# 软件设计文档



项目名称：电影购票系统

项目组长：田珂珂

成员：田满鑫、李绍焜、朱钦、李盛润

2016年7月12日

**修订历史**

|  |  |  |  |
| --- | --- | --- | --- |
| **日期** | **版本** | **描述** | **作者** |
| 2016-7-12 | 0.1 | 草稿 | 李绍焜 朱钦 |
| 2016-7-13 | 0.9 | 初稿 | 李绍焜 朱钦 |
| 2016-7-13 | 1.0 | 初稿 | 李绍焜 朱钦 |

## 目录

1. 开发规划

1.1开发人员

1.2开发环境和工具

1. 总体设计

2.1概念术语描述

2.2基本设计描述

2.3主要界面流程描述

2.4模块列表

1. 技术、思想、模型选型

3.1面向对象

3.2敏捷开发模型

1. 数据结构

4.1数据库表列表

4.2event列表

4.3location列表

1. 设计模式

5.1单例模式

5.2工厂方法模式

5.3观察者模式

5.4适配器模式

1. 接口规范

6.1<模块1 API>

6.1<模块1 API>

6.1<模块1 API>

6.1<模块1 API>

6.1<模块1 API>

6.1<模块1 API>

1. 功能设计 （未写完）

7.1功能1设计

7.1功能1设计

7.1功能1设计

7.1功能1设计

7.1功能1设计

1. 附录

8.1使用的开源代码说明

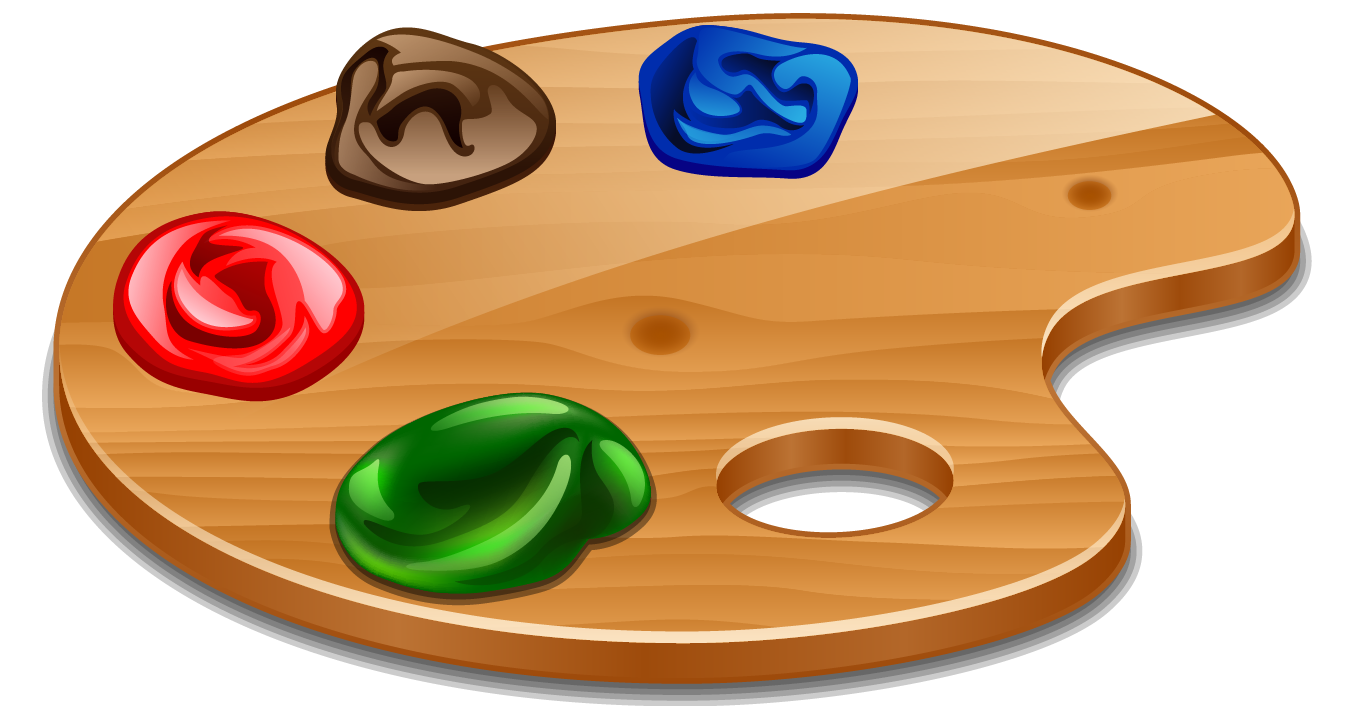
