南 阳 理 工 学 院

本科生毕业设计(论文)

学院(系)： 软件学院

专 业： 软件工程

学 生： 高晶辉

指导教师： 惠征

完成日期 2018 年 05 月

南阳理工学院本科生毕业设计（论文）

基于iOS的手机数据网络备份软件的设计与  
实现

Design and Implementation of Mobile Data Network  
Backup Software Based on iOS

总 计：毕业设计(论文) 29页

表 格： 6个

图 片： 18个

南 阳 理 工 学 院 本 科 毕 业 设 计(论文)

基于iOS的手机数据网络备份软件的设计与  
实现

Design and Implementation of Mobile Data Network  
Backup Software Based on iOS

学 院(系)： 软件学院

专 业： 软件工程

学 生 姓 名： 高晶辉

学 号： 1415925718

指导教师(职称)： 惠征 讲师

评 阅 教 师： 段云涛

完 成 日 期： 2018年05月04日

南阳理工学院

Nanyang Institute of Technology

基于iOS的手机数据网络备份软件的设计与  
实现

软件工程 高晶辉

1. 手机更新换代的速度逐渐加快，但是iOS平台一直存在的一个问题就是数据的转移问题，iOS系统的自带通讯录以及相册不存在导入导出功能，用户不能很方便的转移数据，本课题的意义在于能够很轻松的帮助用户转移数据，项目在完成后将支持一键导出导入手机中的通讯录、视频以及相册。能极大的帮助用户转移数据以及节省手机空间。该系统为原生app，采用oc，storyboardy技术实现界面和主体框架的搭建，使用API相关技术实现主要功能的实现。整个系统各个界面简约风格，操作做到了尽量简单。
2. 手机数据备份；数据转移；iOS；Dropbox

Design and Implementation of Mobile Data Network  
Backup Software Based on iOS

Software Engineering Major Gao Jinghui

1. The speed of upgrading mobile phones is gradually accelerating, but one problem that has always existed on the iOS platform is the problem of data transfer. There is no import/export function for the iOS system's own address book and album. Users cannot transfer data easily, so the significance of this topic is significant. It is easy to help users to transfer data. After the project is completed, it will support one-touch export to import contacts, videos, and albums in the mobile phone. This software can greatly facilitate the user. The system is a native app, using oc, storyboardy technology to achieve the interface and the main framework, using API related technology to achieve the main functions. The overall interface and the operation is simple.
2. mobile phone data backup; data transfer; iOS; Dropbox

目 录

[1 绪论 1](#_Toc516507062)

[1.1 课题研究背景与意义 1](#_Toc516507063)

[1.1.1 课题背景 1](#_Toc516507064)

[1.1.2 课题研究意义 1](#_Toc516507065)

[1.2 国内外研究现状分析 1](#_Toc516507066)

[1.2.1 国外现状 1](#_Toc516507067)

[1.2.2 国内现状 2](#_Toc516507068)

[1.3 课题研究内容与目标 2](#_Toc516507069)

[1.4 本章小结 2](#_Toc516507070)

[2 相关技术简介 2](#_Toc516507071)

[2.1 iOS开发平台 2](#_Toc516507072)

[2.1.1 iOS简介 2](#_Toc516507073)

[2.1.2 iOS系统构架 3](#_Toc516507074)

[2.1.3 Dropbox 3](#_Toc516507075)

[2.1.4 Google Drive 4](#_Toc516507076)

[2.1.5 iOS API 4](#_Toc516507077)

[2.2 Xcode 4](#_Toc516507078)

[2.3 Objective-c 4](#_Toc516507079)

[2.4 本章小结 4](#_Toc516507080)

[3 需求分析 5](#_Toc516507081)

[3.1 可行性分析 5](#_Toc516507082)

[3.1.1 技术方面的可行性分析 5](#_Toc516507083)

[3.1.2 硬件条件方面的可行性 5](#_Toc516507084)

[3.1.3 市场方面的可行性 5](#_Toc516507085)

[3.1.4 社会因素方面的可行性 5](#_Toc516507086)

[3.2 需求描述 6](#_Toc516507087)

[3.2.1 软件需求描述 6](#_Toc516507088)

[3.2.2 软件功能描述 6](#_Toc516507089)

[3.3 面向对象分析 6](#_Toc516507090)

[3.3.1 系统主要功能用例分析 6](#_Toc516507091)

[3.3.2 备份用例分析 7](#_Toc516507092)

[3.3.3 还原功能用例分析 8](#_Toc516507093)

[3.3.4 登录功能用例分析 9](#_Toc516507094)

[3.3.5 注销功能用例分析 9](#_Toc516507095)

[3.4 系统功能分析 9](#_Toc516507096)

[3.4.1 系统的功能分析 10](#_Toc516507097)

[3.4.2 系统非功能性分析 10](#_Toc516507098)

[3.5 本章总结 10](#_Toc516507099)

[4 系统设计 10](#_Toc516507100)

[4.1 概要设计 10](#_Toc516507101)

[4.1.1 概述 10](#_Toc516507102)

[4.1.2 系统框架结构 10](#_Toc516507103)

[4.1.3 系统功能模块图 11](#_Toc516507104)

[4.2 详细设计 13](#_Toc516507105)

[4.2.1 系统总流程 13](#_Toc516507106)

[4.2.2 备份模块设计 14](#_Toc516507107)

[4.2.3 还原功能模块设计 15](#_Toc516507108)

[4.3 用户数据存储设计 16](#_Toc516507109)

[4.3.1 沙盒 16](#_Toc516507110)

[4.3.2 沙盒设计 16](#_Toc516507111)

[4.4 本章总结 17](#_Toc516507112)

[5 系统实现 17](#_Toc516507113)

[5.1 系统界面实现 17](#_Toc516507114)

[5.2 备份实现 19](#_Toc516507115)

[5.3 还原实现 21](#_Toc516507116)

[5.4 本章总结 24](#_Toc516507117)

[6 系统测试 24](#_Toc516507118)

[6.1 测试环境 24](#_Toc516507119)

[6.2 测试目标 25](#_Toc516507120)

[6.3 测试方法 25](#_Toc516507121)

[6.3.1 功能性测试 25](#_Toc516507122)

[6.3.2 结构性测试 26](#_Toc516507123)

[6.4 测试总结 26](#_Toc516507124)

[结束语 27](#_Toc516507125)

[参考文献 28](#_Toc516507126)

[致谢 29](#_Toc516507127)

# 绪论

目前手机的更新速度越来越快，但是iphone的手机用户在更换手机时仍然存在的问题是手机上的数据不容易轻松的转移到另外一个手机.现在新一代的iphone手机刚刚面世，必然有许多人更换新款iphone，因此本文设计的这款软件可以轻松的把手机中的数据备份到网络中并在另一个手机中恢复。

## 课题研究背景与意义

### 课题背景

近几年来，智能手机以一种惊人的速度占据着原来属于功能机的市场，到目前绝大多数人的手机已经是智能手机。智能机有着属于自己的系统可以下载安装属于自己的软件，同时由于硬件及系统的飞速发展，智能机的换代速度非常迅速。

iPhone是[美国](https://baike.baidu.com/item/%E7%BE%8E%E5%9B%BD/125486)[苹果公司](https://baike.baidu.com/item/%E8%8B%B9%E6%9E%9C%E5%85%AC%E5%8F%B8)研发的智能手机系列，搭载苹果公司研发的[iOS](https://baike.baidu.com/item/iOS/45705)操作系统 ，目前iOS手机由于有如下优势备受用户欢迎：1.iOS在使用时产生的后台是伪后台或无后台程序，优化程度也很高因此操作流畅，卡顿很少出现[1]；iOS的高度封闭使得用户的数据相对更安全；3.由于App Store的存在为用户以及开发者提供了便利;4.软件与硬件整合度高. iOS系统的软件与硬件的整合度相当高，使其分化大大降低，在这方面要远胜于碎片化严重的Android[2]。

### 课题研究意义

本课题设计的意义是进行手机软件的备份与恢复，利用Dropbox以Google提供的官方API将数据备份到网盘中，并在用户需要的时候恢复到任意iOS手机中。本系统还支持通过本地局域网连接进行照片传输。目前智能手机的换代速度越来越快，iOS手机目前也存在10种以上的版本，但是iOS手机不能像安卓那样方便的进行数据的转移，本软件可以较为方便的是手机数据转移到其他手机中同时可以帮用户把暂时不需要的数据存在网盘中以节省手机空间，因此该软件可以说是方便用户的软件。

