

# 赛马赛事 Android 应用设计与开发

骆超,靳晟,张太红

(新疆农业大学 计算机与信息工程学院,新疆 乌鲁木齐 830052)

**摘要:**赛马运动作为一项人与动物共同完成的运动,具备独特的魅力,但由于赛马运动在国内没有得到大面积推广,赛马产业发展相对滞后,马匹主要用于肉食或骑乘,严重制约了马匹的经济效益。结合新疆大力发展马产业的契机,对通过信息技术提升新疆赛马产业的发展与升级进行研究;借助 Android 系统的方便携带与移动特性来开发移动应用,设计开发的功能如赛马赛事发布、赛事报名、票务管理、赛马分享以及赛马竞猜等,通过在传播赛马知识与信息的同时,让爱马人士以及普通民众参与到赛马运动中来,助推赛马运动的发展。通过对新疆本地赛马赛事的研究,以及结合与其他赛马赛事应用的对比,设计出合理及方便使用的赛马赛事移动应用,并加以推广。伴随着研究与开发的深入,赛马赛事信息化水平不断提高,通过移动信息技术将赛马运动与普通人群越拉越近,不断增强其认知度,为其发展增添不少动力。

**关键词:**Android;赛马;竞猜;直播;短视频

**中图分类号:**TP311.1

**文献标识码:**A

**文章编号:**1673-629X(2019)09-0169-06

**doi:**10.3969/j.issn.1673-629X.2019.09.032

## Design and Development of Android Application for Horse Racing

LUO Chao, JIN Sheng, ZHANG Tai-hong

(School of Computer and Information Engineering, Xinjiang Agricultural University,  
Urumqi 830052, China)

**Abstract:** Horse racing, as a sport jointly completed by human and animals, has its unique charm. However, due to the lack of wide promotion of horse racing in China, the development of horse racing industry lags behind relatively. Horses are mainly used for meat or riding, which seriously restricts the economic benefits of horses. Combining with the opportunity of vigorously developing horse industry in Xinjiang, the promotion of the development and upgrading of horse racing industry in Xinjiang through information technology is studied. According to the portable and mobile features of Android system, the mobile applications are developed, with functions such as horse race event release, race registration, ticket management, horse race sharing and horse race guessing. By disseminating horse-racing knowledge and information, horse lovers and ordinary people can participate in horse-racing to promote its development. Through the study of local horse racing in Xinjiang and the comparison with other horse racing applications, a reasonable and convenient mobile application of horse racing is designed and promoted. With the deepening of research and development, the level of information technology in horse racing is constantly improving. By bringing horse racing closer to the general population through mobile information technology, it has increased its awareness and added a lot of impetus to its development.

**Key words:** Android; horse racing; guess; live; short video

## 0 引言

随着国民经济的提高,人们对精神食粮的追求也越来越高,娱乐的方式也变得多种多样。骑马这种有趣的娱乐项目,也渐渐进入了大家的眼帘<sup>[1]</sup>。但仅仅是将马用于骑乘游玩,很难带动马产业的发展,因为它的经济效益是有限的。通过借鉴其他发达国家或地区

的经验,可以观察到,赛马运动的发展是带动马产业发展的关键<sup>[2-3]</sup>。发达的赛马赛事产业不仅能够带动马产业的发展,同时能够带来巨大的经济效益。新疆作为国内传统的马产业大区,马匹数量居全国首位,而且正值新疆着力打造马产业,促使新疆的马产业由传统的繁育、骑乘、肉食领域向文化旅游、竞技赛事等领域

收稿日期:2018-11-26

修回日期:2019-03-28

网络出版时间:2019-04-18

基金项目:新疆维吾尔自治区重大科技专项(2017A01002-5)

作者简介:骆超(1994-),男,硕士在读,研究方向为农业信息化;靳晟,硕士,副教授,通讯作者,研究方向为计算机科学与技术;张太红,博士,教授,研究方向为数据库技术。