8.2使用的其他课程项目说明

8.3参考资料

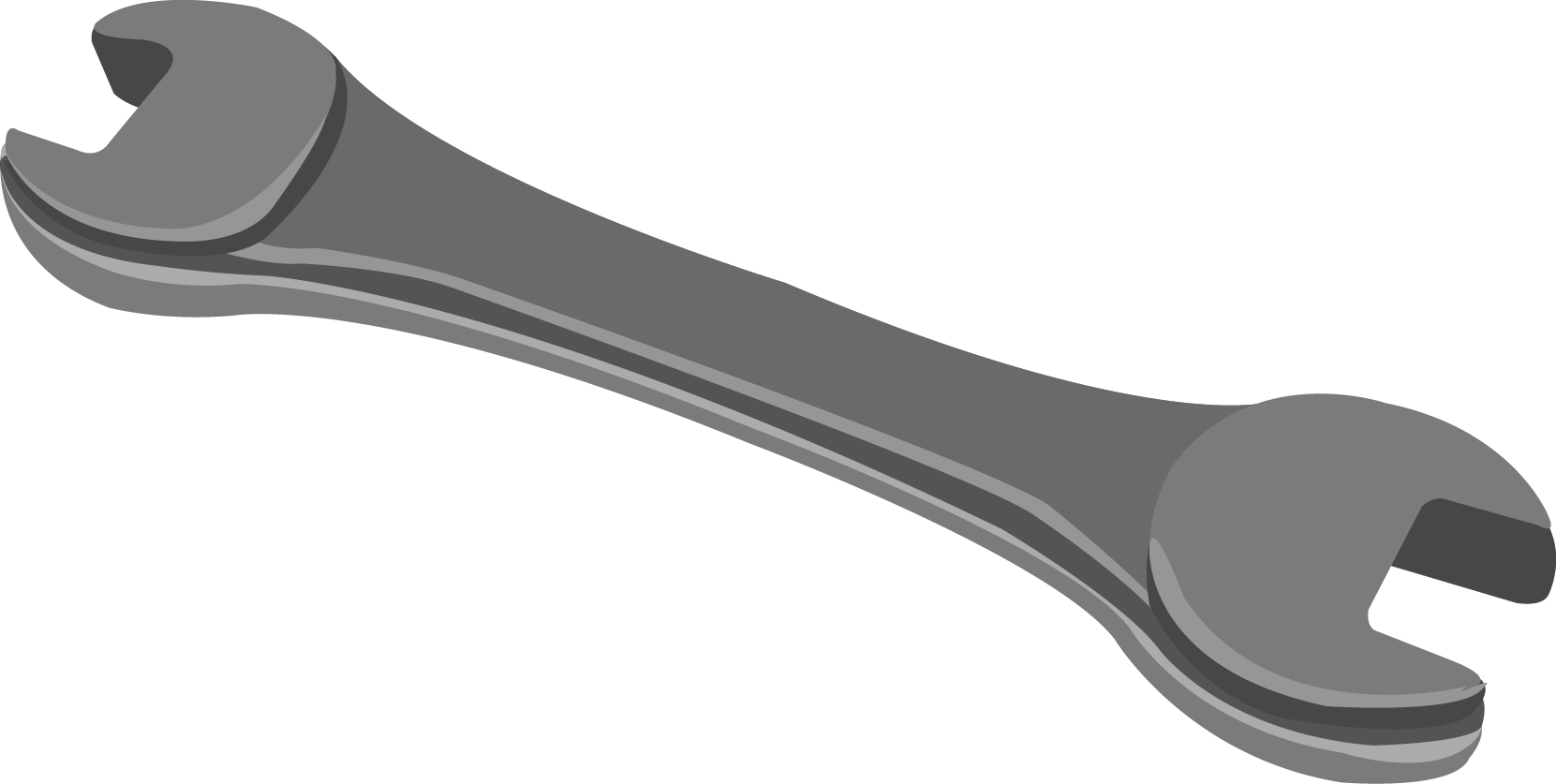
## 1.开发规划

|  |  |  |  |
| --- | --- | --- | --- |
| **1.1开发人员** | | | |
| **角色** | **主要职责** | **人员名称** | **备注** |
| **组长** | **需求分析**  **交互设计**  **架构设计**  **后台代码编写** | **田珂珂** |  |
| **成员** | **后台代码编写**  **软件测试**  **文档编写**  **文档整理** | **李绍焜** |  |
| **成员** | **前端代码编写**  **交互设计**  **架构设计**  **需求分析** | **田满鑫** |  |
| **成员** | **前端代码编写**  **UI设计**  **需求分析**  **文档编写** | **朱钦** |  |
| **成员** | **前端代码编写**  **软件测试**  **文档编写**  **文档整理** | **李盛润** |  |

**1.2开发环境和工具**

设计开发

|  |  |
| --- | --- |
| **工具** | **版本** |
| Mocking Bot | Version:v0.2.4.0 |
| Adobe Photoshop | CS6 |
| 2800 Icons for Developers | Version 2.0.10 (AppStore) |