## 国内外研究现状分析

### 国外现状

国外对数据的重视一直非常高，因此应用而生了多种网盘，常见的有Google Drive，Dropbox等，也正是因此，国外公司为开发者提供了具有多种功能的官方API以及相对应的详细问题供开发者使用，在这种背景下国外数据备份的软件是开发者逐渐增加，同时也推动了国外软件备份软件的发展，Google也为自己的Google Drive发布过专门的软件，该软件能极大的方便用户。

### 国内现状

随着人们生活水平的不断提高，国民的日常需求也变得越来越复杂，手机不仅仅是用来通讯，还有当做摄像机以及视频播放器来使用，因此手机中经常存放一些图片以及视频，但是这些对于iOS手机存在的一些问题，比如手机空间的不足以及当用户更换新手机时不能很方便的将数据转移到另一个手机中，该软件可以很好的完成这两项需求，因此在日常生活中存在重要价值。

## 课题研究内容与目标

数据备份和数据转移逐渐被人们日常所需要。本课题设计的软件可以在手机上进行一键备份，支持对通讯录 相册以及视频的备份到Dropbox或Google drive中，并且每次备份都有记录，用户可以在需要的时候在任意手机选择一次备份的信息进行恢复到手机中。本软件采用oc语言编程，通过Dropbox以及GoogleDrive的官方API进行登录以及数据的备份以及恢复，使用xcode进行代码的调试以及运行，同时为了保证良好的界面，使用storyboard进行界面的布局，使用约束来适配不同屏幕尺寸。

设计的主要目标是了解iOS手机app开发的主要流程以及在开发过程中遇到的各种问题和解决问题的能力，通过该课题，进一步了解了app的主要开发模式，掌握了oc语言并大大加强了对于API文本的阅读以及使用能力。

## 本章小结

本章介绍了数据转移相关的软件在各个地区目前的状况。通过查询多方资料对iOS以及Dropbox和Google drive有了较为详细的了解，同时制定了论文结构以及找到了课题研究的内容和目标，为系统的开发打下了基础。

# 相关技术简介

本章主要介绍系统在开发过程中所需要使用的技术，如iOS系统简介、Dropbox、Google drive、Xcode开发工具、iOS API、Objective-c语言等，在该位置对这些技术做出介绍和说明。

## iOS开发平台

### iOS简介

iOS是由苹果公司最早于07年的大会上公布的这个系统，原本打算只用用于智能手机系统，但后来又用于苹果平板和苹果电视等产品。原名为iPhone OS，在2010年改名为现在的iOS[3]。

iOS使用objectivc-c/swift， XcodeW是苹果公司开发的编程软件开发语言该平台是开发 人员建立0S X和iOS应用程序的最快捷的方式。

自从[Xcode](https://zh.wikipedia.org/wiki/Xcode) 更新到3.1之后，它就成为了iOS软件开发工具包的开发环境。和[Mac OS X](https://zh.wikipedia.org/wiki/Mac_OS_X)的应用程序一样，iOS应用程序使用Objective-C语言，一些应用程序可以写成C或C++语言[4]。苹果通常会公布两个 iOS 软件开发工具包，包括主要的iOS X.0（例如 iOS 10.0）和次要的iOS X.X（例如 iOS 10.1）。一般来说正式版发布之前都会有几个测试版发布，是为了测试应用程序的兼容性和增加新的功能[5]。

截止到17年[App Store](https://zh.wikipedia.org/wiki/App_Store_(iOS))已含有数百万个应用。iOS系统目前为全球第二大移动操作系统市占约一成左右 ，但离占据最大的安卓系统还是有很大差距。[苹果公司](https://zh.wikipedia.org/wiki/%E8%8B%B9%E6%9E%9C%E5%85%AC%E5%8F%B8)于2017年6月5日之2017年[苹果全球开发者大会](https://zh.wikipedia.org/wiki/%E8%8B%B9%E6%9E%9C%E5%85%A8%E7%90%83%E5%BC%80%E5%8F%91%E8%80%85%E5%A4%A7%E4%BC%9A)中介绍了[iOS 11](https://zh.wikipedia.org/wiki/IOS_11)，正式版已于2017年9月20日开放下载。

### iOS系统构架

硬件框架：iOS 使用基于与苹果[电脑](https://zh.wikipedia.org/wiki/Mac%E7%B3%BB%E5%88%97)不同的[中央处理器](https://zh.wikipedia.org/wiki/%E4%B8%AD%E5%A4%AE%E5%A4%84%E7%90%86%E5%99%A8)，使用由[PowerVR](https://zh.wikipedia.org/wiki/PowerVR)视频卡渲染的[Metal](https://zh.wikipedia.org/wiki/Metal_(API))以[Darwin](https://zh.wikipedia.org/wiki/Apple_Darwin)作为基础。

系统架构层次：iOS 分为核心操作系统层（the Core OS layer）、[核心](https://zh.wikipedia.org/wiki/%E6%A0%B8%E5%BF%83)服务层（the Core Services layer）、[媒体](https://zh.wikipedia.org/wiki/%E5%AA%92%E4%BD%93)层（the Media layer）及[触控应用](https://zh.wikipedia.org/wiki/%E8%A7%A6%E6%8E%A7%E5%BC%8F%E8%90%A4%E5%B9%95)层（the Cocoa Touch layer）[6]。系统架构层次如图2-1所示。

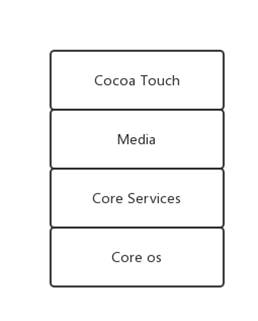


图2-1 iOS系统架构图

### Dropbox

Dropbox是Dropbox公司提供的一项在线存储服务，原来是利用云计算来完成网络上的文件同步，每个注册用户都可以存储和分享文件.Dropbox有收费和免费两种收费模式，而且客户端支持多种系统，同时有网页端。Dropbox的译名意义为百宝箱，寓意着里面可以存储很多宝贝。

### Google Drive

Google云端硬盘是[Google](https://zh.wikipedia.org/wiki/Google)的一个[在线网络云盘功能](https://zh.wikipedia.org/wiki/%E7%BD%91%E7%BB%9C%E7%A1%AC%E7%9B%98)，同时结合了谷歌的其他相应功能，于12年起逐渐开放给普通用户下载使用。但这项服务早在06年就被人们有所提及。

最早出现的是Google Storage，但不太受关注。Google Drive可视为Google Docs的升级版，把空间从1GB升级到5GB，后集成[Gmail](https://zh.wikipedia.org/wiki/Gmail)与[Google+](https://zh.wikipedia.org/wiki/Google%2B)相册的存储空间，总共15GB。Google Drive可以免费为提供用户15GB的空间，但单个文件的大小限不能超过10G[7]。有需要的用户可以付费扩充存储空间，最大可以达到30TB。谷歌云盘的目录管理类似于[Dropbox](https://zh.wikipedia.org/wiki/Dropbox)的，该目录管理在谷歌云盘中叫做Folder location，同样可同步上传文件到网络服务器中。

### iOS API

在iOS系统中很多个人或集体为开发者提供了多种API（Application Programming Interface,[应用程序](http://baike.baidu.com/item/%E5%BA%94%E7%94%A8%E7%A8%8B%E5%BA%8F)编程接口）是一些预先定义的[函数](http://baike.baidu.com/item/%E5%87%BD%E6%95%B0)，目的是提供[应用程序](http://baike.baidu.com/item/%E5%BA%94%E7%94%A8%E7%A8%8B%E5%BA%8F)与开发人员基于某[软件](http://baike.baidu.com/item/%E8%BD%AF%E4%BB%B6)或硬件得以访问一组[例程](http://baike.baidu.com/item/%E4%BE%8B%E7%A8%8B)的能力，而又无需访问源码，或理解内部工作[机制](http://baike.baidu.com/item/%E6%9C%BA%E5%88%B6)的细节[8],本课题中使用到的Dropbox以及Google Drive均有官方iOS API。

