

ZaCloud

an iOS cloud notepad app

Introduction

- Cloud is widely used in current business
- It is a integration of server development and mobile development
- This project is a good opportunity to look inside of the cloud system

Introduction



Requirement

1. The note records should be stored in both locally and in an online “cloud” service.
2. Local client must be able to connect with server through internet.
3. The system requires a rudimentary user management scheme.
4. If internet connection lost, the local client should be able to store the updates and new notes temporarily, and synchronize them with the server after.
5. Deletion will be disabled while internet connection lost.
6. During the edit mode for a particular note, other devices should be blocked out of this note, and reject all of the synchronization from them.
7. Deletion and update require the lock. That means if a device is running offline from the very beginning, it won't be able to update and anything.

Tools and languages

- Server side:
 - Linux - Amazon EC2 cloud system with Linux Ubuntu 14.04 Long Term Support (LTS)
 - Apache 2 Server
 - MySQL database
 - PHP JSON
- Client side:
 - Xcode with Swift 2.0

Server Side: Database

- Two tables: users and notes
- MySQL was chosen due to prior experience
- MySQL server is a robust and well known software application that is common on the web
- Robust PHP libraries exist to work with MySQL

PHP Modules

- PHP pages runs on Apache Lamp
- Retrieves url query and abstract variables from it
- Returns JSON data
- mysqli used for connect to database
- JSON_encode for echo result as JSON format
- One PHP page for each function
- PHP pages power insertion, deletion, and update
- Can be coded in nano

Client Side: Basic Classes

- Mostly like note dictionary, and different kinds of log records
- Most of them must be wrote to the local plist
- Need to extend NSObject and NSCoding, and specify the forKey for plist item

plist

- This is a sample of plist
- It can store variables sorted by key
- User has to explicitly specify the key and type of each class member

Key	Type	Value
▼ Information Property List		
Localization native development region	String	en
Executable file	String	\$(EXECUTABLE_NAME)
Bundle identifier	String	\$(PRODUCT_BUNDLE_IDENTIFIER)
InfoDictionary version	String	6.0
Bundle name	String	\$(PRODUCT_NAME)
Bundle OS Type code	String	APPL
Bundle versions string, short	String	1.0
Bundle creator OS Type code	String	????
Bundle version	String	1
Application requires iPhone environment	Boolean	YES
Launch screen interface file base name	String	LaunchScreen
Main storyboard file base name	String	Main
► Required device capabilities	Array	(1 item)
► Supported interface orientations	Array	(3 items)
► App Transport Security Settings	Dictionary	(1 item)
► Supported interface orientations	Array	(4 items)

Local plist Copy Management

- Need NSObject and NSCoding in the basic classes
- Data types have to be serializable types such as NSDate, NSString etc.
- Need to define the forKey
- NSCoder encode the data, and NSKeyedArchiver save data as plist
- DecodeObjectForKey decode plist items, and initialize the object

Local plist Copy Management

- Need to define/get the path of files
- NSSearchPathForDirectoriesInDomains gets the file directory
- Then hand it over to read/write method

JSON Connections

- Sends url request and retrieves JSON data
- Each PHP page has a JSON connection cooperates with
 - The first thing is setup and decorate the url
 - NSURLConnection handle the network job, but the default setting is work in the background. It needs to configure

JSON Connections



The image shows two side-by-side screenshots. On the left, a code editor displays a JSON object with several fields: title, note, lastModified, createDate, author, locked, lockingTime, isDeleted, updateDevice, and lockingDevice. The value for updateDevice is a long string of characters. On the right, a web browser window shows the same JSON object, but the updateDevice field is displayed as a large block of text, indicating a content-length limit of 1024 bytes.

