

# 《sharing-bicycle-共享单车》 软件设计文档（描述）

宋羽珩 黎文彬

# 目录

1 引言 .....	2
1.1 编写目的.....	2
1.2 背景.....	2
1.3 定义.....	2
1.4 参考资料.....	2
2 总体设计 .....	3
2.1 需求规定.....	3
2.2 运行环境.....	4
2.3 基本设计概念和处理流程.....	4
2.4 结构.....	5
2.5 功能需求与程序的关系.....	5
2.6 人工处理过程.....	6
2.7 尚未解决的问题.....	6
3 接口设计 .....	6
3.1 用户接口.....	6
3.2 外部接口.....	6
3.3 内部接口.....	7
4 运行设计 .....	7
4.1 运行模块组合.....	7
4.2 运行控制.....	7
4.3 运行时间.....	7
5 系统数据结构设计.....	8
5.1 逻辑结构设计要点.....	8
5.2 物理结构设计要点.....	9
5.3 数据结构与程序的关系.....	9
6 系统出错处理设计.....	9
6.1 出错信息.....	9
6.2 补救措施.....	9
6.3 系统维护设计.....	9
7 类设计 .....	10
7.1 超类设计.....	10
7.2 其他类设计.....	10
8 架构设计 .....	12
8.1 架构设计说明.....	12
8.2 服务器架构说明.....	12

# 概要设计说明书

## 1 引言

### 1.1 编写目的

本概要设计说明书建立在《sharing-bicycle-共享单车》需求分析的基础之上，对《sharing-bicycle-共享单车》进行概要设计。此软件概要设计说明书，在软件项目的开发过程中起到引导作用，保证项目小组按时保质地完成项目目标，便于项目团队成员更好地了解项目情况，使项目的实际开发过程合理有序。因此以文档化的形式，记录下软件的总体设计、接口设计、运行设计、数据结构设计、系统出错处理设计，作为开发过程中项目团队成员之间的共识与约定，以及项目团队开展和检查项目工作的依据。

本概要设计说明书面向所有项目开发人员以及用户，为系统后续开发提供依据。

### 1.2 背景

任务提出者：宋羽珩 黎文彬

项目开发人员：宋羽珩 黎文彬

用户：所有使用单车出行的市民

软件的计算中心：本软件的最终形式为运行于移动 Android 平台的 APP，软件服务器端由腾讯云提供支持。

### 1.3 定义

列出本文件中用到的专门术语的定义和外文首字母组词的原词组。

**Android:** 是一种基于 Linux 的自由及开放源代码的操作系统，主要使用于移动设备。

**MyEclipse:** 是在 eclipse 基础上加上自己的插件开发而成的功能强大的企业级集成开发环境，主要用于 Java、Java EE 以及移动应用的开发。

**C/S 架构:** 软件的一种体系结构，即客户端服务器结构。

**MD5:** 为计算机安全领域广泛使用的一种散列函数，用以提供消息的完整性保护。

**JDBC:** 是一种用于执行 SQL 语句的 Java API，可以为多种关系数据库提供统一访问，它由一组用 Java 语言编写的类和接口组成。

**SQL:** 是一种特殊目的的编程语言，是一种数据库查询和程序设计语言，用于存取数据以及查询、更新和管理关系数据库系统。

### 1.4 参考资料

## 2 总体设计

### 2.1 需求规定

名称	输入	输出	性能要求
登录	账号(手机号)和密码	若匹配正确, 则进入主页, 若匹配失败, 则弹出提示信息	在网络状况良好情下, 响应时间不超过 2 秒
注册	账号(手机号)和密码(不少于 6 位, 数字和字母混合)、确认密码	弹出注册是否成功信息	在网络状况良好的情况下, 响应时间不超过 2 秒
预约用车	点击想要预约的车辆, 然后点击预约按钮	向客户端返回预约成功信息	在网络状况良好的情况下, 响应时间不超过 2 秒
查看行程	点击个人信息页面, 点击行程按钮	显示当前用户的行程列表	在网络状况良好情下, 响应时间不超过 3 秒
远程开锁	用户点击车辆, 点击我要用车	车辆完成解锁, 返回骑行页面	在网络状况良好情下, 响应时间不超过 2 秒
故障申报	用户点击反馈, 选择故障类型, 填写备注信息, 点击申报按钮	申报成功, 返回程序的主界面	在网络状况良好情下, 响应时间不超过 2 秒
显示位置信息	用户 GPS 数据(经纬度 double)	客户端显示出所在位置, 并显示出车辆位置	在网络状况良好情下, 响应时间不超过 5 秒
数据分析	用户点击数据分析	返回一周的骑行情况(图表)	在网络状况良好情下, 响应时间不超过 3 秒
导航	用户输入目的地(字符串), 点击导航功能	请求处理完成, 开始导航	响应时间由第三方平台约束, 正常情况下, 开始导航时间不超过 3 秒
使用帮助	用户在点击帮助	返回帮助信息	响应时间不超过 1 秒
查看钱包	用户点击个人信息按钮, 点击余额	查询成功, 返回余额详单	在网络状况良好情下, 响应时间不超过 2 秒
积分功能	用户点击积分图标	返回积分详单	在网络状况良好情下, 响应时间不超过 2 秒