## Xcode

Xcode是目前iOS开发者最多人使用而且最完善的开发工具，由苹果公司官方提供支持。

Xcode继承自[NeXT](https://zh.wikipedia.org/wiki/NeXT)的[项目构架器](https://zh.wikipedia.org/w/index.php?title=Project_Builder&action=edit&redlink=1)。The Xcode suite包含有[GNU Compiler Collection](https://zh.wikipedia.org/wiki/GNU_Compiler_Collection)自由软件（GCC、apple-darwin9-gcc-4.0.1以及apple-darwin9-gcc-4.2.1，默认的是后者），并支持[C语言](https://zh.wikipedia.org/wiki/C%E8%AA%9E%E8%A8%80)、[C++](https://zh.wikipedia.org/wiki/C%2B%2B)、[Fortran](https://zh.wikipedia.org/wiki/Fortran)、[Objective-C](https://zh.wikipedia.org/wiki/Objective-C)、[Objective-C++](https://zh.wikipedia.org/wiki/Objective-C%2B%2B)、[Java](https://zh.wikipedia.org/wiki/Java)、[AppleScript](https://zh.wikipedia.org/wiki/AppleScript)、[Python](https://zh.wikipedia.org/wiki/Python)、[Ruby](https://zh.wikipedia.org/wiki/Ruby)和[Swift](https://zh.wikipedia.org/wiki/Swift_(%E7%A8%8B%E5%BC%8F%E8%AA%9E%E8%A8%80))，还提供[Cocoa](https://zh.wikipedia.org/wiki/Cocoa)、[Carbon](https://zh.wikipedia.org/wiki/Carbon_(API))以及Java等编程模式[9]。

## Objective-c

Objective-c,也就是我们俗说的oc[10]。它是面向对象的语言，主要使用的系统有两个。目前由于Swift语言存在更新速度过快、某些方面不完善等问题，虽然这些问题正在改善，但Objective-c仍然是iOS开发使用最多的语言，使用率远远大于目前的Swift语言。

## 本章小结

本章是对课题需要用到的技术的研究，首先是介绍了iOS系统，接着围绕着课题的主要功能介绍了所需要用的一些技术的环境以及配置，最后针对课题使用的开发语言以及工具进行了阐述，为以后课题的研究打下基础，是软件设计中重要且必须进行的一项工作。

# 需求分析

这款软件需要服务于不同地区的各种用户群体，因此需要进行用户需求分析。总的来说就是需要了解不同地区用户的各种需求，用户需要此软件能实心什么功能，解决哪些问题，以及怎样提高用户体验。

## 可行性分析

### 技术方面的可行性分析

对《软件工程》课程有一定的学习以及了解，基本要求是熟悉软件开发过程中的每个流程以及标准；

对《c语言程序设计》有整体的学习，要全面学习，需要熟悉基本语句和基本程序的编写过程；

熟悉iOS API的基本使用方法，能看懂API文档并根据文档以及该课题需改使用；

学习过《iOS开发从零基础到appstroe上架》，掌握iOS手机应用开发过程的核心技术；

了解xcode工具的基本使用方法，如创建项目，生成文件等；

目前市面上的手机备份软件已经趋于成熟，可以参考借鉴。

### 硬件条件方面的可行性

需要的硬件要求如下：

（1）一台系统为mac的台式电脑，系统需要为[macOS High Sierra](https://baike.baidu.com/item/macOS%20High%20Sierra) 10.11以上；cpu主频在2.0GHz以上；内存大于4g；硬盘空间尽量充足。且需要安装最新版Xcode，word2007或以上版本，以及vpn软件。

（2）一部iphone5s或更新型号的手机，系统为 iOS 11或以上版本。

由于个人及公司拥有达到该要求的硬件，因此硬件方面不存在问题。

### 市场方面的可行性

数据价值近年来越发被人们所重视，且数据备份功能可以面向绝大多数人服务，因此此类软件在市场上用户广大的用户群，而且目前市面上手机数据备份软件种类虽然较多，但本软件是面向iOS手机用户且使用国外常用服务器作为存储依据，因此避免了大多数竞争力。综此可以认为该软件在市场将有良好的前景，在市场方面的可行性不存在问题。

### 社会因素方面的可行性

（1）本软件界面简洁清晰，为了提高用户体验，功能使用尽量简单，多数为一键式操作，大多数会使用手机的人使用本软件不会出现问题。

（2）虽然目前市面上备份软件较多，可能会出现相似软件，但最多是借鉴，绝不存在抄袭，因此不会出现版权和专利相关问题。

## 需求描述

### 软件需求描述

系统根据用户各种需求进行开发设计。系统可以一键把手机中的通讯录，视频或者照片备份，用户可以选择选择任意一种或多种自己想要备份的文件到自己的Google Drive以及Dropbox中，并在需要的时候在任意一台iOS系统中恢复。同时数据在备份后必须能够显示到相应的网盘中，用户可以在网盘中看到相应的文件，以方便用户查看文件和在Web端进行文件的下载、删除或者其他相应的操作。用户操作后将同样影响该系统。

且为了用户方便，用户不需要额外的账号，仅需要相应的网盘账号，且需要对用户账号进行安全存储。

### 软件功能描述

该系统最核心的功能是能转移用户手机中的数据，同时为了满足不同用户的需求，需要能够备份到Google Drive以及Dropbox，备份的位置以及备份数据的类型可以由用户自己选择，另外需要还原数据功能，还原对于细节的要求类似于备份功能。同时需要简单操作的界面设计以及不同的语言来满足不同地区的用户,计划上架的地区为美国和日本地区。

## 面向对象分析

首先进行用例模型的创建。用例模型在整个分析阶段起着重要的重用，用例模型最重要的作用是将系统行为传达给客户或最终用户。该系统中的用例模型主要以用户为中心，围绕着用户进行的功能进行创建，所建立的各个功能都是用户所需要或者能看见的功能进行设计。

### 系统主要功能用例分析

创建系统的用例模型是该系统进行设计的第一项工作。后续的开发都围绕着该用例进行设计。

本系统主要服务于用户，用户又是个庞大的群体，里面有各种各样的人，如成年人，老人学生，为了让不同的用户都能顺利方便的使用软件，软件设计要求界面尽量简介清晰，操作在简化的同时要保证基本功能的顺利完成，所以设计方面界面采用简约风格设计。

本系统只面向用户服务，也就是说本系统只有用户一类角色，因此围绕着用户进行用例模型的创建。用户的主要功能为备份和还原，但这两项功能中用户又有多种选择，例如备份到哪里，从哪里还原，备份哪些文件，还原哪些文件等等，因此根据用户及功能权限创建用系统主要功能用例图如图3-1所示。同时还有一些非主要功能如注销，备份等一些不在系统总用例图的功能。

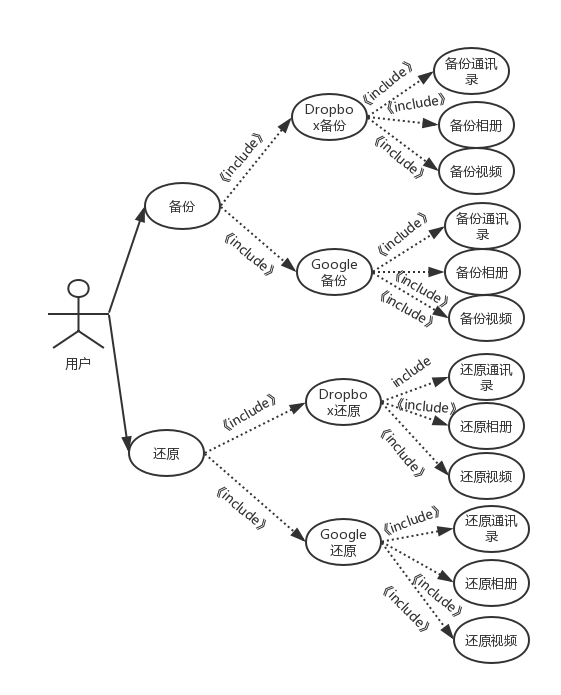


图3-1系统主要功能用例图

### 备份用例分析

主要功能：转移用户数据到Google Drive或Dropbox。

具体描述：可以选择通讯录，视频或者相册的一项或者多项，备份前需要提前登陆自己的账号，备份结束后会提示备份成功的文件数量。

点击事件流：用户点击备份到Google Drive或Dropbox，会提示登陆相应账号，输入正确后会跳转到备份界面，在备份界面当用户选择完毕需要备份的数据类型之后点击开始“按钮”开始备份，备份结束后会还原备份界面。