网络出版地址: <http://kns.cnki.net/kcms/detail/61.1450.TP.20190318.1434.004.html>

迈进<sup>[4-5]</sup>。总体来说,新疆的马产业具备较为完整的产业链,发展赛马产业具备良好的基础。借此机遇与基础,探索如何通过信息化手段对新疆的赛马赛事的发展提供帮助,并提升整个新疆赛马赛事的信息化水平,是文中研究的主要目的。针对目前移动网络用户数量的不断攀升,将重点放在对于手机信息化应用的开发上,研究开发具备什么样功能的应用能使信息化的作用发挥更大<sup>[6]</sup>。通过观察发现,应用市场上已有的赛马赛事应用主要分两类:一类是主要用于马彩的应用,第二类是关注于赛事信息发布的应用。就赛马赛事在中国的发展状况,现有的应用模式借鉴性不大。

对于处于发展初期的赛马赛事,首要任务就是吸引人们的关注,设计的应用就需要对用户有吸引的作用,只有吸引了用户,才能更好地传授赛马知识或信息给用户。关于怎样吸引更多人关注,通过对时下最流行的朋友圈、短视频及直播等信息技术化手段的借鉴,研究如何将这些最新的信息技术糅合到应用中,从而更好地将赛马赛事传达到用户方<sup>[7]</sup>。之后再探索通过如赛马竞猜及马特产商城等功能来巩固用户群。

## 1 总体设计

### 1.1 功能设计

通过分析现有赛马赛事的发展,总结规律,得出功能性需求并列功能结构,如图 1 所示。

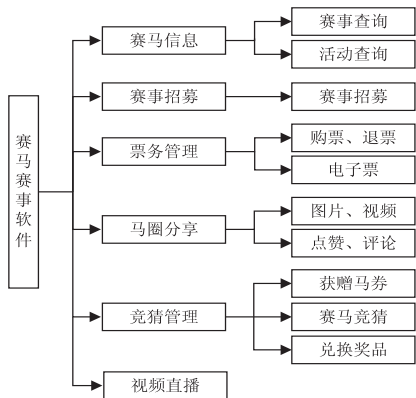


图 1 功能结构

由图 1 可以看出,这款赛马赛事应用所具备的功能涵盖很多内容,如赛马文化传播、互动、购票以及竞猜等等。这种设计能够迎合赛马赛事发展的初期,借助最新的娱乐手段来传播赛马文化,通过应用的装载,可以通过赛事信息发布传播最新的赛马信息,给用户传播正确的赛马文化;通过竞猜模块吸引用户参与到赛马比赛中来,并且还可以出售马产品;而票务与赛事招募,对于赛事组织、管理模块都会有很好的帮助。

### 1.2 软件架构设计

通过功能结构设计,可以发现整个应用的功能比较多,而且很多功能都不能仅仅依托于本地来实现。

以登陆、注册功能为例,如果注册的时候将用户的信息存储在本地,那在同一台设备上登陆是没有问题的,但是当换一台设备以后,那就无法登陆了。因为这个用户的信息不在这台手机上,所以无法登陆。几乎应用的所有功能都会涉及这个问题,将信息保存到本地,换台手机便无法获取相同内容。因此需要将数据存储在云端数据库,这样只要有网络,不管在哪里都能获取到相同的信息<sup>[8]</sup>。解决了数据存储的问题,现在的问题就是数据管理以及数据交互了。现在需要一台服务器,这台机器上面运行着写好的程序,通过这个程序可以操作与管理数据库,并且按照某种逻辑来设计接口,并提供给手机应用使用。整个软件的架构就出来了,手机通过网络与云端服务器中部署的程序建立联系,通过云端程序来操作云端数据库(见图 2)。

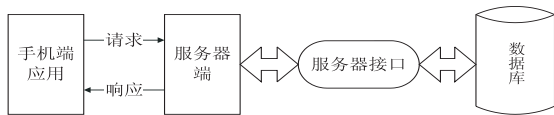


图 2 软件架构

### 1.3 技术选择

#### 1.3.1 应用开发

因为开发的是 Android 应用程序,所以首选谷歌公司推荐的编程语言进行开发,如 Java 或 Kotlin。相较于 Kotlin,Java 的受众更多,相关教程也会更多,所以开发语言选择 Java<sup>[9]</sup>。开发工具也用谷歌公司推荐的 Android Studio 工具。

