|  |  |  |
| --- | --- | --- |
| 文档状态 | 保密级别 |  |
| [ √ ]草稿  [ ]修订  [ ]发布 | 文档编号 |  |
| 管理部门 |  |
| 修订年月 |  |
| 版本号 |  |

**MyWeibo\_App系统开发与设计汇总报告**

**系统竣工报告**

|  |  |  |
| --- | --- | --- |
| **修订人签字** | **审核人签字** | **批准人签字** |
| 日期： | 日期： | 日期： |

变更履历

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| **序号** | **变更日期** | **版本** | **变更位置** | **变更原因** | **修订人** | **审核人** | **批准人** |
| 1 |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |

说明：“变更原因”主要是分为：

1. 建立初稿
2. 内容修订
3. 正式发布

目录

[1. 概要 4](#_Toc369787964)

[2. 项目开发 4](#_Toc369787965)

[3. 硬件 4](#_Toc369787966)

[4. 测试 4](#_Toc369787967)

[5. 培训 4](#_Toc369787968)

[6. 提交成果 4](#_Toc369787969)

[7. 附件 5](#_Toc369787970)

[7.1. 用户签字 5](#_Toc369787971)

[7.2. 项目经理签字 5](#_Toc369787972)

# 开发计划

1.1

本文档是"小a/小b/小c"根据公司为提升公司在社交网络需求的技术预演方案。

主要针对当前热门ios/swift术加以分析理解后编写预研学习成果以及与社交网络务如何结合共同提出一种更具有第三方微博 ios 客户端解决方案。

1.2.文档范围

本文档绝不可代替总体设计、详细设计、代码设计、测试方案、操作手册等项目其他环节之产出。

为更好地理解文档内容，建议读者查阅所列的参考资料，以了解与系统相关的更多信息

1.3 阅读人员

本文档适用于需求人员、设计人员、开发人员、测试人员、部署实施人员、项目合同规定的其他文档用户及相关人员获知需求。

1.4 使用环境

osx 系统，app环境为 ios 8.0+

客户端app在iOS等各操作系统native app

1.5 项目风险

介项目实施过程中，存在的风险可能来自于以下方面：

需求风险：开发过程中有出现业务需求变更的可能。原则上方案部门需要对本需求说明书进行业务需求确认。

若出现重大变更，考虑实施需求变更流程或中止项目。

技术风险：目前方案中，不关心用户的具体内容。用户如果在使用我们第三方weibo客户端时候，此方案有法律风险，需早期与法律专家确认

# 需求分析

2.1

本文档描述软件产品需求规格说明书（SRS）的目的是：

1) 定义软件总体要求，作为用户和软件开发人员之间相互了解的基础；

2) 提供性能要求、初步设计和用户影响的信息，作为软件人员进行软件结构设计和编码的基础；

3) 作为软件总体测试的依据。

2.2.文档范围

需求规格说明书主要包含了该系统整体需求及功能性需求的详细介绍

2.3 阅读人员

本文档适用于需求人员、设计人员、开发人员、测试人员、部署实施人员、项目合同规定的其他文档用户及相关人员获知需求。

2.4 使用环境

osx 系统，app环境为 ios 8.0+

客户端app在iOS等各操作系统native app

2.5 功能性需求

1. 注册功能UI（可点击，可跳转）

2. 登录功能UI （用户名和密码，跳转到 微博图文list页）

3. 微博图文list页，上下滑动，图文和模拟转发的文字列表）

4. 微博长文收缩和展开特效

5. 微博图文list页的图片可以detail查看，图片触摸点双击放大缩小

6. 个人页profile的一个html5跳转展示

# 系统设计

软硬件部署环境：Xcode+ios8





# 3. 系统设计

1. 注册功能UI（可点击，可跳转）

功能描述：用户可以通过ios app注册功能

参与用户：普通的第三方weibo使用者

输入：email/手机号

操作：注册的账号

扩展和异常：无

输出：成功/失败

2. 登录功能UI （用户名和密码，跳转到 微博图文list页）

功能描述：用户可以通过ios app登录功能

参与用户：普通的第三方weibo使用者

输入：用户名和密码

操作：登录ios 第三方微博app，查看微博信息

扩展和异常：无

输出：登录成功/失败

3. 微博图文list页，上下滑动，图文和模拟转发的文字列表）

功能描述：微博图文list页图文列表（富媒体展示）

参与用户：普通的第三方weibo使用者

输入：无法输入

操作：点击关注或者不关注

扩展和异常：无

输出：富媒体图文列表

4. 微博长文收缩和展开特效

功能描述：微博长文收缩和展开特效

参与用户：普通的第三方weibo使用者

输入：无

操作：点击“更多”