备份是该系统的最主要的功能，功能要做到尽量全面，而且操作必须简单易懂以方便用户。

备份用例具体如图3-2所示。

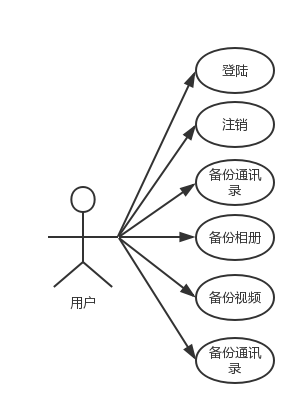


图 3-2备份功能用例图

### 还原功能用例分析

主要功能：还原曾经备份过的数据到手机中。

具体描述：将原来备份过的数据恢复到用户系统中。

点击事件流：用户点击从Google Drive或Dropbox还原，会提示登陆相应账号，输入正确后会跳转到恢复界面，在恢复界面可以跳转备份记录选择界面，当用户选择完毕需要还原的数据类型之后点击开始“按钮”开始还原，还原结束后会还原恢复界面，以便用户进行下一操作。

还原功能与备份功能的最大区别在于还原时需要多进行一次备份记录的选择，需要跳转到另一界面，跳转到备份记录选择界面进行备份记录的选择功能，选择时将用户所有的备份记录提供给用户进行选择，用户必须选择一次未被删除的记录以进行下一步骤，如果用户不选择记录将无法回到备份界面，同时如果用户没有备份记录将无法调整备份记录选择界面。

还原用例图如图3-3所示。

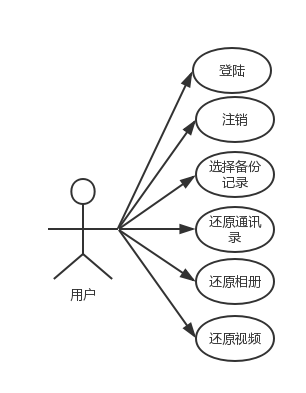


图3-3还原用例图

### 登录功能用例分析

主要功能：登录相应的账号。

具体描述：登录Dropbox或Google账号。

点击事件流：用户未登录账号且点击进行备份或还原时，会弹出相应的登录界面，在，在登录界面输入正确的账号密码后账号会被存储以便下次使用，登录完成后会跳转下一界面。

### 注销功能用例分析

主要功能：注销相应的账号。

具体描述：注销Dropbox或Google账号。

点击事件流：用户登录过账号后且需要注销或更换账号时点击注销按钮，会提示用户当前的账号并询问是否需要注销账号，点击“是”会注销账号并直接跳转登录界面以便用户进行账号更换，如果用户不要更换账号点击返回按钮即可变成未登录状态，处于未登录状态将无法进行还原和备份功能。

## 系统功能分析

系统主要功能有4个，分别为备份到Google Drive，备份到Dropbox，从Dropbox还原，从Google Drive还原，同时还有一些次要功能，如注销账户，存储数据到手机等。

### 系统的功能分析

备份到Google Drive:登录用户的谷歌账号后，可以选择备份不同的文件到自己的Goole Drive中，文件类型选项由开关控制，用户可以一键备份文件，文件在网盘中的命名方式为备份时间。

备份到Dropbox: 登录Dropbox账号后，跳转到Dropbox备份界面，默认为仅备份通讯录，用户可以使用开关控件自由选择，选择完毕后点击“开始“按钮即开始备份，备份完成后所选中的文件将在网盘中。

从Google Drive还原：系统根据登录中的Google账号找到该账号所有的备份记录并显示给用户进行选择，选择备份记录后选择还原文件类型，全部选择完毕后点击“开始”开始还原，还原完毕后数据将直接存放在手机中。

从Dropbox还原: 系统根据登录中的Dropbox账号找到该账号所有的备份记录并显示给用户进行选择，选择备份记录后选择还原文件类型，全部选择完毕后点击“开始”开始还原，还原完毕后数据将直接存放在手机中。

保存数据到手机：用户每次登陆时都会将用户的账号信息存储在沙盒中，以便用户下次使用时不需要重复登陆，同时为了安全方面，一次只存储一个账号信息。

### 系统非功能性分析

目前iphone以及ipad的尺寸有10种以上，为了适配不同的界面，需要使用storyboard以及layout功能，为不同的界面建立不同的layout。

## 本章总结

本章对课题进行了详细的需求分析，对可行性分析，需求描述，面对对象分析以及系统功能分析做了分析描述，同时给出了系统用例图，为课题的后续开发提供了便利。

# 系统设计

通过前几章已经确实了解需求分析以及课题需要使用的各项技术后开始进行对系统的详细设计，系统设计师开发的核心包括三种设计方式。

## 概要设计

### 概述

该系统是一个实现数据转移功能的手机app，核心机制是让用户转移自己的手机数据到网上云盘或者从网上云盘转移到手机中。

### 系统框架结构

目前所有主流的iOS设备均可以上网，利用该功能访问Internte并连接到Dropbox以及Google Drive的Web service备份或从中恢复数据，并且恢复的数据将保存在手机中。如图4-1所示。

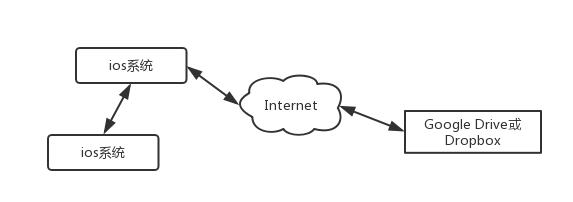


图4-1系统框架图

### 系统功能模块图

系统启动后进入主界面，主界面有两个模块，分别为备份模块和还原模块。在备份时可以选择使用Dropbox备份或者Google Drive备份，两种方式均会进入备份界面，备份界面可以选择是备份通讯录，视频还是相册，备份的数据来源于iOS系统。还原时，同样可以选择是从Dropbox或者Google以及还原通讯录，视频还是相册，不同的是需要额外进行一步备份记录的选择。还原结束后下载的相册和通讯录数据将存储到用户的手机中。如图4-2所示。

备份功能设计如下：

（1）Dropbox备份：备份手机文件到Dropbox中。

①备份通讯录，备份手机中的通讯录数据到 Dropbox云盘中，默认为开启状态，用户可以自由关闭。

②备份相册，备份手机中的相册数据到 Dropbox云盘中，默认为关闭状态，用户可以自由开启。

③备份视频，备份手机中的视频数据到 Dropbox云盘中，默认为关闭状态，用户可以自由开启。

（2）Google Drive备份：备份手机文件到谷歌云盘中。

①备份通讯录，备份手机中的通讯录数据到 Google Drive云盘中，默认为开启状态，用户可以自由关闭。

②备份相册，备份手机中的相册数据到 Google Drive云盘中，默认为关闭状态，用户可以自由开启。

③备份视频，备份手机中的视频数据到Google Drive云盘中，默认为关闭状态，用户可以自由开启。

还原功能设计如下：

（1）Dropbox还原：还原Dropbox数据到手机中。

①选择备份记录，选择用户备份的一次记录进行还原，默认为空，用户需要进行选择才能进行下一步。

②还原通讯录，还原Dropbox云盘通讯录数据到手机中，默认为开启状态，用户可以自由关闭.