## 2.2 运行环境

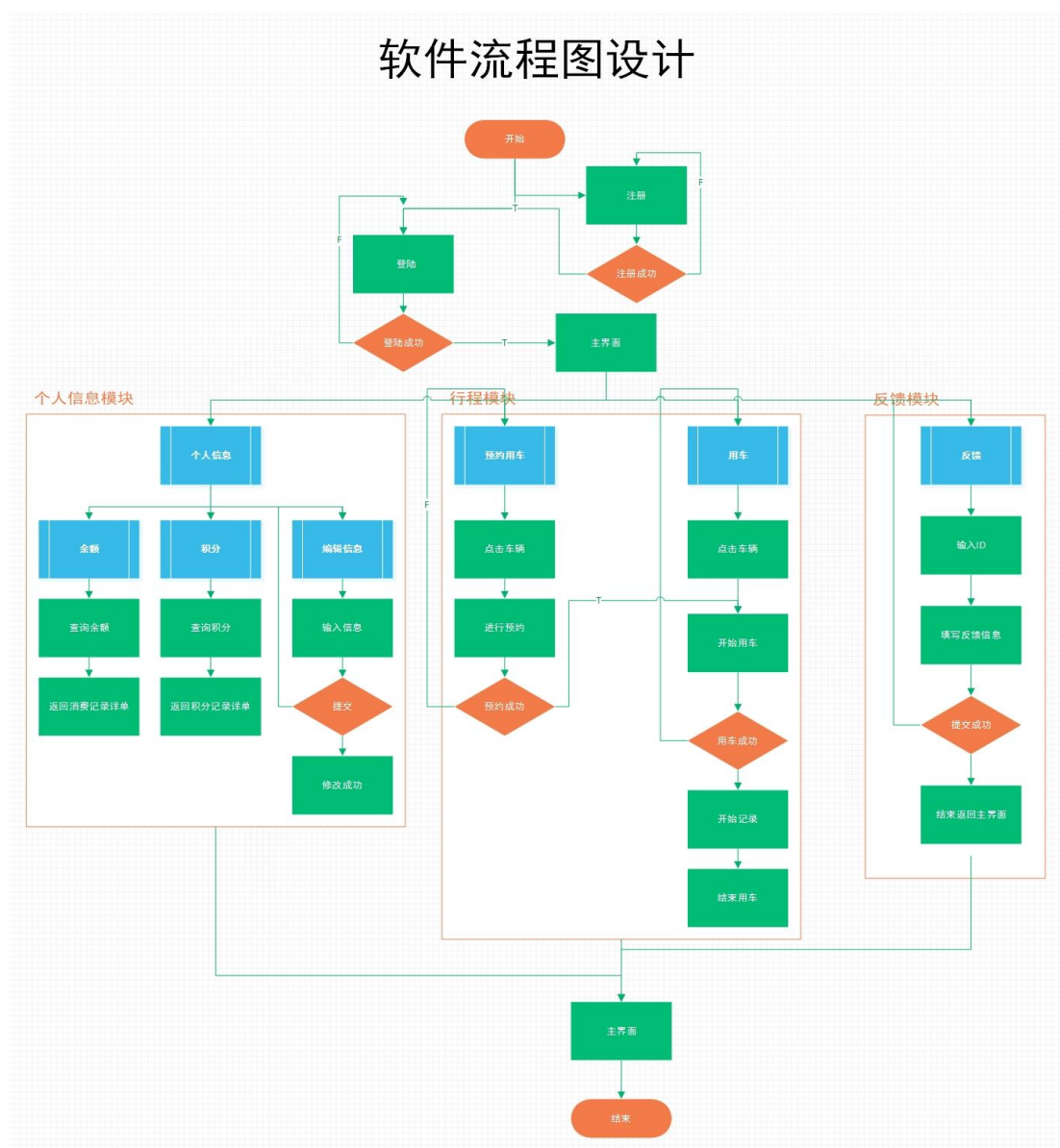
**APP 运行环境：**Android 5.0 系统以上，运行内存 2G 以上