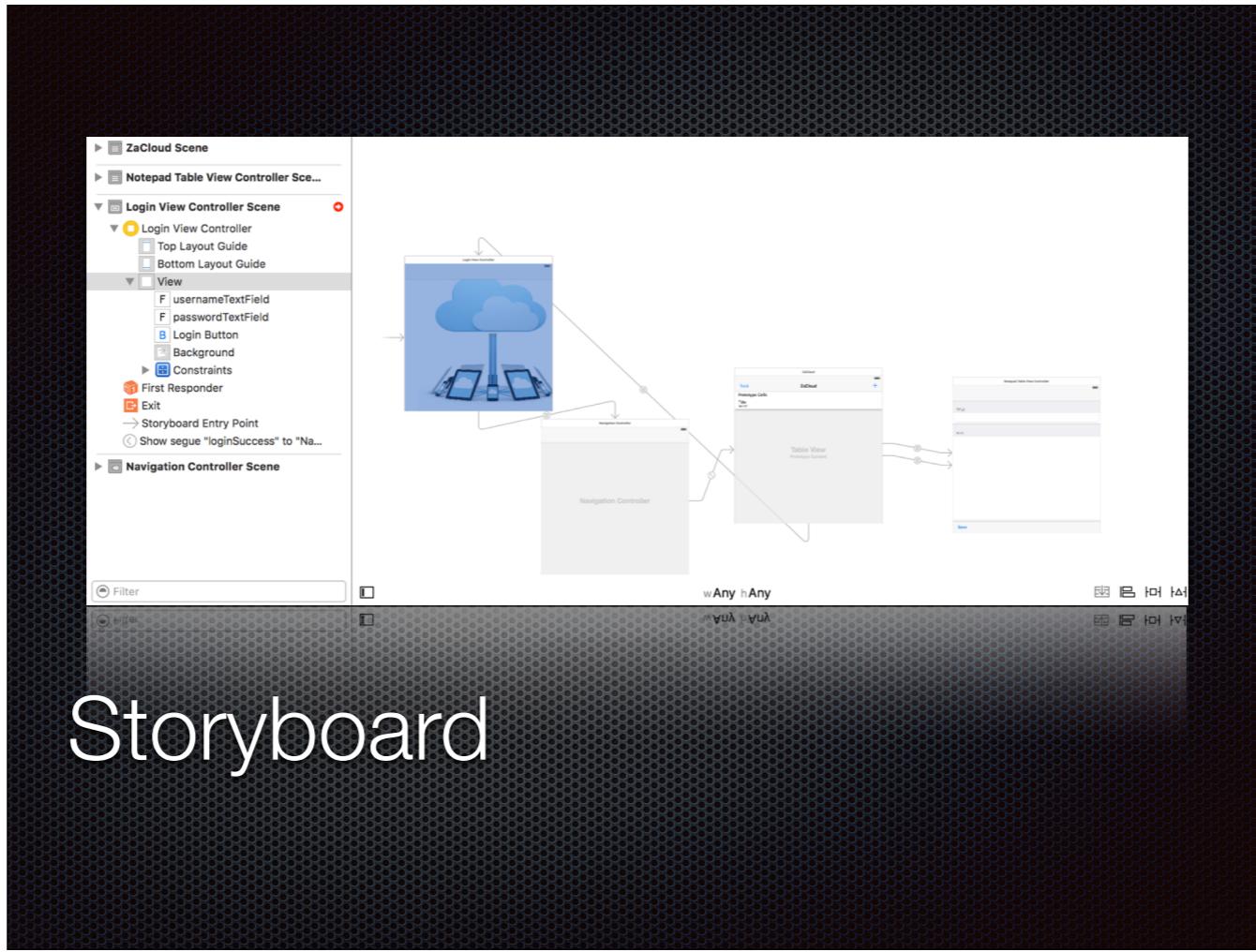
```
[{"title": "b", "note": "ddd", "lastModified": "2016-04-01 19:46:29", "createDate": "2016-04-01 19:46:29", "author": "zach", "locked": "0", "lockingTime": "0000-00-00 00:00:00", "isDeleted": "0", "updateDevice": "A982E995-20F9-4B77-9443-B73CDD5AC4B9", "lockingDevice": null}, {"title": "c", "note": "", "lastModified": "2016-04-01 19:46:21", "createDate": "2016-04-01 19:46:21", "author": "zach", "locked": "0", "lockingTime": "0000-00-00 00:00:00", "isDeleted": "0", "updateDevice": "95FE4188-BDA7-429B-9DBB-EAC8402B80199", "lockingDevice": null}, {"title": "a", "note": "Hello world", "lastModified": "2016-04-01 19:44:45", "createDate": "2016-04-01 19:43:30", "author": "zach", "locked": "1", "lockingTime": "2016-04-01 19:46:19", "isDeleted": "0", "updateDevice": "C8E90590-8C7F-4636-8DFE-0A5D567D1DE6", "lockingDevice": "A982E995-20F9-4B77-9443-B73CDD5AC4B9"}, {"title": "b", "note": "Ddd", "lastModified": "2016-04-01 19:46:29", "createDate": "2016-04-01 19:46:29", "author": "zach", "locked": "0", "lockingTime": "0000-00-00 00:00:00", "isDeleted": "0", "updateDevice": "A982E995-20F9-4B77-9443-B73CDD5AC4B9", "lockingDevice": null}, {"title": "c", "note": "", "lastModified": "2016-04-01 19:46:34", "createDate": "2016-04-01 19:46:34", "author": "zach", "locked": "0", "lockingTime": "0000-00-00 00:00:00", "isDeleted": "0", "updateDevice": "A982E995-20F9-4B77-9443-B73CDD5AC4B9", "lockingDevice": null}]
```