③还原相册，还原Dropbox云盘相册数据到手机中，默认为开启状态，用户可以自由开启。

④还原视频，还原Dropbox云盘视频数据到手机中，默认为开启状态，用户可以自由关闭。

（2）Google Drove还原：还原Google Drive数据到手机中。

①选择备份记录，选择用户备份的一次记录进行还原，默认为空，用户需要进行选择才能进行下一步。

②还原通讯录，还原Google Drive云盘通讯录数据到手机中，默认为开启状态，用户可以自由关闭。

③还原相册，还原Google Drive云盘相册数据到手机中，默认为开启状态，用户可以自由开启。

④还原视频，还原Google Drive云盘视频数据到手机中，默认为开启状态，用户可以自由关闭。

用户中心设计如下：

（1）登录：用户选择某一种账号进行登陆。

①登录Dropbox账号，用户第一次打开应用或者重启手机后需要进行登录，使用跳转web的方式进行登录，登录成功跳转下一界面，失败返回原界面。如果没有账号可以选择使用谷歌账号登录。

②登录Google账号，用户第一次打开应用或者重启手机后需要进行登录，使用跳转web的方式进行登录，登录成功跳转下一界面，失败返回原界面。Goole仅可以使用谷歌账号登录。

（1）注销

①注销Dropbox账号，注销当前登录的Dropbox账号，注销时不会提示用户当前账号并询问是否注销。注销完成后直接跳转登录界面以便用户更换账号，无论用户是否更换账号，都可以点击左上角返回上一界面，即返回到功能选择界面以便用户进行下一步骤。

②注销Google账号，注销当前登录的Google账号，注销时会提示用户当前账号并询问是否注销，且不会跳转登录界面而是会返回上一界面，如果用户当前为登录账号，会询问是否登录账号。

系统功能如图4-2所示。

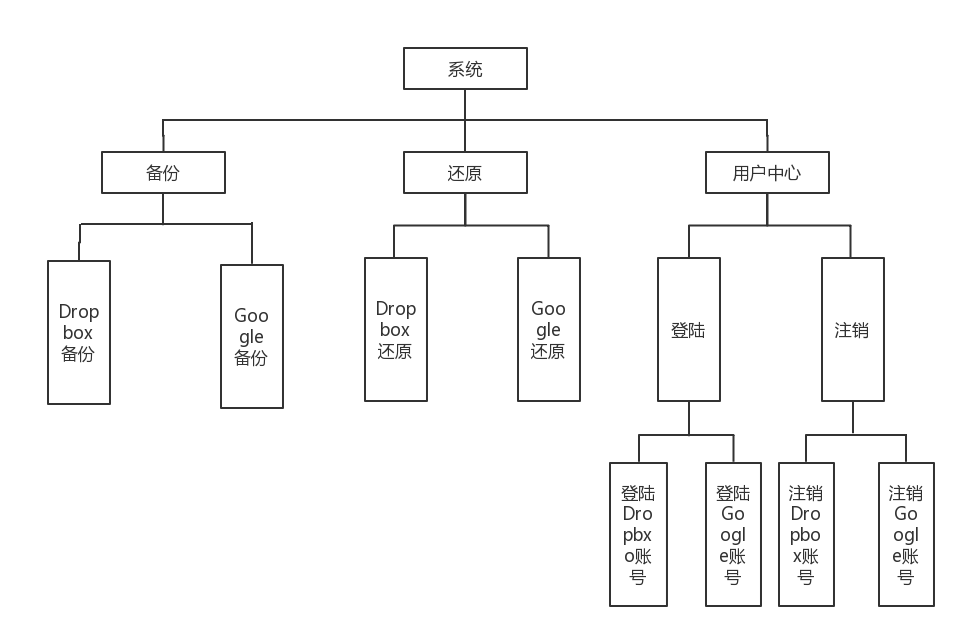


图4-2 系统功能结构图

## 详细设计

### 系统总流程

系统总流程决定了系统执行的步骤以及范围，表达了信息在各个模块之间流动的情况，是详细设计中重要的步骤，决定了系统所需要的基本功能和。

用户打开软件，首先看到的是启动界面，然后进入主界面，主界面有每个功能的按钮。用户根据自己的需要选择点击。如果需要进行备份，会让用户选择是备份到Dropbox还是Google Drive,然后选择需要备份的数据，最后开始备份。如果选择还原功能，与备份功能类似，不同的是用户需要额外选择一次备份记录进行还原。备份和原因都需要用户选择某种方式，例如只备份通讯录或者全部备份等。在进行备份记录选择时还需要跳转另一选择界面，如果选择为空会提示用户需要进行选择，并询问是确认否跳转选择界面。

所有备份或还原时都会检查用户是否授予系统通讯录和相册权限，如果某种所需的权限用户没有授予，会暂停功能并询问用户是否同意授权，用户同意会重启应用以便开启授权，用户不同意将无法进行下一步骤，同时如果用户还是想要进行下一步骤，会询问用户是否跳转设置界面进行授权，用户同意将跳转设置界面，用户不同意将无法跳转相应的功能界面，因为功能必须有相应权限才能进行，将重复此流程直到软件用所需要的权限，同时由于中国的政策原因，中国地区需要额外的网络权限，将在用户第一次打开软件时进行询问授权，总流程如图4-3所示。

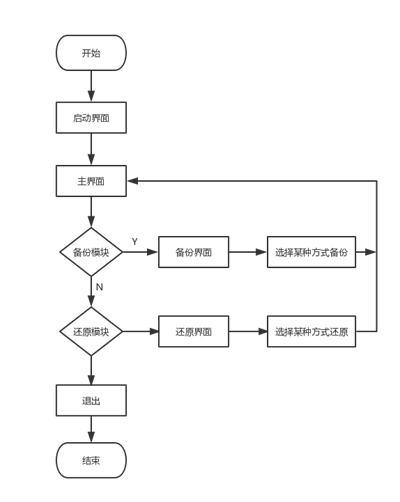


图4-3系统总流程图

### 备份模块设计

该模块实现对手机数据的备份，有备份通讯录，备份视频，备份相册三种备份方式，首先通过GoogleBackupButton,DropboxBackupButton判断用于通过什么方式备份，然后通过(IBAction)backup:(id)sender 这个方法添加按钮点击事件，如果用户点击按钮将触发这个函数，跳转登录查询方法API，如果已经登录将会跳转到备份相关界面，(IBAction)contactCharge:(id)sender,(IBAction)ablumCharge:(id)sender ,(IBAction)videoCharge:(id)sender三个点击事件可以实现都备份文件的选择。然后通过findAll方法找到从手机中找到所有用户需要备份的数据，最后通过-(void)backContact, -(void)backAbulm, -(void)backVideo三个方法实现对数据的备份。完成备份完成后数据将保存到相应网盘中并回到备份界面。备份流程图如图4-4所示。

具体实现的思路是准备好两张不同的图片以对应是否用户关闭和开启该权限，用户确认将跳转系统设置界面进行授权，如果用户同意将重启应用以便打开相应的通讯录或者相册权限。

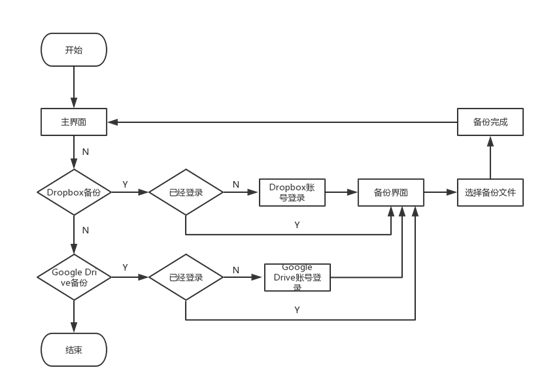


图 4-4 备份流程图

### 还原功能模块设计

还原功能模块用来还原用户曾经备份过的数据，可以从Dropbox或者Google Drive恢复。当用户选择恢复时，首先会根GoogleTranser,DropboxTranser,判断通过什么方式恢复，然后判断用户是否已经登录过相应账号，如登录过将直接跳转还原界面，然后通过选择界面完成行备份记录的选择，但是用户可以先进行备份文件的选择(IBAction)contactCharge:(id)sender,(IBAction)ablumCharge:(id)sender ,(IBAction)videoCharge:(id)sender，这三个方法将完成备份文件的选择，全部选择完毕后通过-(IBAction)downstart:(id)sender按钮点击事件开启还原功能的开始，开始后首先通过intall完成数据的初始化，然后检查系统相册以及通讯录问题，如果用户只备份通讯录或者相册，将只检查用到的权限。接着通过-(void)star方法完成网盘数据到手机中的转移，数据转移完成后通过saveContact, saveVideo, saveAbulm三种方法实现数据存储到手机中，存储到手机以后用户可以直接在手机系统的通讯录以及相册中找到文件。

另外还原功能中使用了监听来完场对用户进度进行提示，使用progess控件监听还原进度已提示用户还原进度，但progess的值为1时将结束还原，并通过alert提示用户还原结束，用户点击确认后将对还原界面进行还原，以方便用户进行多次还原，还原流程图如图4-5所示。