| UMLet | Version 14.1.1 |

代码开发工具

|  |  |
| --- | --- |
| **工具** | **版本** |
| Android studio | Version 2.0 |
| Android Development Tookit | Version: 23.0.0.1245622 |
| Android SDK Tools | Rev.23 |
| Android SDK Platform-tools | Rev.20 |
| Android SDK Build-tools | Rev.20 |

## 2.总体设计

**2.1概念术语描述**

**术语表**

设计

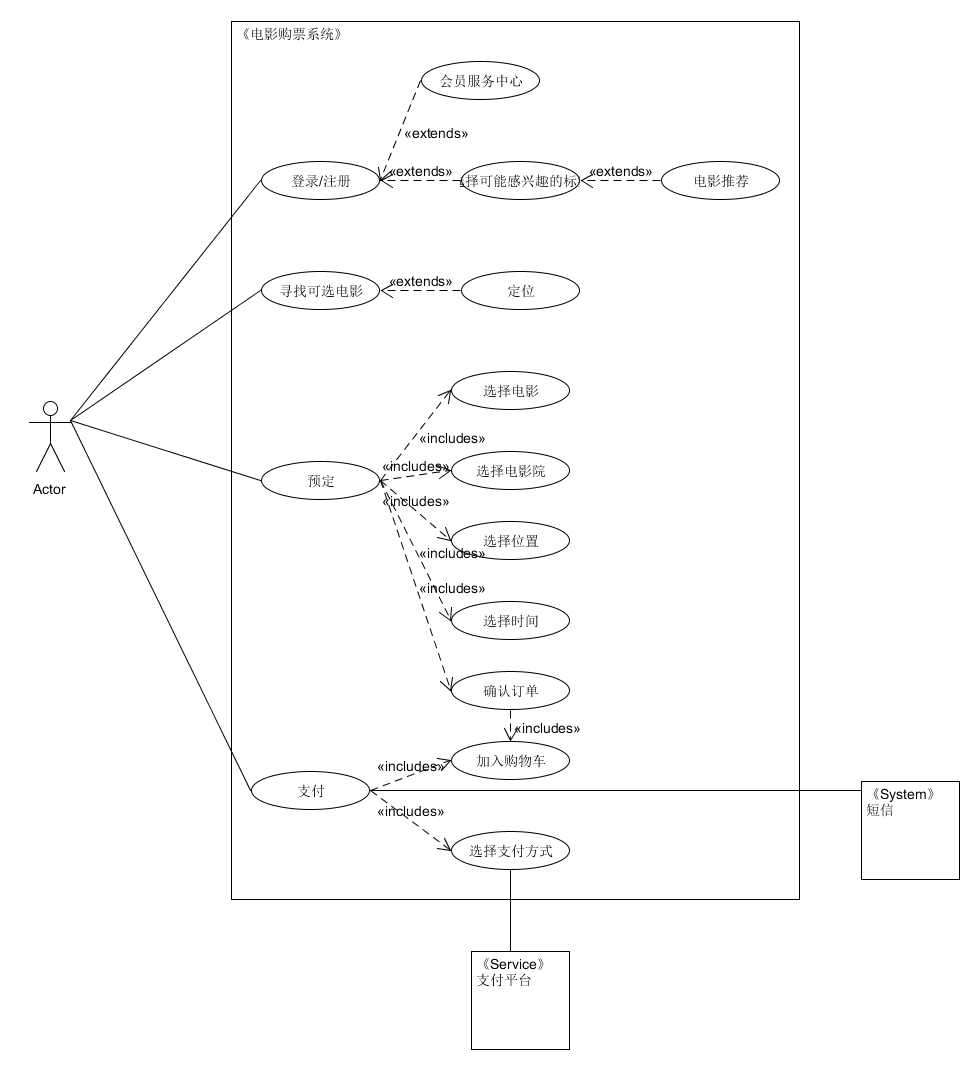
|  |  |  |
| --- | --- | --- |
| **术语** | **定义和信息** | **格式** |
| 电影介绍 | 展示一部电影的简介、宣传海报、观众评价等的信息 | 一个可以显示图片的区域和几个显示文本的区域 |
| 附近的电影院 | 按照第三方软件提供的定位信息，给出用户方圆几公里的电影院的信息 | 电影院名称-地址-距离-最低票价 |
| 电影推荐 | 根据用户购票记录，个性化推荐电影 | 电影名称-电影简介-电影评价 |
| 轮播图 | 自动循环切换显示在首页的图片，鼠标点击可以跳转入详细页 | 一个可以显示图片的矩形区域 |
| 列表项 | 展示一个电影的简要信息的载体 | 一个可以显示图片和文本的矩形区域 |