Inner Functions

- Two kinds of inner functions: simple assistance function, and complex function
- Simple functions: DateTimeFromString, LockingTimeGenerator, isConnectedToNetwork
- Complex functions: sendLocking, synchronization, loadData, Login

UIViews

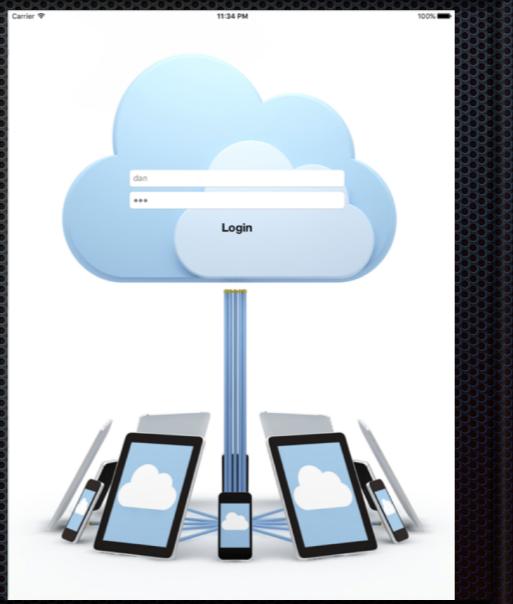
- All of the UIViews have been edited in storyboard
- Functions have been defined in UIViewController or UITableViewController
- Segue for trigger the next view



Storyboard

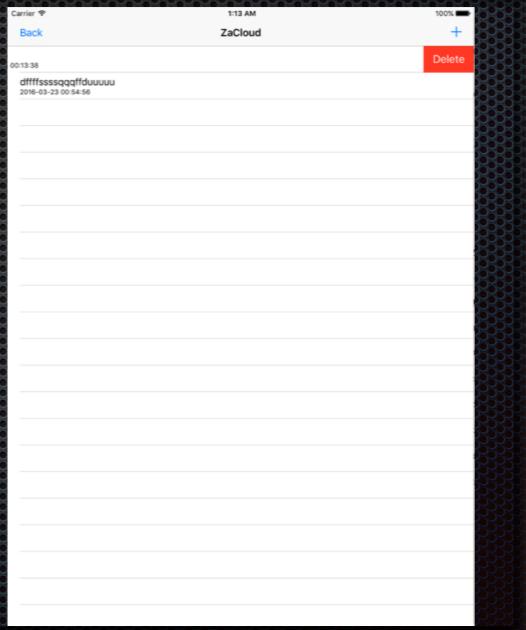
Login View

- Retrieves username and password
- Sends them to the server
- Triggers the segue to Index view if possible
- Syncs to server every minute