#### 1.3.2 服务器端程序的开发

服务器端开发技术选择较多,但是大多数技术学习难度大,或者技术维护上做的不是特别好。通过参考资料也可以发现,Python 语言的使用量在不断增长,可见 Python 因为它的易用性以及实用性,越来越受到欢迎。所以根据以往经验和排名,选择基于 Python 的 Django 框架进行开发。

#### 1.3.3 数据库开发

对于数据库,主要结合免费、满足需求以及使用排名来进行选择,选择 MySQL 作为云端数据库。

## 2 关键技术

### 2.1 OkHttp

OkHttp 是一个处理网络请求的开源项目,是安卓端较为火热的轻量级框架,由移动支付 Square 公司贡献。通过这个框架,可以轻松完成 Http 的请求及响应。

### 2.2 Json

Json(JavaScript object notation)是一种轻量级的数据交换格式。简洁和清晰的层次结构使得 Json 成为理想的数据交换语言,其易于阅读和编写,同时也易

于机器解析和生成,并能有效提升网络传输效率。在网络传输中将主要传递 Json 格式的数据。

### 2.3 多线程、异步

多线程是指从软件或者硬件上实现多个线程并发(同步)执行的技术。具有多线程能力的计算机因有硬件支持而能够在同一时间执行多于一个线程,进而提升整体处理性能。

异步可以理解计算机多线程的异步处理。与同步处理相对,异步处理不用阻塞当前线程来等待处理完成,而是允许后续操作,直至其他线程将处理完成,并回调通知此线程。通过多线程和异步技术,可以提升应用的用户体验。

## 3 软件架构实现

### 3.1 数据库端的实现

在开始数据库实现之前需要做两方面工作,首先确定怎样创建数据库中的表,其次就是设计应用逻辑,将逻辑转换成数据库中的表。创建数据库有多种方式,传统方式通过 Sql 语句进行数据库的创建,或者通过可视化数据库管理工具(如 Navicat)进行可视化创建。如同创建 Excel 数据表一般,相较于通过 Sql 语句进行创建,通过工具创建会方便很多。因为服务器端的实现采用的是 Django 框架,而该框架提供了 ORM(对象关系映射)工具,通过 ORM 工具不仅可以轻松地创建数据库,还可以在不写一句 Sql 语句的情况下对数据库进行操作。

在真实创建表的时候,ORM 工具会把每个表用类的形式来表示。如下面的代码,创建了 CommonTimeInfo、EveryDaySign 这两个类,要实现的是每日打卡获取马券的功能,按规定打卡只能一天打一次。所以创建的表需要包含用户标识以及日期,其中 EveryDaySign 是用于存放这些信息的表,而 CommonTimeInfo 类通过在自身的 Meta 内部类中设置 abstract = True,就表明它是抽象基类,只能用于被继承,而不能被用于生成数据表。EveryDaySign 类继承了 CommonTimeInfo 类,同时也会拥有 CommonTimeInfo 类具有的属性<sup>[10]</sup>。

```
class CommonTimeInfo(models.Model):
    pub_date = models.DateTimeField(
        '发表时间',
        auto_now_add = True,
        editable = True)
    update_time = models.DateTimeField(
        '更新时间',
        auto_now = True,
        null = True)
```

```
    activated = models.BooleanField(
        default = 0,
        blank = False,
        verbose_name = "是否可用")
    class Meta:
        abstract = True
    class EveryDaySign(CommonTimeInfo, models.Model):
        user = models.ForeignKey(
            User,
            verbose_name = "关联用户")
        signday = models.DateField(
            auto_now_add = True,
            verbose_name = "打卡日期")
```

通过先后执行 makemigrations 以及 migrate 命令,就可以将 EveryDaySign 类转换为相应的数据表,并且继承 CommonTimeInfo 类的属性。在上面的代码中可以发现,EveryDaySign 中有一个与 User 类进行关联的外键,当生成真正的数据表时,也能够体现出来(见图 3)。