**APP 服务器：**腾讯云，单核，1M 带宽，1G 内存

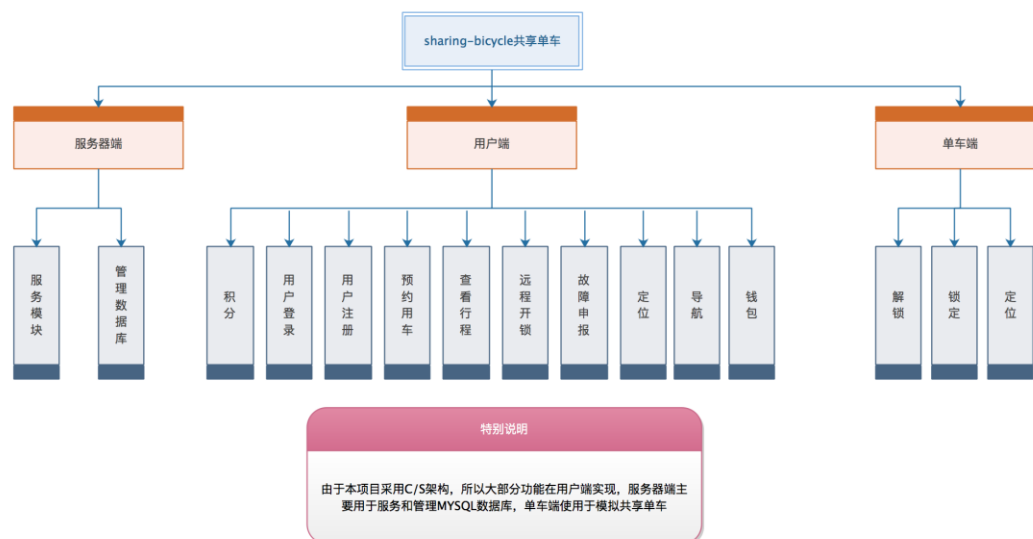
**服务器数据库：**MySQL

**支持软件：**tomcat，Android studio, Myecclipse

## 2.3 基本设计概念和处理流程



## 2.4 结构



## 2.5 功能需求与程序的关系

本条用一张如下的矩阵图说明各项功能需求的实现同各块程序的分配关系：

	登 录 程 序	注 册 程 序	预 约 程 序	查 询 行 程	查 询 积 分	开 锁 程 序	反 馈 程 序	定 位 程 序	导 航 程 序	查 询 钱 包	登 录 服 务	注 册 服 务	预 约 服 务	开 锁 服 务	查 询 服 务	反 馈 服 务
用户 登录	√										√					
用户 注册		√										√				
预约 用车			√										√			
查看 行程				√											√	
远程 开锁						√								√		
故障 申报							√									√
定位								√								
导航									√							
钱包										√					√	

积分					√											
解锁														√		
锁定														√		
定位								√								

## 2.6 人工处理过程

本系统在后期维护的时候，会对服务器数据库表进行定期维护，删除没有实际意义的数  
据，保证良好的数据存储，减小数据库压力，避免造成数据冗余。

除上述情况，本系统不需要任何人工处理过程。

## 2.7 尚未问决的问题

暂无

# 3 接口设计

## 3.1 用户接口

说明将向用户提供的命令和它们的语法结构，以及软件的回答信息。

- 登录界面：由手机号输入框，密码输入框，登录按钮构成
- 注册界面：由手机号输入框，密码输入框，确认密码输入框，注册按钮组成
- 主界面：由用车，反馈，预定，“我的”四个点击按钮组成

