SJTU BUSS

测试用例

版本 <1.0>

、

修订历史记录

|  |  |  |  |
| --- | --- | --- | --- |
| **日期** | **版本** | **说明** | **作者** |
| <日/月/年> | <x.x> | <详细信息> | <姓名> |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |

目录

1. 简介 2

1.1 目的 2

1.2 定义、首字母缩写词和缩略语 2

1.3 参考资料 2

2. 整体说明 2

3. 具体需求 2

2

软件需求规约 (简化版)

# 简介

## 目的

[阐明此 **SRS** 的目的。**SRS** 应详细地说明所确定的应用程序或子系统的外部行为。它还要说明非功能性需求、设计约束以及提供完整、综合的软件需求说明所需的其他因素。]

## 定义、首字母缩写词和缩略语

[本小节应提供正确理解此 **SRS** 所需的全部术语的定义、首字母缩写词和缩略语。建议单独撰写项目术语表，本处只需要参见该术语表即可。]

## 参考资料

[本小节应完整列出此 **SRS** 中其他部分所引用的任何文档。每个文档应标有标题、报告号（如果适用）、日期和出版单位。列出可从中获取这些参考资料的来源。这些信息可以通过引用附录或其他文档来提供。]

# 安卓客户端单元测试

# 安卓客户端集成测试

## 用户登录测试

前置条件：用户未登录（司机、管理员同理）

测试输入：输入手机号码、密码

观察点：点击“登录”

控制点：手机号码是否已注册；手机号码与密码是否匹配

预期结果：若手机号未注册，提示“该账号不存在哦~”

若手机号码与密码不匹配，提示“手机号码或密码填写不正确哦~”

若正确，提示“登陆成功~”

若手机号码已经登录，提示“该账号已经登录了哦~”

## 注册测试

前置条件：用户未登录

测试输入：输入错误的信息

观察点：点击“注册”

预期结果：若用户名未填写，提示“用户名不能为空哦~”

若手机号码未填写，提示“手机号码不能为空哦~”

若手机号码不合规范，提示“请输入正确的手机号码哦~”

若密码为空，提示“密码不能为空哦~”

若密码非6-16位，提示“请设置6-16位的密码哦~”

若两次密码输入不一致，提示“两次密码输入不一致哦~”

若验证码为空，提示“验证码不能为空哦~”

若未勾选协议，提示“请确保您已阅读并同意《SJTUBUS用户协议》哦~”

若验证码输入错误，提示“验证失败”

## 忘记密码测试

前置条件：用户进入登录界面

观察点：用户点击忘记密码

预期结果：还没做

## Jaccount登录测试

前置条件：用户未登录

测试输入：输入jaccount用户名、密码以及验证码

观察点：点击“登录”

控制点：输入信息是否正确

预期结果：若正确，提示“登陆成功”

否则，显示错误提示

HINT点击忘记密码和创建jaccount账号是白页，但这没办法

## 预约单程班车测试1

前置条件：用户已登录，若未登录提示“请先登录~”

测试输入：调整单程的起点，终点

观察点：点击“查询”

预期结果：若起点终点相同，提示“起点和终点不能相同哦~”

若起点终点为徐汇和七宝的连线，提示“没有徐汇到七宝的班车哦~”

若试图调时间到今天之前，提示“不能预约已经发出的班次哦~”

若正常，进入下一页面

## 预约单程班车测试2

前置条件：用户选择预约的线路和日期

测试输入：点击前一天，后一天，或直接调整日期

预期结果：若调到今天之前，提示“不能预约已经发出的班次哦~”

若当日班次是0，加一句“快去预约其他的班次吧~”

若点击班次信息，显示班次详细信息

若点击收藏，提示“收藏功能还不能使用哦~”

若点击预约，进入订单填写页面

## 预约单程班车测试3

前置条件：用户选择了班次

测试输入：用户调整预约提醒和特殊需求的设置

观察点：点击“预约”

预期结果：如果有座位，显示预约成功的提示框，再次进入界面查看剩余座位数减少

如果没座位或者预约失败，提示预约失败

HINT 设置那里要加一个“开启预约提醒”的按钮，绑定在手机或者邮箱上

HINT 为了体现预约提醒和特殊要求的作用，appoint表建议多加一个comment（较大改动）

## 预约往返班车测试1

前置条件：用户已登录，若未登录提示“请先登录~”

测试输入：调整往返的起点，终点，出发时间，返程时间

观察点：点击“查询”