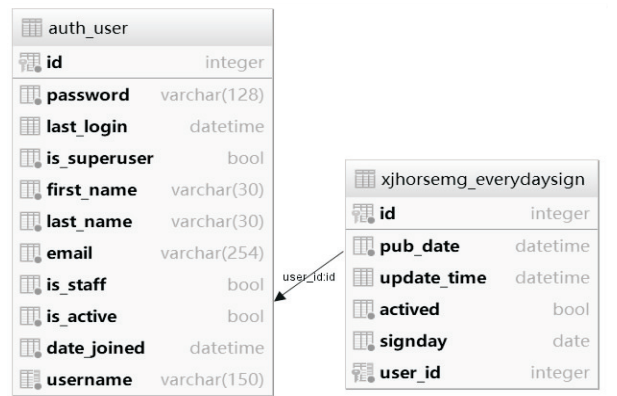


图 3 EveryDaySign 表结构

通过 ORM 工具,不用再纠结于创建 Sql 语句,而是可以专注于实现功能逻辑,只需要将逻辑转化成一个一个的类并且构建它们之间的关系。其他任务都可以就交给 ORM 工具处理。

### 3.2 服务器端的实现

构建应用的系统架构,不仅需要创建数据库和设计应用结构,在数据库与软件之间建立连接以及承担起整个系统之间的逻辑关系也是十分重要的。需要提供一个云端服务承担起这个角色。而应用与服务端之间建立联系是需要基于某种协议的。一般实时性要求较高的会采用 TCP 或者 UDP 协议,还有一些使用面相对广泛的协议,如 HTTP、FTP、SMTP。由于该系统实时性要求不高,而且 Web 服务基本上都是基于 HTTP 协议的,所以选用支持 Http 协议的 Django 框架来实现服务端程序的开发。

#### 3.2.1 请求与响应

应用与服务器之间的联系是基于请求与响应的,



Django 基于访问的路径来拦截软件发起的请求,然后调用相应的视图方法来处理相关操作,根据返回语句决定返回的是何种格式的数据(见图 4)。

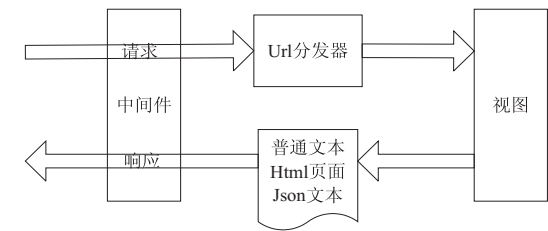


图 4 请求与响应过程

HTTP 协议常用于响应 Html 页面,而一般的手机应用不直接处理 Html 页面。软件需要接受能够表达完整意义以及易于传输、易于处理的格式化文本,如 Xml 和 Json。采用更受欢迎的 Json 语句来完成响应。

```
rs = {"status":0,"msg": "获取失败!" }
return JsonResponse(rs)
```

如上面的代码,Json 语句具备最基本的两个变量。应用可以判断每次请求的发送是否成功,但是并不能确定每次请求逻辑上是否正确,所以需要服务端将结果通过响应返回到软件,然后应用根据返回来执行相关提示和操作。表 1 便是服务端响应的不同消息,应用会根据响应的状态来进行判断。

表 1 响应码说明

响应码	响应消息	意义
1	操作成功	操作成功
0	操作失败	操作失败
-1	请先登录	需要登陆
-2	无权操作	权限不够

3.2.2 权限验证与装饰器

服务器端不仅要完成基本的逻辑构建,还要完成复杂的权限鉴定,如实现判断用户是否登陆的鉴定等。如果没有相关鉴定,其他用意不良的人,很容易便可以破坏后台的数据。所以必须要在关键操作中检查权限。但是肯定不能在每一次检验的时候,都将相同的检验代码写一遍,这样会出现很多重复的代码。因为 Django 是基于 Python 开发的,所以可以使用 Python 提供的装饰器功能来完成检验。就以登陆权限鉴定为例,Django 提供了登陆检验,但是当检测到用户没有登陆,会响应 Html 页面,希望返回 Json 文本,所以不符合要求。可以重写一个装饰器。如下面的代码,写好一个装饰器,可以反复使用,而且复用的方式非常简单,只需在视图函数上添加@符号和装饰器名称,就能实现对用户进行登陆鉴定。

```
def xjhorse_login_required(func):
def _wrapped_view(request, * args, ** kwargs):
rs = {"status": -1, "msg": "请先登陆账号!" }
```

```
if request.user.is_authenticated:
return func(request, * args, ** kwargs)
return JsonResponse(rs)
return _wrapped_view
@ xjhorse_login_required
def somerequest(request):
```