用车按钮：在点击当前车辆时，用户在选择车辆的情况下，可以点击用车，使用  
车辆

反馈按钮：显示反馈界面

预定按钮：在点击当前车辆时，用户可以进行预定

“我的”按钮：当点击“我的”可以返回个人信息界面

- 个人信息界面：由余额按钮，积分按钮，和个人信息组成。

个人行程：返回用户骑行记录

积分按钮：点击可以查看详细的积分记录

余额按钮：点击可以查看自己的消费记录，以及余额信息

注：所有界面设计，以及在用户界面设计文档中给出。

## 3.2 外部接口

本系统无外部接口

## 3.3 内部接口

本系统主要基于后端数据库的系统，所以本系统的主要接口为与 MYSQL 数据库进行数据库连接，采用 JSP 中的 JDBC 进行数据库连接，连接驱动由 MYSQL 提供。

# 4 运行设计

## 4.1 运行模块组合

本系统以 APP 的形式呈现，由主界面和各个窗口界面组成，任何运行模块的组合都必须包括登录模块，注册模块，主界面。可以通过主界面调用各个窗口界面，实现模块的组合，各个窗口界面的相互独立。除身份验证信息（即用户 ID 外），一般各个模块不需要进行数据的传递。每个界面实现一个特定的功能，属于交互采用，请求服务器的方式，进行数据的获取。

## 4.2 运行控制

运行控制采用模块间的调用函数来实现。

在客户端与服务器进行交互时，即网络传输方面，客户机向服务器发送请求，服务器接收到数据后，与数据库进行数据交互，进行增删改查操作，完成后对结果进行封装，返回客户端，客户端接收数据后，对数据进行处理，然后返回界面，呈现数据。