预期结果：同预约单程班车测试1

若调整时间后回程早于去程，提示“回程的时间不能早于去程哦~”

## 预约单程班车测试2

前置条件：用户进入设置去程/返程界面

测试输入：调整去程的时间，选择班次

观察点：点击“预约”

预期结果：同预约单程班车测试2

若调整时间后回程早于去程，提示“回程的时间不能早于去程哦~”

若调整时间后回程和去程时间又重叠，提示“不能预约行程冲突的班次哦~”

BUG！！这里的跳转逻辑啊啊啊

## 班次信息测试1

前置条件：用户进入班次信息界面

测试输入：用户选择类型，点击具体的班次

观察点：点击“预约”

预期结果：默认的显示就是当天正确的类型

显示当天正常运行的班次及当天的首班车、末班车

WARN！用户根本想不到左上角那个按钮是干嘛的

WARN！用户也不一定想得到点班次可以看到详细的信息

HINT 节假日什么时候纳入考虑呢，这其实是和普通双休日一个概念的

HINT 找个地方提示一下上车地点啊！

## 班次信息测试2

前置条件：用户进入时刻表/路线站点界面

测试输入：如果是校内巴士，可选择站点查看不同站点的信息

预期结果：显示当前类型及线路条件下的出发时间，备注及站点信息

HINT 标题栏能不能滚动一下？最右边看不到的一直

BUG！！校园巴士的时间貌似不太对。找个时间跑一圈拍个照

## 路线查询测试

前置条件：用户点击进入路线查询界面

观察点：用户点击路线切换并且切换线路

预期结果：界面显示为起点和终点的最优路线，路线切换后线路图随之改变

## 实时位置测试1

前置条件：用户点击实时位置进入界面

测试输入：用户对地图进行移动、放缩，点击图标

预期结果：若用户对地图进行缩放，达到一定缩放比时会显示站点中文名

用户什么都不做，巴士每隔3-5秒刷新一次当前位置

若司机打开了司机定位，则在实时位置上可以看到司机的位置及其移动

HINT 在地图上加一个悬浮text，标注现在是实时还是模拟

## 实时位置测试2

前置条件：用户点击实时位置进入界面

观察点：点击站点标记，点击巴士

预期结果：点击站点标记显示站点相关的所有时刻信息，上拉可以全屏显示

点击巴士显示巴士的车牌和司机

## 管理员扫描预约码测试

前置条件：管理员已登录，正在扫描用户预约记录中的预约码

观察点：管理员扫描预约码

预期结果：若预约码正确，则提示“验证成功”

若扫描的不是预约码，则提示“验证错误”

## 司机定位测试

前置条件：已登录并且身份是司机

测试输入：点击开启定位

预期结果：显示定位已开启，用户手机在实时位置上可以get到司机

司机在屏幕熄灭的情况下应该也能继续，但是退出就不能继续被定位了

## 个人信息测试1

前置条件：用户已登录

测试输入：用户点击“退出登录”

预期结果：用户退出登录，侧拉栏显示已经退出登录

## 个人信息测试2

前置条件：用户已登录

测试输入：当用户手机号/学号/真实姓名为空时，用户点击进行编辑

预期结果：用户编辑个人信息确定后，退出重新进入界面个人信息已经更新

## 预约记录测试

前置条件：用户已登录且进入预约记录界面

测试输入：用户点击预约记录的操作，排序

预期结果：用户可以对预约记录进行排序和筛选

这个明天继续修补

## 取消预约测试

前置条件：用户在预约记录界面

测试输入：用户点击取消预约

预期结果：弹出对话框让用户确认取消预约，确认后预约记录消除此条记录，并在预约处座位回升，数据表的数据最好是加一个flag设置成0，但现在没精力的情况下可以直接删掉

明天后端连上应该就能用了

## 预约提醒测试

前置条件：用户在预约记录界面

测试输入：用户点击预约提醒

预期结果：弹出对话框让用户确认预约提醒，并可以选择提醒的时间，并到时推送提醒

这个还没做

## 班次收藏测试

前置条件：用户在班次收藏界面

预期结果：用户可以看到自己收藏的班次，并且选择取消收藏

这个还没做

## 公告测试

前置条件：用户在我的消息界面

测试输入：后台发送一条公告

预期结果：用户在公告界面可以收到来自后台的公告

基本好了吧，不过没认真测试过

## 意见测试

前置条件：用户在用户反馈界面

测试输入：用户输入文字或者发送图片

预期结果：在界面上保留着用户发出的文字和图片，并且退出重新进入后仍然存在

## 设置测试

前置条件：用户在设置界面

测试输入：目前预计可以设置：是否在order界面提示预约提醒；账号切换；关于与帮助；主界面显示的功能

预期结果：这个还没做

# 后端单元测试

## StringCalendarUtils测试用例1

测试StringCalendarUtils类的stringToDate方法

测试输入：“2018-07-25”