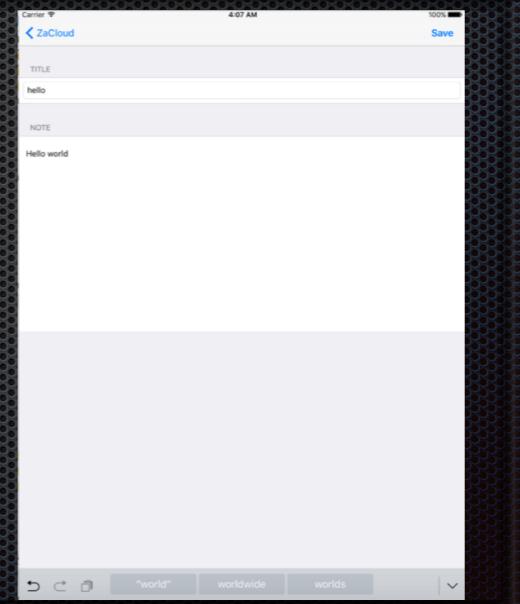
Notepad Index View

- Main view of browser notes' title
- Pull down to sync
- Slide left to show the delete button
- Can edit existed notes and create new notes
- Should load note's data is depended on segue.identifier



Notepad View

- The place that check and edit data
- User can only check note if lock request failed
- It refreshes lock every 110 seconds, if user got it in the first place
- Save function sits in unwind segue



Instruction Dispatch

- iOS has powerful parallel computing ability, but it causes problems for developers
- Grand Central Dispatch has been introduced solve this problem
- Two way of GCD have been implemented in this project: semaphore, dispatch_group

GCD: Semaphore

- Usage: functions we really wish it runs in background, and functions that need to return value after JSON connection
- Semaphore_signal to rise the signal
- Semaphore_wait to lower the signal, and it will hold the resource forever until instruction rise the signal

GCD: dispatch_group

- Some of the JSON Connection functions have to be finished before tableView.reloadData()
- dispatch_group has been implemented for this issue
- The thread will be congested if multiple processes try to enter the group and grab the resource, and the whole program will stuck there forever
- dispatch_group_enter and dispatch_group_leave to indicate a single task status
- dispatch_group_wait waits for every task is done, then continues executing

GCD: group_dispatch

```
dispatch_group_enter(self.syncGroup)
NSURLSession.sharedSession().DataTaskWithURL(url, completionHandler:
{ (data: NSData?, response: NSURLResponse?, error: NSError?) -> Void in
    do {
        if(data == nil){
            print("nil")
            return
        }
        let jsonDictionary = try NSJSONSerialization.JSONObjectWithData(data!, options: [NSJSONReadingOptions()])
        print("num of updated note: ", jsonDictionary.count)
        var i:Int
        var newNote: Note
        //print(jsonDictionary.count)
        for i=0; i<jsonDictionary.count; i++ {
            //print(jsonDictionary[i]["title"])
            newNote = Note()
            newNote.note = jsonDictionary[i]["title"] as! NSString
            newNote.note = jsonDictionary[i]["note"] as! NSString
            newNote.author = jsonDictionary[i]["author"] as! NSString
            if jsonDictionary[i]["createDate"] != nil{
                newNote.createDate = self.stringToDate.convertStringtoNSDate(jsonDictionary[i]["createDate"])
            }
            if jsonDictionary[i]["lastModified"] != nil{
                newNote.lastModified = self.stringToDate.convertStringtoNSDate(jsonDictionary[i]["lastModified"])
            }
            if jsonDictionary[i]["lockingTime"] as! String != ""{
                newNote.lockingTime = self.stringToDate.convertStringtoNSDate(jsonDictionary[i]["lockingTime"])
            }
            newNote.updateDevice = jsonDictionary[i]["updateDevice"] as! String
            var index = self.getNoteIndex(newNote.createDate)
            if index != -1 {
                self.notes[index]=newNote
            }else{
                self.notes.append(newNote)
            }
        }
    }catch{
        print("oops")
    }
    self.tableView.reloadData()
    self.saveNotesToLocal()
    dispatch_group_leave(self.syncGroup)
}).resume()
```

```
read sync date test: 2016-04-02 07:25:24 +0000
sync deletion record from server
sync update notes from server
sync new notes from server
end of sync
wait is over
save sync date test: 2016-04-02 07:26:43 +0000
num of delete: 0
num of updated note: 0
num of new note: 0
```

```
read sync date test: 2016-04-02 07:21:10 +0000
sync deletion record from server
sync update notes from server
sync new notes from server
end of sync
num of delete: 0
num of updated note: 0
num of new note: 0
wait is over
save sync date test: 2016-04-02 07:22:41 +0000
```