3.2.3 奖金分配表达式

结合现有的新疆本地赛马规则,根据不同的比赛以及单场参赛马数的不同,奖金分配规则也是不同的,所以不能直接把程序写死,必须在服务器后台根据不同的比赛进行填写。为了方便设置规则,根据具体需求,设计出了一套奖金分配表达式,具体如下:

6 = 50+30+20  
15 = 30+25+20+15+10

这两个表达式的意思是:当参赛数量超过 6 个时,前三名按照 50%、30% 及 20% 的比例进行分配;当参赛数量超过 15 个时,按照前五名 30%、25%、20%、15% 及 10% 的比例进行分配。

服务器端程序首先会通过回车与换行符,将所有表达式分割,然后通过正则表达式对所有的分配表达式进行比对,如下代码:

```
re.match(r'\d+=\d{1,2}([\+]{1}[\d]{1,2})+$',6=50+30+20`)
```

比对通过以后,再对表达式进行拆分,主要有两项检验,分别是分配比例的总和必须等于 100,以及分配比例必须是从大到小排列。如果全部检验通过,将会把表达式存放在数据库中,供下次计算奖金使用。

3.3 应用端的实现

3.3.1 Csrp 问题以及 Cookie 的保存

应用与服务端之间有许多交互,而这些交互有些是 Get 请求,有些是 Post 请求。当应用发起 Post 请求的时候,Django 会检验请求是否包含 CSRF(跨站请求伪造)密钥,如果没有,Django 将拒绝服务。CSRF 攻击是一种常见的网络攻击手段。当用户登陆了某个网站没有退出会话,现在去访问另一个网站,而这个用意不良的网站获取到了相关重要操作的链接,并修改了链接,诱导用户去点击这个链接,因为相关会话没有关闭,所以用户只要一点,很可能就会被攻击。CSRF 密钥是保存在 Cookie 中的,必须在执行 Post 请求的时候,将 CSRF 密钥添加到请求头里面或者请求参数里面,才能起到效果。

因此在服务端添加一个专门响应 CSRF 密钥的方法,当软件启动时,请求服务端获取密钥,然后将密钥存放到变量中。在进行 Post 请求时,添加 CSRF 密钥,响应便能成功(见图 5)。

现在 CSRF 的问题解决了,接下来就需要解决会话保存的问题了。服务器一般通过存储在 Cookie 里面的

Session 值来判断请求方的状态<sup>[11]</sup>。所以一定要保存好 Cookie 值,并做到及时更新,这样就能保存应用的会话状态了。

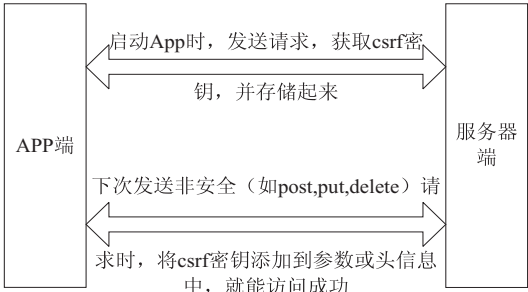


图5 CSRF 密钥处理过程

3.3.2 视频直播功能

直播是一个重要的手段,可以为养马或爱马人士提供一个很好的宣传窗口,普通用户可以通过这个窗口近距离地接触马匹,一举两得。搭建一个基本的直播功能必须具备三个基本的组成部分,那就是一台 RTMP(实时消息传输协议)媒体服务器、一个推流端以及一个拉流端。采用支持 RTMP 协议的 Srs 软件作为媒体服务器;而推流端分为电脑端和手机端,电脑端直接使用 Obs 软件进行推流,手机采用第三方推流应用实现,无需再开发应用;而拉流端,该软件直接集成百度拉流 Sdk,便可获取在线直播流。