预期结果：值为2018-07-25的date类

## StringCalendarUtils测试用例2

测试StringCalendarUtils类的stringToTime方法

测试输入：“2018-07-25 23:56:00”

预期结果：相应的值为2018-07-25 23:56:00应的date类

## StringCalendarUtils测试用例3

测试StringCalendarUtils类的dateToString方法

测试输入：值为2018-07-26的date类

预期结果：“2018-07-26”

## StringCalendarUtils测试用例4

测试StringCalendarUtils类的timeToString方法

测试输入：值为00:00:05的date类

预期结果：“00:00:05”

## StringCalendarUtils测试用例5

测试StringCalendarUtils类的isBeforeCurrentTime方法

1. 测试输入：”2018-07-25 23:56:00”

预期结果：true

1. 测试输入：”2018-07-26 00:00:02”

预期结果：true

1. 测试输入：”2019-01-01 12:00:00”

预期结果: false

## StringCalendarUtils测试用例6

测试StringCalendarUtils类的getCurrentDate方法

测试输入：无

预期结果：当前日期 eg：“2018-07-26”

## StringCalendarUtils测试用例7

测试StringCalendarUtils类的getCurrentTime方法

测试输入：无

预期结果：当前时间 eg “2018-07-26 00:00:05”

## ShiftUtils测试用例1

测试ShiftUtils类的getEngType方法

测试输入：“在校期-工作日”

预期结果：“NormalWorkday”

## ShiftUtils测试用例2

测试ShiftUtils类的getChiType方法

测试输入：“NormalWorkday”

预期结果：“在校期-工作日”

## ShiftUtils测试用例3

测试ShiftUtils类的getEngLineName方法

测试输入："校园巴士逆时针"

预期结果："LoopLineAntiClockwise"

## ShiftUtils测试用例4

测试ShiftUtils类的getChiLineName方法

测试输入："LoopLineAntiClockwise"

预期结果："校园巴士逆时针"

## AppointmentService测试用例1

测试AppointmentService类的getAppointInfo方法(预约班车界面)

测试输入：line\_name = "MinHangToQiBao";

type = "HolidayWorkday";

appoint\_date = "2018-07-27";

预期结果：相应的AppointInfo类, size = 2, get(1).shiftid = MQHD1700

## AppointmentService测试用例2

测试AppointmentService类的getAddAppointment方法

测试输入：lineNameCN = "闵行到七宝";

line\_type = "HolidayWorkday";

time = “17:00:00”;

date = “2018-07-27”

预期结果：相应的Appointments，size = 1

## AppointmentService测试用例3

测试AppointmentService类的verifyAppointment方法

测试输入：username = "姚子航";

departure\_date = "2018-07-27";

shift\_id = "MQHD1700";

预期结果：验证前 isNormal = false, 验证后 isNormal = true

## LineService测试用例1

测试LineService类的getStationByLineName方法

测试输入：line\_name = "MinHangToQiBao";

预期结果：相应的stations，get(0)是闵行，get(1)是七宝

## LineService测试用例2

测试LineService类的getLineInfo方法

测试输入：line\_name = "MinHangToQiBao"; type = "HolidayWorkday";

预期结果：相应的lineinfo类，首班车8:50，末班车17:00

## LineService测试用例3

测试LineService类的getAllLineName方法

测试输入：type = "HolidayWorkday";

预期结果：相应的list，size = 6

## ShiftService测试用例1

测试ShiftService类的getScedule方法

测试输入：String line\_name = "MinHangToQiBao";

String type = "HolidayWorkday";

预期结果：相应的schedule，第一班车8:50

## ShiftService测试用例2

测试ShiftService类的getShiftInfo方法

测试输入：String shift\_id = "MQHD1700";

预期结果：相应的shiftinfo, 出发时间17:00， 荷载50

## ShiftService测试用例3

测试ShiftService类的changeTypeToId方法

1. 测试输入："NormalWorkday"

预期结果：相应的shiftinfo, 出发时间17:00， 荷载50，类型是WD

1. 测试输入："NormalWeekendAndLegalHoliday"