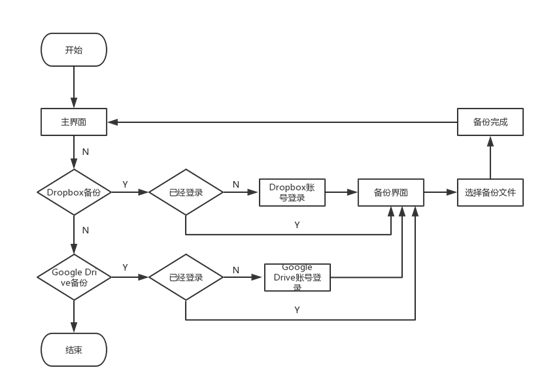


图 4-5 还原流程图

## 用户数据存储设计

本系统中使用到的主要数据为用户的相册，通讯录，视频，这些数据在iOS系统中已经有着自己的存储方式，不需要为为他们建立数据库。在本系统中需存储的数据为用于的账号信息，在这里使用沙盒的方式创建文件存储。

### 沙盒

沙盒相当于一个仅主人可见权限的文件，每个应用程序在安装是iOS都会为它生成主目录和一些关键的子目录，这就是沙盒文件[11]。沙盒的主要特点在于1.一个程序只能访问自己的沙盒，对去其他程序的不能访问;2.应用之间的数据是独立的且不能共享，每个应用的沙盒文件只能存放在自己的沙盒中，无法被拷贝到其他沙盒;3.沙盒文件不会被轻易清空，用户在下次打开软件时能读取到上次存储的沙盒数据。由上面的几点特点可以看出沙盒文件的特点在于封闭性和稳定性，因此在存储用户账号信息这个需要安全性和稳定性的数据是较为合适。

### 沙盒设计

本系统中沙盒存储的数据有，账号种类，账号名，账号key值，如表4-1所示。

表4-1账号数据表

| 字段名 | 数据类型 | 长度 | 含义 |
| --- | --- | --- | --- |
| AccountSelect | NSString | 4 | 账号类型 |
| Account | NSString | 50 | 账号 |
| AccountKey | NSString | 50 | 账号对应的key值 |

## 本章总结

本章介绍了系统的框架以及系统框架图，以及各个流程的设计过程，并站在用户的角度介绍了模块的流程并给出了各个模块的流程图，包括总流程，备份流程以及还原流程。

# 系统实现

## 系统界面实现

系统界面在手机系统中各位重要，是用户的直接感受，用户会通过界面来使用与判断一款软件，因此界面的易懂易用非常重要，同时为了满足不同用户的需求，需要适配不同尺寸。本系统使用大多数界面Storyboard配合代码完成界面的建立以及适配，某些简单页面使用纯Storyborad界面完成，但所有界面的适配均使用Storyboard中的约束完成。

启动界面：程序运行时首先进入启动界面，完毕之后系统自动进入主界面，启动界面如图5-1。



图 5-1 启动界面

启动界面完成之后会进入主界面，主界面为Storyboard辅助代码完成布局，使用Storyboard完成基本布局的建立以及约束，然后在代码中添加点击事件和一些细节的实现，例如按钮的边框圆角等。同时对于有些简单的按钮点击事件，可以直接通过连线方式进行跳转，本系统中的大部分跳转方式使用直接跳转，少数需要关联的跳转使用代码跳转。界面如图5-2所示。

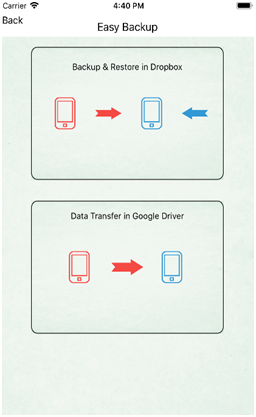


图 5-2 主界面图

主界面的核心代码：

@interface ViewController : UIViewController

{

CAShapeLayer \*\_shapeLayer;

// BorderType \_borderType;

CGFloat \_cornerRadius;

CGFloat \_borderWidth;

NSUInteger \_dashPattern;

NSUInteger \_spacePattern;

通过图5-2可以看出主界面有两个按钮，如果想使用某个功能需要点击按钮，每个按钮通过Storyborad的segue跳转连接到不同的界面。对于直接跳转不需要传值的页面使用Storyborad中的连线方式进行直接跳转，某些需要传值的页面需要根据segue的id在通过prepareForSegue方法进行一些设置一遍达到期望要求，主要采用普通的segue方式进行跳转，对于某些有NavigationController的界面，为了更方便的返回上一界面，使用push方法进行跳转，使用此方法的界面将会有系统提供的返回按钮。界面跳转图如图5-3所示。

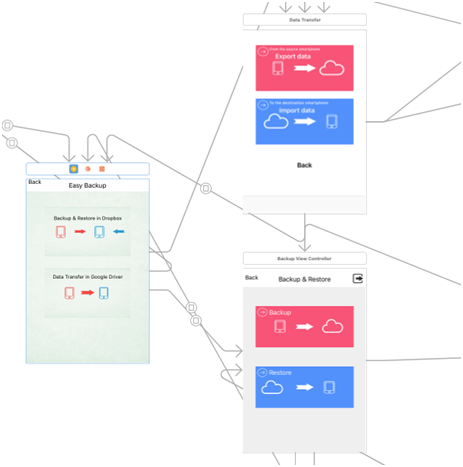


图 5-3界面跳转图

## 备份实现

在这一模块中，Google备份与Dropbox使用基本相同界面，如图5-4、图5-5所示，这一功能是首先找到所有需要备份的文件转换为NSDate或UIimage格式然后通过相应API连接到Web service，通过上传数据的方式完成备份。

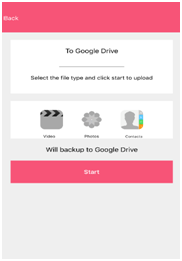


图 5-4 Dropbox备份界面

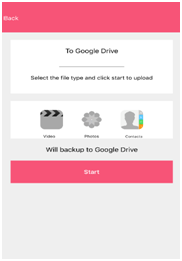


图 5-5 Google Drive备份界面

当用户选择好需要备份的文件后，会通过getContacts，getVideo以及getPictures三个函数来获取手机中的通讯录，视频以及相册，默认为获取全部文件。

通讯录的获取方式采用苹果公司提供的官方文档中的一种方法，首先需要检测用户是否拥有通讯录权限，拥有时将访问用户手机中的通讯录，以手机号为关键字获取所有联系人存储到内存当中，之后从内存中读取，使用NSDate格式存储，存储完毕后上传。

获取通讯录核心代码：

NSMutableArray \*contactsArray=[[NSMutableArray alloc] init];

CNContactStore \*store = [[CNContactStore alloc] init];

[store requestAccessForEntityType:CNEntityTypeContacts completionHandler:^(BOOL granted, NSError \* \_Nullable error) {

if (!granted) {

dispatch\_async(dispatch\_get\_main\_queue(), ^{

});

return;

}

获取通讯录完成后将进行相册的备份，相册的备份过程使用线程以及block来完成，使用线程的目的是同时可以上传多个文件，最多可同时上传5个文件，当一个文件上传完毕后，队列中的将会有一个新文件开始上传。照片的读取同样需要相册权限，读取完毕后数据的保存格式为UIimage格式。如果没有权限会提示“没有相册权限，无法备份相册。

获取相册核心代码：

NSMutableArray \*arr = [NSMutableArray array];

// 所有智能相册

PHFetchResult \*smartAlbums = [PHAssetCollection fetchAssetCollectionsWithType:PHAssetCollectionTypeSmartAlbum subtype:PHAssetCollectionSubtypeAlbumRegular options:nil];

for (NSInteger i = 0; i < smartAlbums.count; i++) {

PHCollection \*collection = smartAlbums[i];

//遍历获取相册

if ([collection isKindOfClass:[PHAssetCollection class]]) {

PHAssetCollection \*assetCollection = (PHAssetCollection \*)collection;

PHFetchResult \*fetchResult = [PHAsset fetchAssetsInAssetCollection:assetCollection options:nil];

PHAsset \*asset = nil;

if (fetchResult.count != 0) {

for (NSInteger j = 0; j < fetchResult.count; j++) {

//从相册中取出照片

asset = fetchResult[j];

PHImageRequestOptions \*opt = [[PHImageRequestOptions alloc]init];

PHCollection \*collection = smartAlbums[i];

因为视频一般较大，所以放在最后读取和上传，与相册和视频不同的是由于视频占用字节过大，全部存入内存中会导致内存溢出，因此需要单个进行读取上传，采用读取一个上传一个的方式，但是如果第二个读取完毕的时候第一个并没有上传完毕的话第二个也将开始上传以减少用户等待时间。由于在iOS中视频也保存在相册中，所以跟相册的备份一样，软件需要相册权限。