扩展和异常：无

输出：更多信息的展示

5. 微博图文list页的图片可以detail查看，图片触摸点双击放大缩小

功能描述：微博图文list页的图片可以detail查看，图片触摸点双击放大缩小

参与用户：普通的第三方weibo使用者

输入：无

操作：点击查看detail页

扩展和异常：无

输出：查看detail页具体的图文信息或者gif信息

6. 个人页profile的一个html5跳转展示

功能描述：个人页profile的一个html5跳转展示（github/其他第三方网站展示页面）

参与用户：普通的第三方weibo使用者

输入：无

操作：点击

扩展和异常：无

输出：具体第三方账号展示

# 数据设计

WeiboItem

headPic

nickName

time

source

title

images

User

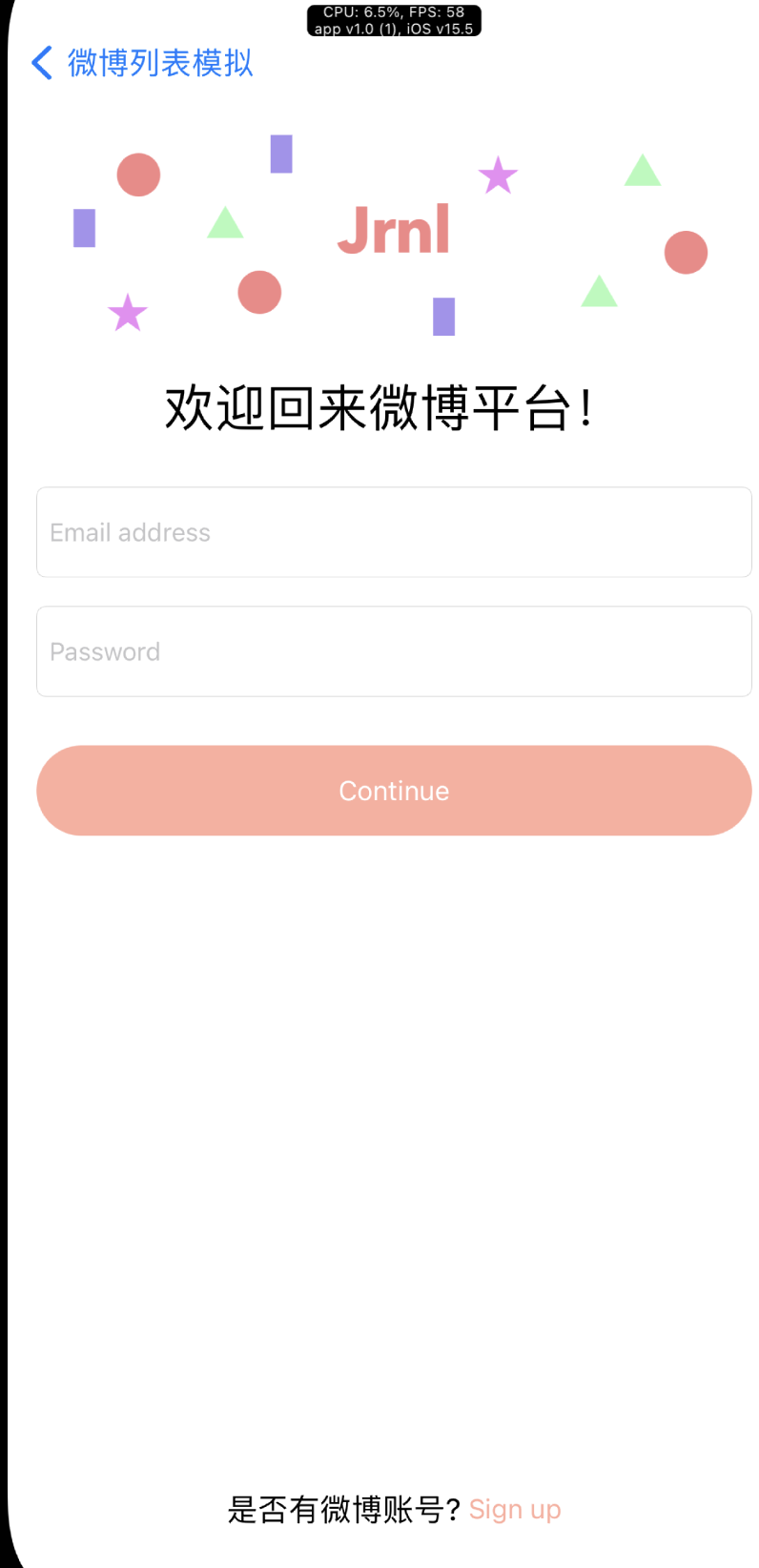
email

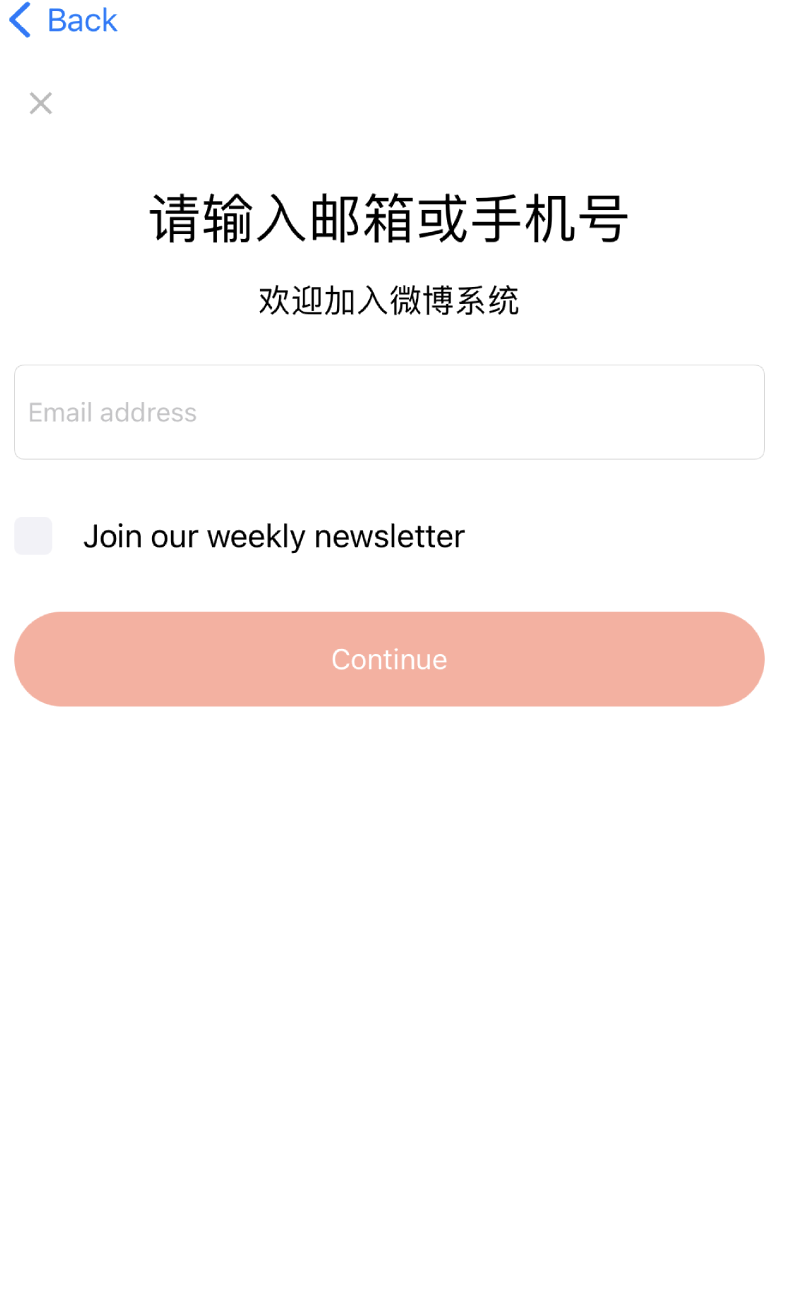
phone

password提交了《XXX培训文档》和《XXX系统用户手册》，对ROW工程师、ONECALL调度、运营维护工程师进行了使用培训。

提交了《XXX系统安装维护手册》，对IT系统管理员进行了系统维护培训。

# 功能展示













# 测试和测试结果

测试工程师

负责编写项目测试用例

负责搭建测试环境

负责测试用例执行

负责跟踪BUG

测试负责人

测试用例评审、测试报告评审

负责编写项目测试计划

负责编写测试报告

性能测试工程师

负责编写项目测试脚本

负责测试脚本执行

负责跟踪性能BUG及解析

负责编写测试报告

SwiftStudyTests/SwiftStudyTests.swift

class SwiftStudyTests: XCTestCase {

override func setUp() {