预期结果：相应的shiftinfo, 出发时间17:00， 荷载50，类型是WE

1. 测试输入："HolidayWorkday"

预期结果：相应的shiftinfo, 出发时间17:00， 荷载50，类型是HD

## ShiftService测试用例4

测试ShiftService类的ChangeTimeToStringTime方法

测试输入：值为17:00的time类

预期结果：“1700”

## ShiftService测试用例5

测试ShiftService类的ModifySeat方法

测试输入：modifySeat(shift\_id,100);

预期结果：返回值1，成功

## ShiftService测试用例6

测试ShiftService类的GetTimeList方法

测试输入： String lineNameCn = "闵行到徐汇";

String type = "HolidayWorkday";

2)预期结果：相应的times，size = 2

## UserService测试用例1

测试UserService类的findByUserName方法

1）测试输入：String username =“ly”

预期结果：返回响应的user对象，其userId为3

2）测试输入：String username =“hello”

预期结果：返回Null

## UserService测试用例2

测试UserService类的findByJaccountUserName方法

1）测试输入：String username =“ly”

预期结果：返回响应的JaccountUser对象，其userId为3

2）测试输入：String username =“hello”

预期结果：返回Null

## UserService测试用例3

测试UserService类的findUserByPhone方法

1）测试输入：“12345678900”

预期结果：返回响应的User对象，其userId为21

2）测试输入：“00000000000”

预期结果：返回Null

## UserService测试用例4

测试UserService类的findJaccountUserByPhone方法

1）测试输入：“12345678900”

预期结果：返回响应的User对象，其userId为22

2）测试输入：“00000000000”

预期结果：返回Null

## UserService测试用例5

测试UserService类的listAllUsers方法

测试输入：无

预期结果：user列表，长度为8

## UserService测试用例6

测试UserService类的addUser方法

测试输入：String username=”xyx”;

String password=”password”;

Boolean isTeacher=false;

String phone=”13262600000”;

int credit=100;

预期结果：调用listAllUsers方法得到的user列表size比原来的增加1；

调用findByUserName方法查找xyx得到的结果非Null；

## UserService测试用例7

测试UserService类的modifyUser方法

测试输入：user.getId();

String username =”xyx”

String phone=user.getPhone();

Int credit=user.getCredit()-5;

预期结果：调用findByUsername方法查找xyx得到的用户的credit值比原来少5;

## UserService测试用例8

测试UserService类的deleteUser方法

测试输入：userId

预期结果：调用findByUsername方法查找xyx得到的用户为Null

## AdministratorService测试用例1

测试Administrator类的searchAdministrator方法

1. 测试输入：String username = “admin”

预期结果：得到一个非空列表，列表第一个元素的id为0；

1. 测试输入：String username = “abc”

预期结果：得到一个空列表

## AdministratorService测试用例2

测试Administrator类的saveAdministrator方法

测试输入：String username = “abc”

String password = “abc”

预期结果：调用findAll()方法的到的管理员列表的size比原来增加1；

调用searchAdministrator()方法查找abc得到非空列表；

## DriverService测试用例1

测试Driver类的getDriverInfo方法

1. 测试输入：String content=”driver”

预期结果：非空列表，其中每个元素的名字都包含“driver”关键词

1. 测试输入：String content=”1326”

预期结果：非空列表，其中每个元素的电话都包含“1326”关键词

## DriverService测试用例2

测试Driver类的addDriver方法

测试输入：String username=”driver3”;

String password=”driver3”

String phone=”13262600000”

预期结果：调用getDriverInfo的方法，得到的列表size比原来的增加1；

调用queryDriverByUsername方法，查找到的用户非空

## DriverService测试用例3

测试Driver类的modifyDriver方法

测试输入：String username=”driver3”;

String password=”driver3”

String phone=”12345678900”

预期结果：调用queryDriverByUsername方法，查找到的用户的电话为12345678900

## DriverService测试用例4

测试Driver类的deleteDriver方法

测试输入：userId;

预期结果：调用queryDriverByUsername方法，得到的用户为Null

# Web端后台管理系统接口测试

## 管理员登录接口测试

接口：http://localhost:8080/administrator/login?username=admin&password=admin

预期结果：状态为200，返回的json对象msg为“success”

## 管理员登出接口测试

接口：http://localhost:8080/administrator/logout

预期结果：状态码为200，返回的json对象msg值为“success”

## 判断管理员是否登录接口测试

接口：http://localhost:8080/administrator/judgestate

预期结果：状态码为200，若已登录则msg为“logged”，若未登录则msg为“not logged”