开发

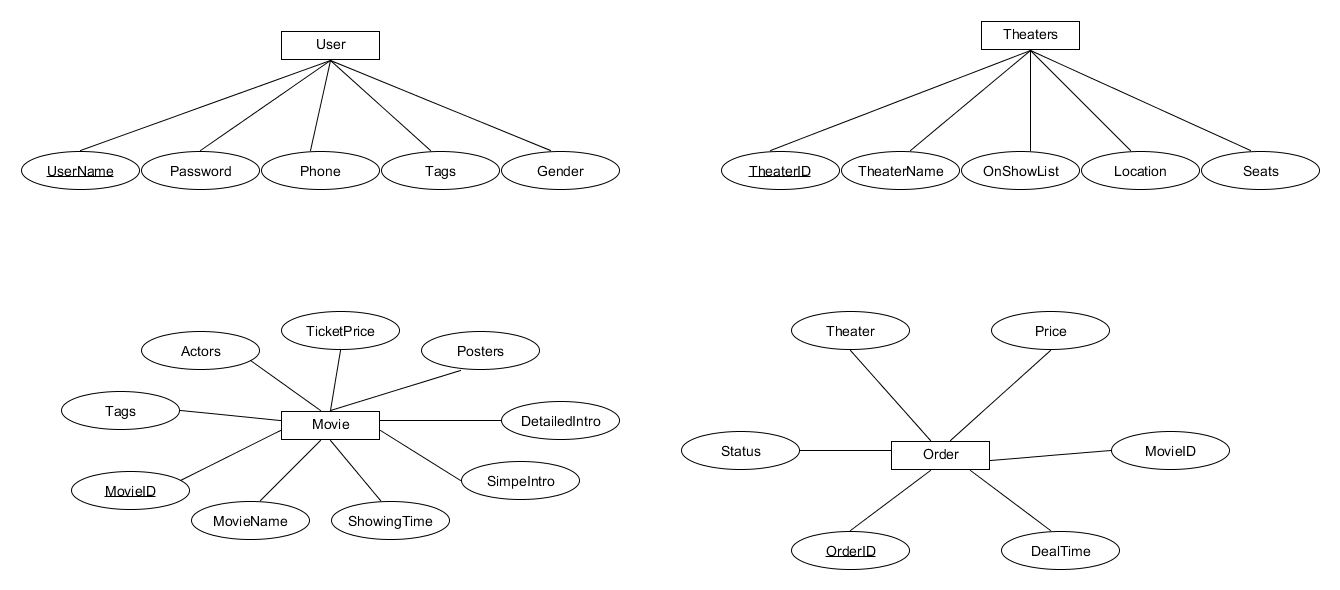
|  |  |  |
| --- | --- | --- |
| **术语** | **定义和信息** | **格式** |
| 请求 | 从客户端向服务器发送的请求信息 | Json |
| 数据库 | 记录用户账号、密码、历史行为等信息 |  |
| 事件 | 登录注册和购票的总称 |  |