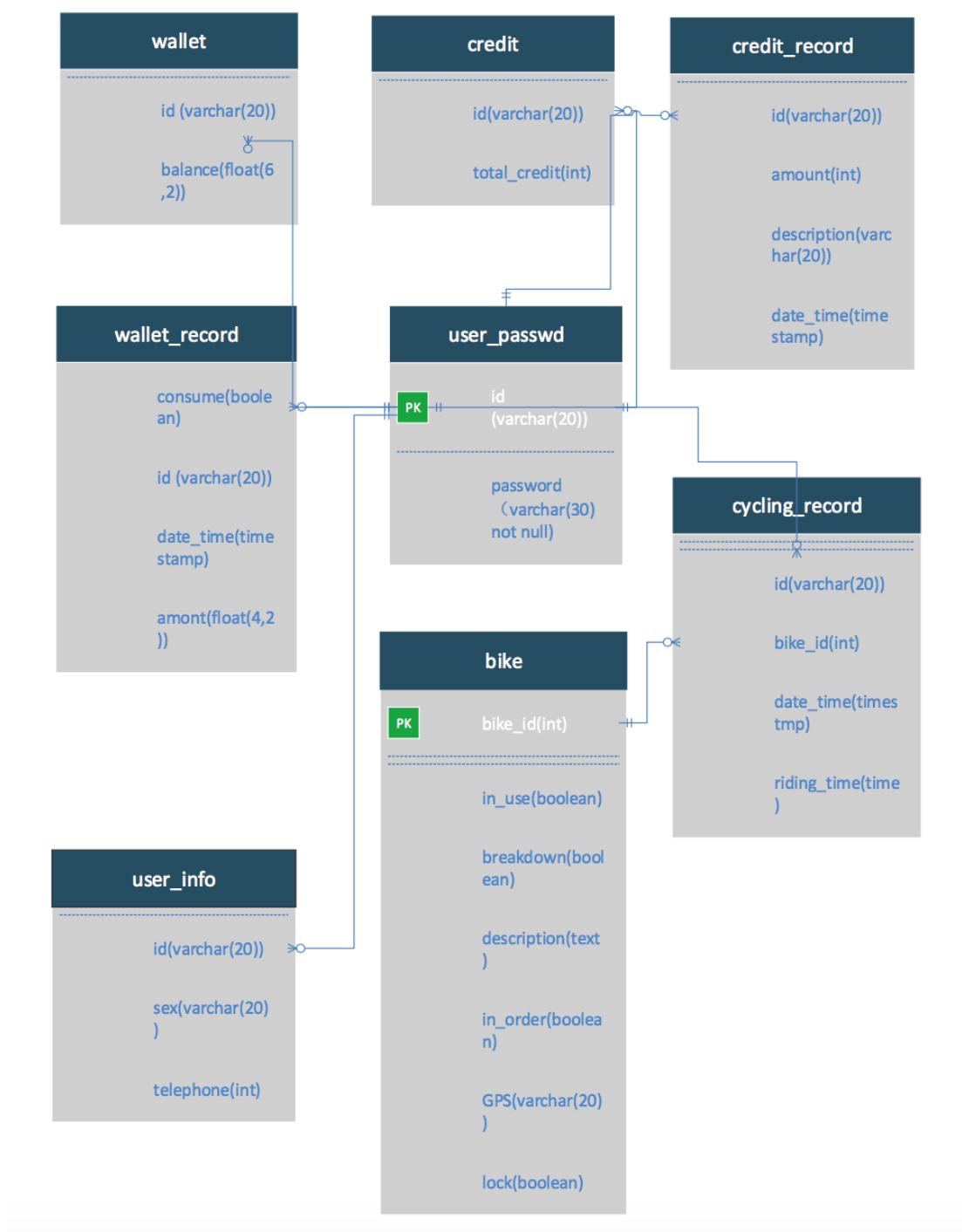
## 4.3 运行时间

每种模块组合的时间不能确定，这跟用户的操作，网路情况，以及实际操作的数据量的大小有关。



## 5 系统数据结构设计

### 5.1 逻辑结构设计要点



## 5.2 物理结构设计要点

用户密码数据项采用 MD5 加密的方式，存储于数据基表中，除密码数据项外，其他数据项以明文的方式存储于数据基表中。

数据的访问方法由 JDBC 连接提供访问方法。

数据库的存取单位根据数据项的数据类型进行确定。

## 5.3 数据结构与程序的关系

数据库基表的数据结构在进行定义基表的时候，进行数据项的定义。程序通过 JDBC 连接，采用 SQL 语句，调用 JDBC 程序执行的方式，对数据库数据结构以及数据项进行修改操作。

# 6 系统出错处理设计

## 6.1 出错信息

用一览表的方式说明每种可能的出错或故障情况出现时，系统输出信息的形式、含意及处理方法。

故障名称	输出形式	处理方式
服务器无法访问	服务器输出报错信息	对服务器进行重启
数据库连接失败	服务器输出报错信息	检查数据库服务是否启动
用户无法连接服务器	APP 提示连接失败	检查网络问题
用户输入不合法	APP 提示正确输入	用户操作无法进行,重新输入

## 6.2 补救措施

## 6.3 系统维护设计

系统目前没有设计维护专门的维护模块用于系统的维护，仅仅对服务器端数据库信息进行及时维护操作。

## 7 类设计

### 7.1 超类设计

- UserInfo 类
  1. 类属性: `account(String),isLogin(boolean)`
  2. 类方法: 无类方法
  3. 类描述: 用于表示用户信息的类。
- DAOfactory 类
  1. 类属性: `con(Connetion), datasource(DataSource)`
  2. 类方法: `getConnection()`
  3. 类描述: 用于从连接池中取出连接对象。

### 7.2 其他类设计

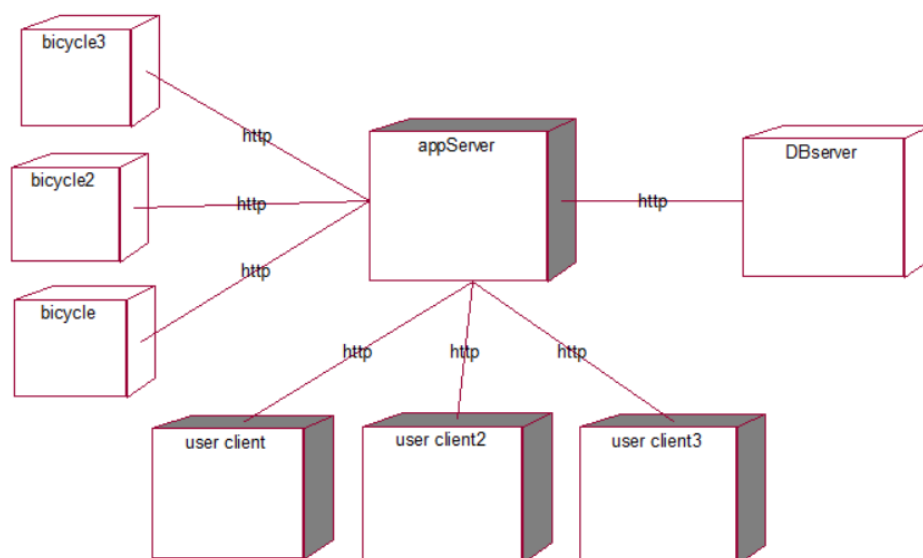
- LoginActivity 类
  1. 类属性:  
`account(String),password(String),inputAccount(EditText),inputPassword(EditText),loginButton(ImageButton),toRegist(ImageButton)`
  2. 类方法:  
`Public void startAnimationThreadStuff (long delay)`  
`Public void onResponse(String s)`  
`Public void onErrorResponse(VolleyError volleyError)`
  3. 类描述: 用于实现用户登陆
- AnalysisActivity 类
  1. 类属性: `record(ArrayList) analysisButton(ImageButton)`
  2. 类方法:  
`showChart(),getData(),analysis()`
  3. 类描述: 用于实现用户分析
- LocationActivity 类
  1. 类属性: `userLocation(ArrayList),bdmap(BDMap)`
  2. 类方法: `getLocation(),refresh()`
  3. 类描述: 用于实现用户定位
- NavigateActivity 类
  1. 类属性: `userLocation(ArrayList),bdmap(BDMap),navigate(ImageButton)`
  2. 类方法: `showRoute(),getDestination()`
  3. 类描述: 用于实现导航功能
- RegisterActivity 类
  1. 类属性: `account(String),password(String),confirmPassword(String)`
  2. 类方法: `Confirm(),register()`
  3. 类描述: 用于实现注册功能

