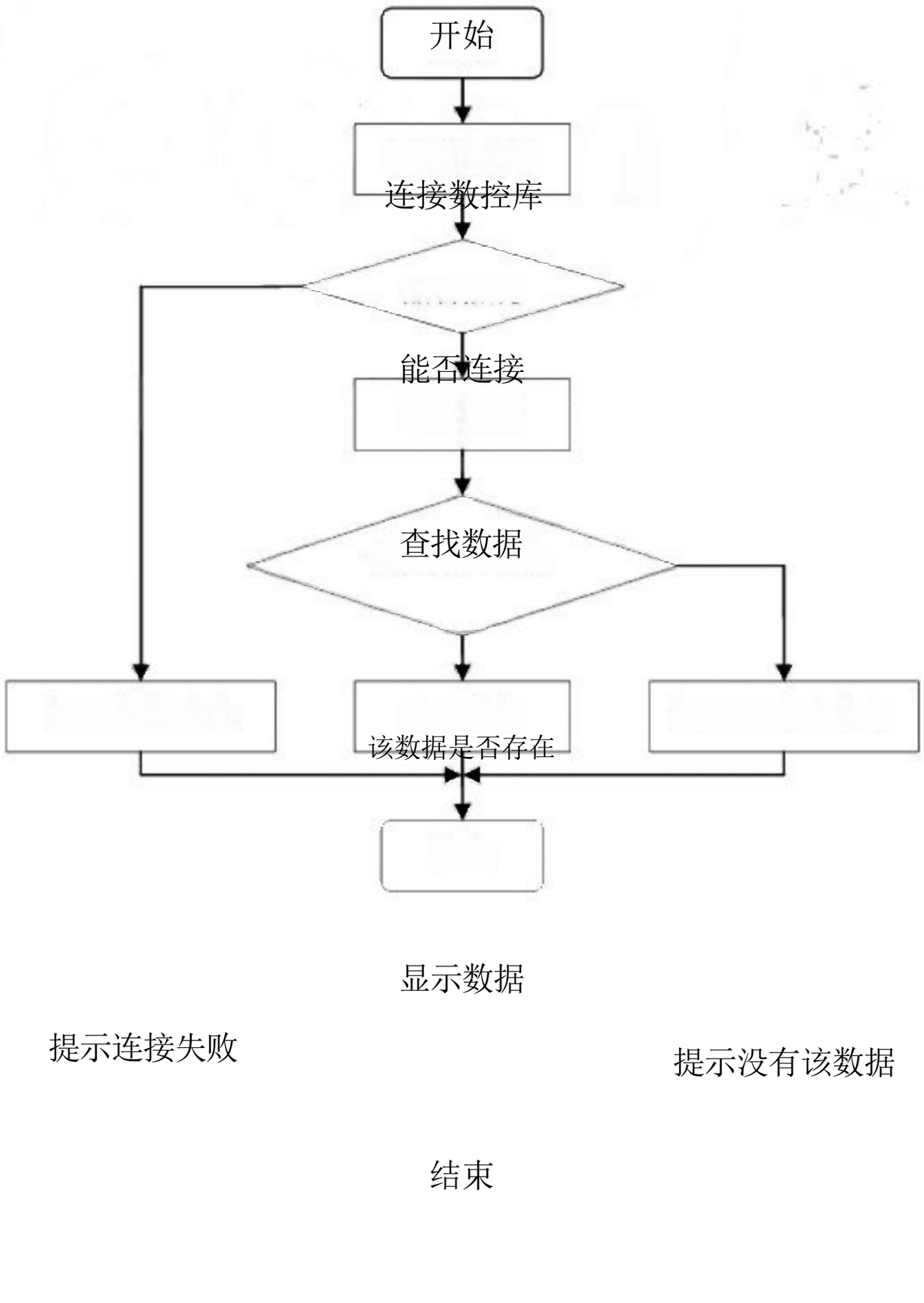
六特色算法分析

在gengxin 类中的check 算法

功能：连接数据库，查找出相应的数据

算法思路：定义几个Cstring 变量，分别存放输入搜索相关值和搜索操作，然后按步骤执行操作

流程图：



代码：

```
CString  u_id,u_yu;
CString  check;


m_id.GetWindowText(u    id);
```

```
if(sql4.m_db.IsOpen())

 sql4.m_db.Close();



sql4.m_db.OpenEx(conn,0);
if(!sql4.m_db.IsOpen())
 {
 AfxMessageBox("fail to connect to the db");
 return;



sql4.rs.m_pDatabase=&sql4.m_db;
check="select*from  users  where  u_id="+u_id+"";
sql4.rs.Open(CRecordset::snapshot,check,CRecordset::readOnly);
if(sql4.m_db.CanUpdate())
 {
 sql4.rs.GetFieldValue("u_yu",u_yu);
 m_yu.SetWindowText(u_yu);


 else
 {
 AfxMessageBox("not such data in db");
 }
```

七功能测试

图 1 登陆界面

图2操作员A 主界面

图3添加用户操作

图4删除用户操作