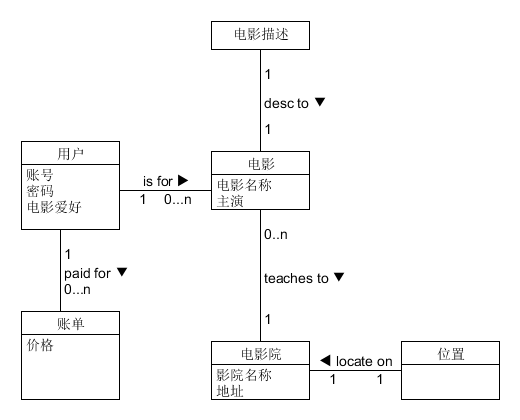
**2.2基本设计描述**

2.2.1用例图

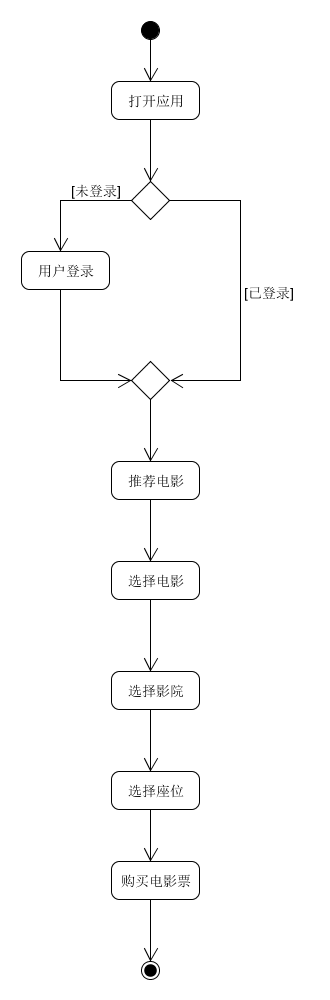


2.2.2E-R图

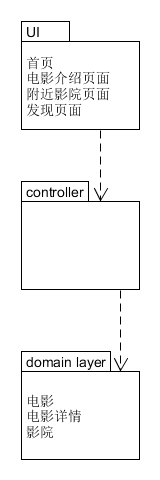


2.2.3领域模型

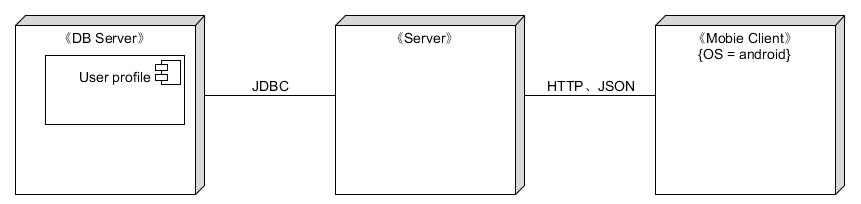
2.2.4流程图



2.2.5系统总体逻辑结构图（包图）



2.2.6系统部署图



**2.3主要界面流程描述**

|  |  |
| --- | --- |
| **功能编号** | **功能描述** |
| 功能1 | 登录、注册 |
| 功能2 | 定位用户位置 |
| 功能3 | 查看电影信息 |
| 功能4 | 查看影院信息 |
| 功能5 | 查看自己的信息 |
| 功能6 | 发表评论 |
| 功能7 | 购买电影票 |