Future Extensions

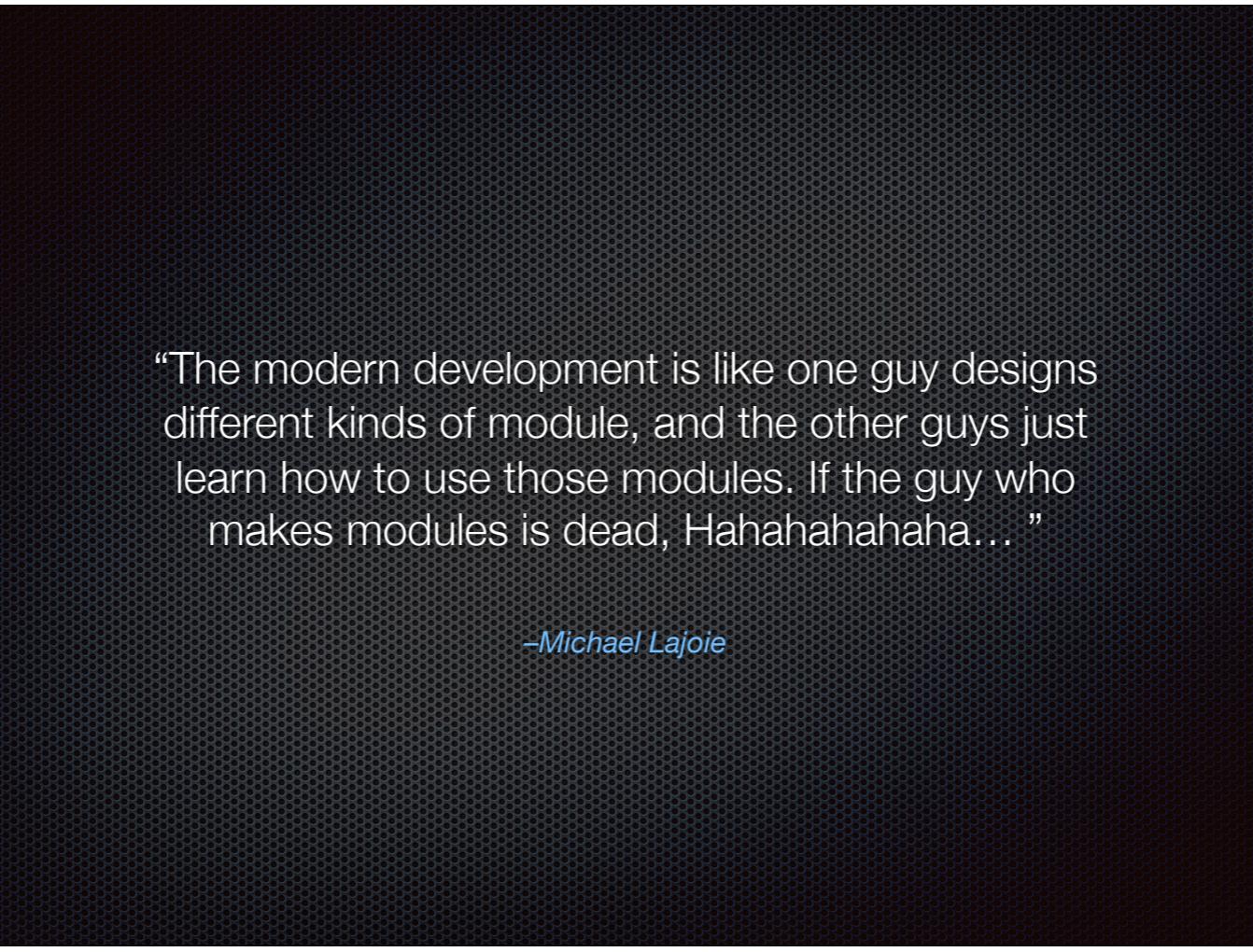
- Substitute plist by Core Data
- Optimize instruction dispatch
- Reduce database redundancy
- Improve synchronization, increase its refreshing frequency at the critical time
- Better internet error protection

Special Thanks

- Mr. Daniel Friyia
- Professor Michael Lajoie

Special Thanks

- Mr. Steve Jobs



“The modern development is like one guy designs different kinds of module, and the other guys just learn how to use those modules. If the guy who makes modules is dead, Hahahahahaha...”

-Michael Lajoie