3.3.3 打卡获取马券功能

为了不单单将该应用做成一个传播平台,而是将它做成用户粘性很高的具备娱乐功能的平台,在应用中集成了竞猜功能。为了完成奖励与惩罚,构建了一套数字货币的规则,就是马券,它不仅可以用于竞猜,还可以用于兑换礼品。用户可以通过打卡来获取马券,因此设计了打卡的功能。用户只要登录该应用,然后点击获取马券,后台会判断用户当天是否点击获取过马券,获取马券的数量也会随机分配给用户,当数据库中已经存在用户当天打卡记录,软件将提示用户已经获取过马券。

3.3.4 赛事竞猜功能

为了调动用户参与到赛马运动中以及传播赛马竞猜的规则,通过对现有马匹竞猜规则的研究,总结与提炼规则便构建了赛事竞猜功能。管理员在服务端添加可以进行竞猜的比赛项目,然后对项目添加参赛马匹和参赛骑手。普通用户可以通过应用选择要竞猜的项目,然后选择下注的马与骑手以及下注马券数目。当比赛结果公布时,如果用户猜中了,系统将返还用户下注的马券,并扣除所有输掉比赛的用户的马券作为奖励金,然后按照赢得竞猜的用户下注比例,将奖励金发放给获胜用户(见图6)。用户可以用马券不断进行竞猜,也可以在该应用的马特产商城中兑换礼品。马券

不能用于兑换现金,也不能用现金充值。

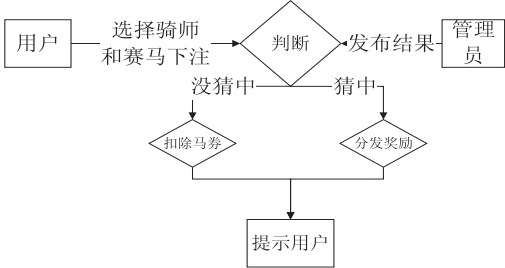


图6 竞猜流程

3.3.5 马圈分享功能

当应用具备了初步的娱乐功能,有一定的用户粘性后,最关键的就是怎样让用户接触赛马文化,真正喜欢上赛马。最简单有效的方式,就是通过时下最流行的方式,如短视频,所有用户都可在上面分享他们关于赛马的视频或者图片。将这个功能命名为马圈,用户可以发布视频和图片,并且可以配上文字说明,其他用户也可以进入马圈,获取别人发布的马圈,然后可以对别人的马圈进行点赞和评论。

3.3.6 解决内存溢出错误

应用中包含了大量的列表页,而每个列表项基本上都会关联一张照片,当通过大量子线程去下载图片的时候,因为有的图片特别大,而且图片数量也多,由于这些图片数据是存储在内存中的,而每个应用所分配的内存是有限的,便造成了内存溢出现象。为了解决这个问题,软件使用 Picasso 第三方开源框架,该框架可以转换图片以适应布局大小并减少内存占用,并且提供图片缓存功能,以节省用户流量<sup>[12]</sup>。因此使用该框架既避免了内存溢出的问题,又节省了用户的手机流量,同时也提高了列表的滑动速度。

3.3.7 版本适配

由于 Android 系统的版本更新较快,不同手机的系统版本跨度较大,为了让该应用在更多的手机上应用,需要做好版本适配,让所有功能都能在不同的 Android 系统中正确执行。主要适配要点如下:

- (1) Android5: 提供了新的摄像头接口 Camera2, 所以使用 Camera2 接口时,一定要对低版本的系统进行适配。
- (2) Android6: 新增了运行时权限确认。以往的 Android 版本要使用某项权限,只需要在 Android Manifest.xml 文件中添加相关权限就可以了,但是现在如果要调用某些权限(例如:通讯录、位置、相机),还需要在应用运行的时候动态地申请并被用户允许才能正常使用。
- (3) Android7: 禁止直接通过路径访问应用私有文件,如某个应用调用系统自带相机进行拍照,不进行处理是无法拍照的,因为系统相机应用无法访问该应用