2.3.1功能1界面流程

（1新用户可以点击“注册”，来注册新账号，注册完账号可以用账号登录

（2已有账号的用户可以输入账号及密码来登录

2.3.2功能2界面流程

（1登录之后，左上角会根据定位显示地理位置

（2用户也可以自行修改上面的位置信息

2.3.3功能3界面流程

（1系统会依据用户注册时填写的电影爱好及以往的购票记录来个性化推荐电影

（2点击任意一部电影Item也会进入“选中电影”页面，查看电影信息

2.3.4功能4界面流程

（1点击“影院”进入“影院列表”

（2点击任意一个影院Item，进入“选中影院”页面，查看影院信息

2.3.5功能5界面流程

（1点击“我的”，进入个人信息页面

（2选择“我的订单”，查看订单信息

2.3.6功能6界面流程

（1点击“待评价”，进入评价页面

（2在文本框输入评论，点击“发表评论”，发表个人对电影的评价

2.3.7功能7界面流程

（1找到合适的时间段所在的Item，点击“选座购”进入选座页面

（2选座心仪的座位后，“确认座位”进入“确认订单”页面（未完成）

（3确认订单，进入“支付”页面

2.4模块列表

|  |  |
| --- | --- |
| **模块功能** | **开发者** |
| DB(数据库) | 田珂珂 |
| UI设计 | 朱钦 |
| 前端 | 田满鑫、朱钦、李盛润 |
| 后台 | 田珂珂、李绍焜 |

## 3.技术、思想、模型选型

**3.1面向对象（OO）**

·概念：面向对象程序设计（Object-oriented design,OOD）是一种程序设计范型，同时也是一种程序开发的方法，对象指的是类的实例。它将对象作为程序的基本单元，将程序和数据封装其中，以提高软件的重用性、灵活性和扩展性。【来自维基百科：<https://en.wikipedia.org/wiki/Object-oriented_design】>

·选择理由：选择面向对象的程序设计不仅仅是因为安卓手机开发语言Java所要求的。在《人月神话》这本书里面，作者提出了关于软件设计与开发的“银弹问题”。虽然到目前为止，对于“银弹”的讨论还在继续。但是面向对象的程序设计是公认的提高软件设计效率，增加软件设计和开发的可行性的到目前为止最有效的方法。从大一下学期开始我们就开始学习面向对象的方法，而且在系统分析与设计这门课里面，也讲了关于面向对象的设计，所以，我们有足够的理由选择它。

·实际应用：我们在实际设计和编写的时候，用到了很多面向对象的思想和方法。主要包括以下的几个部分：类，对象，消息传递，继承，封装性，多态和抽象性。

**3.2敏捷开发 （Scrum）**

·概念：敏捷开发以用户的需求进化为核心，采用迭代、循序渐进的方法进行软件开发。在敏捷开发中，软件项目在构建初期被切分成多个子项目，各个子项目的成功都经过测试，具备可视、可集成和可运行使用的特征。换言之，就是把一个大项目分为多个相互联系，但也可独立运行的小项目，并分别完成，在此过程中软件一直处于可使用状态。

·选择理由：敏捷开发更强调程序员团队之间的紧密协作、面对面的沟通、频繁地交付新的软件版本、紧凑而自我组织型的团队、能够很好地适应需求变化的代码编写和团队组织方法。这正是我们所需要的。因为所有的需求都需要我们自己来思考，也没有业务专家来跟我们沟通，所以我们所做的需求分析可能会发生很多次变化，如果使用如瀑布模型这种，后果可能是很严重的。同时，敏捷开发更适用于较小的队伍，它的核心就是迭代、循序渐进，对于我们这样一个5人的小队伍，敏捷开发可能是效率最高的模型了。