## 查找司机接口测试

1）查找用户名包含关键词的司机

接口：http://localhost:8080/driver/search?content=driver

预期结果：状态码为200，返回的json对象中driverList长度大于0

2）查找一个不存在的司机

接口：http://localhost:8080/driver/search?content=driver3

预期结果：状态码为200，返回的json对象中driverList长度等于0

## 添加司机接口测试

1）成功添加一个新司机

接口：http://localhost:8080/driver/add?username=driver3&password=driver3&phone=13262600000

预期结果：状态码为200，返回的json对象中msg值为“success”

2）添加一个用户名已存在的司机

接口：http://localhost:8080/driver/add?username=driver1&password=driver1&phone=13262600000

预期结果：状态码为200，返回的json对象中msg值为“existed”

## 修改司机信息接口测试

接口：http://localhost:8080/driver/modify?driverId=driver3&username=driver3&phone=12345678900

预期结果：状态码为200，返回的json对象中msg值为“success”

## 删除司机接口测试

接口：http://localhost:8080/driver/delete?driverId={{driverId}}

预期结果：状态码为200，返回的json对象中msg值为“success”

## 查找用户接口测试

1）根据姓名查找用户

接口：http://localhost:8080/user/search?content=ly

预期结果：状态码为200，返回的json对象中userList长度为1

2）根据电话查找用户

接口：http://localhost:8080/user/search?content=1326

预期结果：状态码为200，返回的json对象中userList长度为7

3）查找一个不存在的用户

接口：http://localhost:8080/user/search?content=xyx

预期结果：状态码为200，返回的json对象中userList长度为0

## 添加用户接口测试

接口：http://localhost:8080/user/add?username=xyx&password=password&phone=13262600000&credit=100&isTeacher=false

预期结果：状态码为200，返回的json对象中msg值为“success”

## 修改用户信息接口测试

接口：http://localhost:8080/user/modify?userId={{userId}}&username=xyx&phone=12345678900&credit=50

预期结果：状态码为200，返回的json对象中msg值为“success”

## 删除用户接口测试

接口：http://localhost:8080/user/delete?userId={{userId}}

预期结果：状态码为200，返回的json对象中msg值为“success”

## 查询校园巴士班次接口测试

1. 普通工作日顺时针

接口：http://localhost:8080/shift/search\_schedule?line\_name=LoopLineClockwise&type=NormalWorkday

预期结果：状态码为200，返回的json对象中schedule的scheduleList长度大于0

1. 寒暑假双休日逆时针

接口：http://localhost:8080/shift/search\_schedule?line\_name=LoopLineAntiClockwise&type=HolidayWeekend

预期结果：状态码为200，返回的json对象中schedule的scheduleList长度等于0

## 添加校园巴士班次接口测试

接口：http://localhost:8080/shift/add?lineName=LoopLineAntiClockwise&lineNameCn=校园巴士逆时针&lineType=NormalWorkday&departureTime=6:00:00&reserveSeat=0&comment=&busId=1&arriveTime=6:20:00

预期结果：状态码为200，返回的json对象中msg值为“success”

## 删除校园巴士班次接口测试

接口：http://localhost:8080/shift/delete?shiftId=LLAW0600

预期结果：状态码为200，返回的json对象中msg值为“success”

## 查找校车班次接口测试

接口：http://localhost:8080/shift/search\_schedule?line\_name=MinHangToXuHui&type=NormalWorkday

预期结果：状态码为200，返回的json对象中schedule的scheduleList长度等于22

## 查找预约信息接口测试

接口：http://localhost:8080/appointment/search?lineNameCn=闵行到徐汇&lineType=NormalWorkday&departureTime=17:00:00&appointDate=2018-07-15

预期结果：状态码为200，返回的json对象中appointmentList长度等于7

## 添加校车班次接口测试

接口：http://localhost:8080/shift/add?lineName=MinHangToQiBao&lineNameCn=闵行到七宝&lineType=NormalWorkday&departureTime=6:00:00&reserveSeat=35&comment=&busId=1&arriveTime=7:00:00

预期结果：状态码为200，返回的json对象中msg值为“success”

## 修改班车信息接口测试

接口：http://localhost:8080/shift/modify?shiftId=MQWD0600&reserveSeat=40

预期结果：状态码为200，返回的json对象中msg值为“success”

## 删除班车接口测试

接口：http://localhost:8080/shift/delete?shiftId=MQWD0600

预期结果： 状态码为200，返回的json对象中msg值为“success”