获取视频核心代码：

[self.driveService executeQuery:query completionHandler:^(GTLRServiceTicket \*ticket,

GTLRDrive\_File \*file,

NSError \*error) {

if (error == nil) {

NSLog(@"File ID %@", file.identifier);

//self.nowfile++; if([self.charge2isEqualToString:@"off"]&&[self.charge1 isEqualToString:@"off"]){

[self dismiss];

}

//self.progress.progress=self.nowfile/self.allfile;

PHImageManager \*imageManager = [[PHImageManager alloc] init];

{

if (result) {

[arr addObject:result];

}

## 还原实现

在这一模块与备份模块类似，两个功能的界面基本相同，不同的地方在于多了一步备份记录选择的，还原功能的实现核心思路是首先通过用户登录的账号，通过官方API，遍历用户所用的文件，通过特殊标识符找到用户使用该系统备份过的文件，并将文件名存储在数组中，然后将数组传到选择界面，将数组中的数据通过tableview的方式展示给用户，用户可以点击任何一次记录进行选择，选择完毕后跳转备份界面，如果不选择一次记录将不能进行备份，并提示用户是否跳转选择界面进行备份记录的选择。还原时用户必须选择至少一种文件，如没有选择文件，同样将提示用户进行选择。选择完毕后通过备份记录以及文件类型找到对应的文件夹并将其中的文件id保存下来，通过API提供的方法进行逐一下载，所有成功的文件将直接存储到用户手机中。还原界面图如图5-6、图5-7所示。

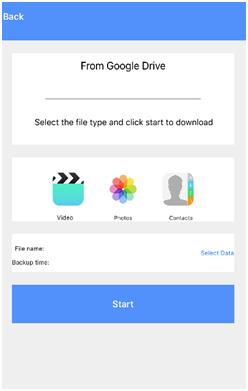
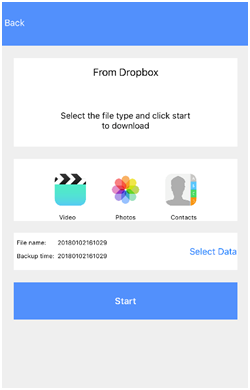


图5-6 Dropbox还原图 图 5-7 Google Drive还原图

由于iOS的控件switch不提供自定义功能，没用方法替换图片，为了达到图中自定义开关的功能，这里使用按钮来完成。具体的实现思路是准备好开关两种不同的图片样子，并设置其点击事件，每当用户点击时，就变换当前显示的样式，当显示为黑白时代表该关闭已经打开用户不需要交进行该文件的还原，显示为彩色时代表该开关已经打开用户需要交进行该文件的还原，如图5-8所示。在还原进行时需要通过按钮当前显示的图片标识符进行判断。



图5-8选择文件类型图

选择文件核心代码：

if(self.charge1==nil||[self.charge1 isEqualToString:@"on"])

{

[self.pickure setImage:[UIImage imageNamed:@"pictureoff"] forState:UIControlStateNormal];

self.charge1=@"off";

}

else{

[self.pickure setImage:[UIImage imageNamed:@"pictureon.png"] forState:UIControlStateNormal];

self.charge1=@"on";

}

由于API的原因Dropbox可以在恢复时直接找到最近一次的备份记录并设为默认选择，而Goole Drive[12]则要求用户手动选择一次备份记录，且会提升用户进行选择功能的进行。

选择界面的实现需要调用API的查询功能，通过特定的标识符查复合条件的文件，然后将所有符合的文件名加入到数组中，然后建立一个tableview界面，界面中的每个cell用数组中的数据填充，tableview中的每个数据都由Backup Time:开头，这是还原中使用到的key值，后面的数字是时间，由NSData生成，在这里采用常用格式。具体如图5-9所示。

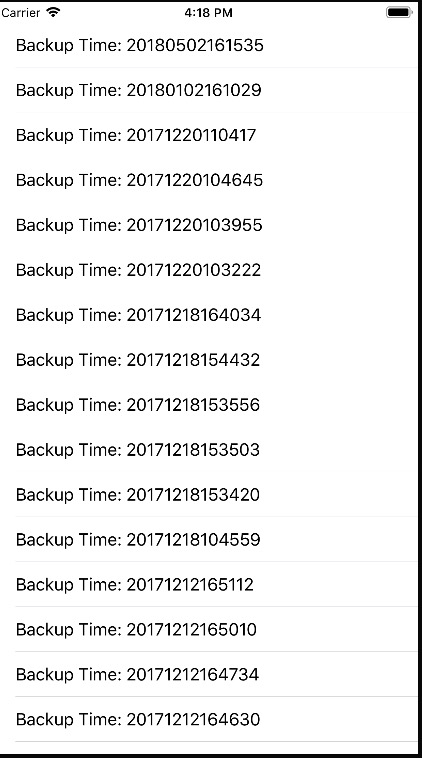


图5-9选择界面图

还原完成后数据将转换为NSDate格式，之后转换为手机上能够使用的UIimage,NSfilemage格式以便存入手机系统中。存储过程中需要调用手机的通讯录以及相册。

还原功能核心代码:

UIAlertAction \*cancelAction = [UIAlertAction actionWithTitle:@"Cancel" style:UIAlertActionStyleCancel handler:^(UIAlertAction \*action){

}];

UIAlertAction \*okAction = [UIAlertAction actionWithTitle:@"OK" style:UIAlertActionStyleDefault handler:^(UIAlertAction \*action){

[self performSegueWithIdentifier:@"toSeclect" sender:nil];

另外，用户每次进行功能时都需要检测用户是否需要登陆账号以及存储账号信息，存储信息使用沙盒功能，检测使用API自带方法，该方法需要将跳转相应公司的WEB端登录界面。

检测是否登陆账号核心代码：

if ([authResult isSuccess]) {

NSString \*surl=[url absoluteString];

[Backt setUrl:surl];

[Backt setObject:Backt path:[self documentPath]];

NSLog(@"Success! User is logged into Dropbox.");

[[NSNotificationCenter defaultCenter] postNotificationName:@"allow" object:nil userInfo: [NSDictionary dictionaryWithObject:@"allow" forKey:@"message"]];

} else if ([authResult isCancel]) {

NSLog(@"Authorization flow was manually canceled by user!")}

由于API的原因存储信息使用的沙盒文件主要用于Google Drive[13]，且最核心的功能是存储账号的key值。

存储用户数据核心代码：

- (void)encodeWithCoder:(NSCoder \*)aCoder{

[aCoder encodeObject:\_addressTime forKey:@"AccountId"];

[aCoder encodeObject:\_picktureTime forKey:@"Key"];

[aCoder encodeObject:\_filecount forKey:@"Time"];

}

- (id)initWithCoder:(NSCoder \*)aDecoder{

self = [super init];

if (self) {

\_addressTime = [aDecoder decodeObjectForKey:@"AccountId"];

\_picktureTime = [aDecoder decodeObjectForKey:@"Key"];

\_filecount =[aDecoder decodeObjectForKey:@" Time"];

## 本章总结

本章为系统的实现过程，是论文的核心部分，介绍了系统中存在的各个界面以及它们的相互联系，最为重要的是给出了各个界面以及功能实现的方式，步骤和实现功能的核心代码。

# 系统测试

软件开发过程中因为各种原因总会有一些预料之外且无法避免的错误，这是因为开发者不可能做到十全十美来避免遇到的所有错误，因此肯定会有一些难以发现的问题，所以我们要在测试中发现问题并改正，这样才能够降低软件在使用后出现问题。

## 测试环境

硬件环境：iMac；

处理器：2GHz Intel Core i5 处理器；

内存：8 GB 2400 MHz DDR4；

硬盘空间：1TB (5400-rpm) 硬盘；

操作系统：macOS High Sierra 10.13 ；

编译软件：Xcode；

## 测试目标

系统测试是对整个产品的健壮性、安全性以及各性能参数的测试，各模块的验证性测试和功能性测试[13]。

为了给用户最好的体验，改系统严格按照标准进行测试，首先对主要功能进行测试，其次对细节方面测试，找出了一些潜在的问题。尽量做到没有错误。

## 测试方法

软件测试中常用的两种方式黑盒测试和白盒测试。

### 功能性测试

功能性测试即为黑盒测试，测试程序是否能在接受数据后产生正确的输出，以完成目的功能[14]。

（1）登陆功能是由API提供的，用于检测用于账号是否登陆，如未登陆将调账登陆界面，若已经登陆则直接跳转相应的备份和还原功能界面，同时一次登陆后如果用户不注销应能在下次启动时不需要登陆，登录功能测试如表6-1。