// Put setup code here. This method is called before the invocation of each test method in the class.

}

override func tearDown() {

// Put teardown code here. This method is called after the invocation of each test method in the class.

}

func testExample() {

// This is an example of a functional test case.

// Use XCTAssert and related functions to verify your tests produce the correct results.

}

func testPerformanceExample() {

// This is an example of a performance test case.

self.measure {

// Put the code you want to measure the time of here.

}

}

}

SwiftStudyTests/SwiftStudyUITests.swift

class SwiftStudyUITests: XCTestCase {

override func setUp() {

// Put setup code here. This method is called before the invocation of each test method in the class.

// In UI tests it is usually best to stop immediately when a failure occurs.

continueAfterFailure = false

// UI tests must launch the application that they test. Doing this in setup will make sure it happens for each test method.

XCUIApplication().launch()

// In UI tests it’s important to set the initial state - such as interface orientation - required for your tests before they run. The setUp method is a good place to do this.

}

override func tearDown() {

// Put teardown code here. This method is called after the invocation of each test method in the class.

}

func testExample() {

// Use recording to get started writing UI tests.

// Use XCTAssert and related functions to verify your tests produce the correct results.

}

}

# 提交成果

| **序号** | **成果名称** |
| --- | --- |
| 1 | 《01项目开发计划》 |
| 2 | 《02需求分析规格说明》 |
| 3 | 《03系统设计文档》 |
| 4 | 《04数据设计文档》 |
| 5 | 《05测试计划和结果分析》 |
| 6 | 《06项目总结》 |
| 7 | 《07项目敏捷管理和分工进度表.xlsx》 |
| 8 | 《08项目实施进度计划及甘特图GANT》 |
| 9 | 《readme部署》 |
|  |  |

# 附件

## 用户签字

|  |
| --- |
| **日期：** |

## 项目经理签字

|  |
| --- |
| **日期：** |