- **HelpActivity 类**
  1. 类属性: troubleType(int),.solution(String)
  2. 类方法: getType(),getSolution()
  3. 类描述: 用户实现帮助功能
- **UnlockActivity 类**
  1. 类属性: bicycleStatus(boolean),location(ArrayList)
  2. 类方法: findBicycle(),unlock(),isPreorder(),sendData(),getData()
  3. 类描述: 用于解锁车辆
- **PreorderActivity 类**
  1. 类属性: bicycleID(String),countTime(int),userID(String)
  2. 类方法: getBicycleID(),getUserID(),sendData(),getData()
  3. 类描述: 用于实现车辆预定
- **ReportActivity 类**
  1. 类属性: reportText(String),UserID(String),Time(String),Tag(String)
  2. 类方法: sendReportData(),getReport()
  3. 类描述: 用于实现故障申报功能
- **UserWallet 类**
  1. 类属性: balance(int)
  2. 类方法: decreaseBalance(),increaseBalance(),updateBalance(),commWithDB()
  3. 类描述: 用于实现用户钱包功能
- **UserCredit 类**
  1. 类属性: credit(int)
  2. 类方法: decreaseCredit(),increaseCredit(),updateCredit(),communWithDB()
  3. 类描述: 用于实现用户积分功能
- **UserTrip 类**
  1. 类属性: trip(ArrayList)
  2. 类方法: getTrip(),commuWithDB(),showTrip()
  3. 类描述: 用于记录用车记录
- **UserDataDao 类**
  1. 类属性: account(int),id(String),password(String)
  2. 类方法: findUser(),addUser()
  3. 类描述: 用于与数据库相连，管理用户信息
- **BicycleDataDao 类**
  1. 类属性: bicycleID(String)
  2. 类方法: findBicycle(),addBicycle(),deleteBicycle()
  3. 类描述: 用于与数据库相连，管理车辆信息。
- **ReportDataDao 类**
  1. 类属性: bicycleID(String),description(String)
  2. 类方法: reportDescription()
  3. 类描述: 用于和数据库相连，管理故障信息

## 8 架构设计

### 8.1 架构设计说明

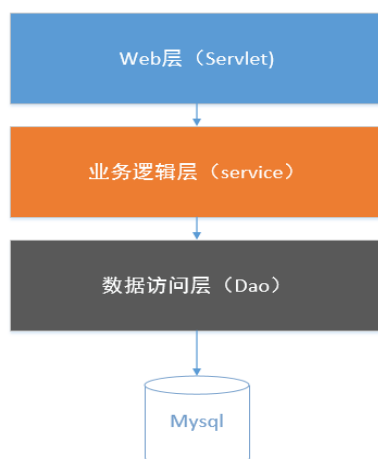
- 《sharing-bicycle》共享单车软件采用 C/S 架构，即客户端服务器架构模式，采用分布式系统。



系统架构设计部署图

### 8.2 服务器架构说明

- 服务器架构采用三层架构的方式，web 层，业务逻辑层，数据层。
- Web 层：主要对用户的请求接受，以及数据的返回，为客户端提供应用程序访问。
- 业务逻辑层：处理主要的业务逻辑。
- 数据层：进行对数据库的操作。



服务器架构图