表6–1 登录测试表

| 测试过程：用户点击使用Google Drive或Dropbox备份且为登录账号时 | | | |
| --- | --- | --- | --- |
| 序号 | 输入 | 预期结果 | 是否通过 |
| 1 | 账号和密码为空 | 提示输入账号和密码 | 通过 |
| 2 | 账号为正确账号 | 提示输入密码 | 通过 |
| 3 | 账号为正确账号，密码为正确密码 | 提示登录成功 | 通过 |
| 4 | 账号为正确账号，密码为错误密码 | 提示密码错误 | 通过 |
| 5 | 用户已经登陆过账号且未注销 | 直接跳转功能界面 | 通过 |

（2）备份文件选择时，但用户没有选择文件时应提示用户选择文件，同时需要达到用户可以选择任意一种或多种文件的要求。备份文件类型选择测试如表6-2。

表6-2备份文件类型选择测试表

| 测试过程：用户在备份界面选择文件类型时 | | |
| --- | --- | --- |
| 输入 | 预期结果 | 是否通过 |
| 不选择任何文件，点击“开始按钮” | 提示没有选择文件 | 通过 |
| 选择任意一种文件，点击“开始按钮” | 开始备份 | 通过 |
| 选择任意两种文件，点击“开始按钮” | 开始备份 | 通过 |
| 选择全部文件，点击“开始按钮” | 开始备份 | 通过 |

（3）还原是首先需要进行记录选择，用户必须已经进行过备份并且用记录，该项测试主要进行选择文件时的测试，测试用户在不选择记录以及选择记录异常的时候会出现什么情况，出现的情况是不是预计的情况，并对出现的情况对用户进行说明，备份记录选择测试如表6-3。

表6-3备份记录选择测试表

| 测试过程：用户在还原界面选择备份记录时 | | |
| --- | --- | --- |
| 输入 | 预期结果 | 是否通过 |
| 不选择记录 | 提示没有选择，并询问是否跳转选择界面 | 通过 |
| 选择任意一次备份记录且该记录的文件没有被删除 | 开始还原 | 通过 |
| 选择任意一次备份记录且该记录的文件已经被删除 | 提示文件已经被删除 | 通过 |

（4）读取手机数据测试如表6-4。

表6-4读取手机数据测试表

| 测试过程：用户在还原过程中将数据存入手机过程 | | |
| --- | --- | --- |
| 输入 | 预期结果 | 是否通过 |
| 软件有通讯录，相册权限 | 提示备份成功，数据已经存入网盘 | 通过 |
| 软件没有通讯录权限，且需要备份通讯录 | 提示需要开启通讯录权限 | 通过 |
| 软件没有相册权限，且需要备份相册或视频 | 提示需要开启相册权限 | 通过 |

（5）数据存入手机测试如表6-5。

表6-5数据存入手机测试表

| 测试过程：用户在还原过程中将数据存入手机过程 | | |
| --- | --- | --- |
| 输入 | 预期结果 | 是否通过 |
| 软件有通讯录，相册权限 | 提示还原成功，数据已经存入手机 | 通过 |
| 软件没有通讯录权限，且有文件需要存入通讯录 | 提示需要开启通讯录权限 | 通过 |
| 软件没有相册权限，且有文件需要存入相册 | 提示需要开启相册权限 | 通过 |

### 结构性测试

结构性测试即为白盒测试[15]，与黑盒测试的最大区别是在完全了解了程序的结构和处理过程的前提下，按照软件内部的逻辑来测试每条通路是否能达到理想目标。白盒测试的独特之处在于是根据源程序代码进行。

## 测试总结

本次测试使用的方式为黑盒测试，主要测试了用户使用过程中主要功能是否正确，能发正确跳转各个界面等。测试完成后效果良好，用户评论好评率较高。

结束语

经过数个月的一边学习一边工作，终于这款iOS手机数据备份的软件基本完成了，在这几个月期前，有一些不真实的感觉。在整个开发期间，遇到了大大小小无数的问题，能解决的就解决，不能解决的网上查找资料解决，国内查不到就去国外，国外查不到就加群去问经验丰富的开发者。虽然问题很多，但也就是这样的问题让我知道在开发iOS软件时遇到问题如何去分析问题，读取问题，然后怎么去解决它。虽然可能一个别人看起来很简答的问题在我这需要好久才能解决，但是问题使我进步，遇到的问题越多，我收获的也就越多。

在这期间，我不光对进行了设计，编码，自我总结，还撰写了该论文，撰写过程中，我不断的进行研究各种文档以对自己的论文进行完善和修复，但是占用时间最多的地方在于资料的查阅。现在回过头看来，这款软件的开发并不很轻松，我不仅要自己建立环境，还要重头学习iOS开发的很多的常用知识。

这次的毕业设计对我来说无疑是收获巨大的，从几个月前的对iOS开发一无所知，到现在的能够上架自己的参与应用到Appstore盈利，不仅让我学习到了iOS开发的相关技术，更让我对iOS开发产生了浓厚的兴趣，学习到知识的感觉是如此美好。但是我知道自己还是有很多的不足，我会在日后的学习中逐渐完善。

参考文献

1. 孙晓文. iOS与Android操作系统的优缺点比较[J]. 无线互联科技, 2013(12):51-51.
2. 钱克苍. 基于IOS平台的汽车安全驾驶系统[J]. 软件工程, 2014, 17(8):58-62.
3. 刘彦. 基于IOS环境的应用软件交互设计策略[J]. 信息化研究, 2011, 37(5):24-26.
4. 单平平, 邢静宇. 基于iOS平台的婚礼客户端的设计与实现[J]. 南阳理工学院学报, 2015, 7(6):43-46.
5. 关东升. iOS开发指南:从零基础到App Store上架.第3版[M]. 人民邮电出版社, 2015.
6. 李柏岚, 谷大武, 李卷孺,等. iOS备份机制中隐私威胁问题的分析[J]. 通信技术, 2012, 45(2):25-28.
7. 盛蒙蒙, 钟鸣. 基于IOS缩略图文件的数据恢复研究和开发[J]. 计算机时代, 2015(5):19-21.
8. 臧磊. API接口浅析[J]. 电信网技术, 2004(3):20-23.
9. Daryabar F, Dehghantanha A, Eterovic-Soric B, et al. Forensic investigation of OneDrive, Box, GoogleDrive and Dropbox applications on Android and iOS devices[J]. Australian Journal of Forensic Sciences, 2016, 48(6):615-642.
10. 刘畅. 基于iOS的国际酒店预订APP的设计与实现[D].北京：北京交通大学, 2015.
11. 刘乐廷, 李敬兆, LIULe-ting,等. IOS内存开发管理机制的研究[J]. 计算机与现代化, 2013(3):196-199.
12. 李聪聪. 基于iOS的互动电子周刊[D]. 天津：天津大学, 2012.
13. RON PATTON. 软件测试(英文版.第2版)[M]. 机械工业出版社, 2006.
14. 刘洋. 白盒测试技术概述[J]. 广西大学学报(自然科学版), 2008, 33(s1):146-149.
15. 向润. 黑盒测试方法探讨[J]. 软件导刊, 2009(1):33-35.

致谢

该论文在我的指导老师，惠征老师的帮助下完成。在这里我要由衷的感谢他，他不仅帮助我们完成论文的撰写，还无时无刻为了我们能够顺利完成毕业论文着想，每当有消息时总是提醒我们，并给我我们相关的建议帮助我们解决问题。同时老师严肃认真的科学态度，严禁的治学理念，精益求精的工作作风都深深的感染和激励着我们。

另外，我还要感谢每个同学们和老师，不仅不遗余力的帮我解决了问题，还提供了各种相关的软件以帮助我快速撰写论文。

最后，我要对所有帮助过我但是没有提到的人表示感谢，例如我的实习负责人，网上为我提供帮助的不知名网友，在这里，请允许我对你们表达诚挚谢意，感谢你们！