的私有文件路径,需要通过 FileProvider 类给相机应用提供一个可以访问的路径才行。

通过 Build.VERSION.SDK\_INT 常量,可以获取当前系统的版本号,应用在需要适配的地方,根据不同的版本号执行不同的程序,就可以实现在不同的版本系统上正常运行<sup>[13]</sup>。

3.3.8 应用自更新

由于 Android 系统的开源性,不仅催生了大量的手机厂商使用 Android 系统,也同样造成了一个手机厂商一个应用商城,没有统一的应用商城。为了减少维护成本,应用升级离不开自动更新,只有具备自动更新功能,最新的功能才能更好地传达到用户,而且不用被不同的应用商场所束缚。

可以在服务器端创建一个表,结构如表 2 所示。

表 2 版本更新

字段名	字段类型	字段解释
version_code	float	版本号
version_name	varchar(30)	版本名
download_link	varchar(250)	下载链接
is_force_update	int(1)	是否强制更新

当应用启动时,去服务器获取最新版本信息,并将 AndroidManifest.xml 中配置的 android:versionCode 与最新版本进行比对,如果不一致,就提示更新,如果服务器要求强制更新,那用户只有更新后才能正常使用当前应用<sup>[14]</sup>。

4 系统运行结果展示

该应用实现的主要功能包括用户登录、赛事信息、伯乐识马、门票购买、赛事竞猜、礼品兑换以及马圈分享等。图 7 与图 8 给出了该款应用的部分截图。



图 7 马品种展示



图 8 竞猜项列表

5 结束语

在智能手机飞速发展的今天,移动设备的便携性使其可以在某些方面代替传统的桌上型计算机,给用户带来了极大便利,并且移动设备的用户量在不断攀升,用户量也是开发移动应用的重要考虑因素。因为在赛马运动的发展过程中,离不开越来越多人的关注,只有借助用户量多的平台才能吸引更多的人去关注。通过应用的开发可以将最新的赛马赛事信息传递给用户,实现赛马竞猜等功能可以让用户尝试与体验怎么参与到赛马运动中来,不断结合时下流行的信息技术在做到传递赛马知识的同时,吸引更多的用户关注赛马赛事是该应用设计与开发的期望结果。

参考文献:

[1] 王薛红. 基于产业发展视角的我国赛马彩票发行可行性研究[J]. 财政科学,2016(11):73-84.

[2] 杨风华. 赛马文化对赛马产业影响研究[J]. 武汉商业服务学院学报,2010,24(5):8-11.

[3] 秦尊文,路洪卫. 世界商业赛马现状与中国开放赛马彩票前景[J]. 江汉论坛,2007(3):56-58.

[4] 萧健. 香港赛马文化中的博弈[D]. 金华:浙江师范大学,2014.

[5] STRUNK R C, VERNON K L, BRIDGES W C, et al. Effects of rider experience level on horse kinematics and behavior [J]. Journal of Equine Veterinary Science,2017,68(3):52-64.

[6] 熊伟平,余刚. 信息技术在赛马赛事管理中的应用[J]. 当代体育科技,2017,7(31):165-166.

[7] 马强,王阳. 体育信息化与计算机应用学科发展综述[J]. 电子测试,2014(22):50-52.

[8] 耿俊豹,梁乾,魏曙寰,等. 综合 C/S 和 B/S 模式的船舶信息管理系统[J]. 计算机工程,2005,31(13):215-216.

[9] 宋小倩,周东升. 基于 Android 平台的应用开发研究[J]. 软件导刊,2011,10(2):104-106.

[10] TSOUKALOS M. Using Django and MongoDB to build a blog[J]. Linux Journal,2014,238:3.

[11] YUE Chuan, XIE Mengjun, WANG Haining. An automatic HTTP cookie management system[J]. Computer Networks, 2010,54(13):2182-2198.

[12] 刘娜娜. 基于 Android 的校园活动推荐系统的设计与实现[D]. 武汉:华中科技大学,2016.

[13] ALEPIS E, PATSAKIS C. Session fingerprinting in Android via Web-to-App intercommunication[J]. Security and Communication Networks,2018,79(6):1-13.

[14] 郭健侯,蔡雪松. 基于 Android 平台的公交信息发布终端软件的设计与实现[J]. 计算机应用与软件,2014,31(9):322-324.