·实际应用：我们作为一个整体工作，按短迭代周期工作，每次迭代交付一些成果，关注需求优先级，检查以及调整。

## 数据结构

**4.1数据库表列表**

|  |  |
| --- | --- |
| **表名** | **作用** |
| User | 存储用户账户、密码、个人信息和历史信息等 |
| Movie | 存储电影的简介、评价等信息 |
| Theaters | 存储电影院的位置、上映电影等信息 |
| Order | 存储用户订单的记录信息 |

**4.2User列表**

|  |  |  |
| --- | --- | --- |
| **属性名** | **是否主键** | **作用** |
| UserName | √ | 识别每一个用户 |
| Password | × | 登录密码 |
| Phone | × | 密码找回，短信通知 |
| Tags | × | 系统根据标签生成个性化推荐 |
| Gender | × | 性别 |

**4.3Movie列表**

|  |  |  |
| --- | --- | --- |
| **属性名** | **是否主键** | **作用** |
| MovieID | √ | 区分每一场不同的电影 |
| MovieName | × | 电影名 |
| ShowingTime | × | 电影上映时间 |
| SimpleIntro | × | 电影简介 |
| DetailedIntro | × | 电影详细介绍 |
| Posters | × | 电影海报 |
| TicketPrice | × | 电影票价 |
| Actors | × | 电影主演 |
| Tags | × | 电影属于的标签 |

**4.4Theaters列表**

|  |  |  |
| --- | --- | --- |
| **属性名** | **是否主键** | **作用** |
| TheaterID | √ | 识别每一间影院 |
| TheaterName | × | 影院名称 |
| OnShowList | × | 正在上映的电影列表 |
| Location | × | 地址、位置 |
| Seats | × | 座位 |

**4.5 Order 列表**

|  |  |  |
| --- | --- | --- |
| **属性名** | **是否主键** | **作用** |
| Order | √ | 识别每一个订单 |
| DealTime | × | 交易时间 |
| MovieID | × | 购买的电影的ID |
| Price | × | 票价 |
| Theater | × | 影院 |
| Statur | × | 状态 |

## 5.设计模式

**5.1单例模式（Singleton Pattern）**

·概念：在应用这个模式时，单例对象的类必须保证只有一个实例存在。许多时候整个系统只需要拥有一个的全局对象，这样有利于我们协调系统整体的行为

·分析：在单例模式下，一个类能返回对象的一个引用（永远是同一个）和一个获得该实例的方法（必须是静态方法）。当调用这个方法的时候，如果类持有的引用不为空就返回这个引用；如果为空就创建该类的实例并将实例的引用赋予该类来保持

·具体应用：

·代码位置：

·代码截图：

**5.2工厂方法模式（Factory method Pattern）**

·概念：定义一个创建对象的接口，但让实现这个接口的类来决定实例化哪个类。工厂方法让类的实例化推迟到子类中进行。

·分析：当创建对象需要大量重复的代码的时候可以使用工厂方法模式。它通常包含一个或多个方法，用来创建这个工厂所能创建的各种类型的对象。这些方法可能接受参数，用来指定对象创建的方式，最后返回创建的对象。

·具体应用：

·代码位置：

·代码截图：

**5.3观察者模式（Observer Pattern）**

·概念：在此种模式中，一个目标对象管理所有相依于它的观察者对象，并且在它本身的状态改变时主动发出通知。这通常透过呼叫各观察者所提供的方法来实现。此种模式通常被用来实现事件处理系统。

·分析：观察者是安卓原生的，EditText中注册了一个TextWatcher当文本框有修改的时候，回调观察者。

·具体应用：

·代码位置：

·代码截图：

**5.3适配器模式（Adapter Pattern）**

·概念：将一个类的接口转接成用户所期待的。一个适配使得因接口不兼容而不能在一起工作的类工作在一起，做法是将类别自己的接口包裹在一个已存在的类中。

·分析：通过适配器，客户端可以调用同一接口，因而对客户端来说是透明的。这样做更简单、更直接、更紧凑。复用了现存的类，解决了现存类和复用环境要求不一致的问题。

·具体应用：

·代码位置：

·代码截图：

## 接口规范

## 7.功能设计

|  |  |
| --- | --- |
| 功能编号 | 功能描述 |
| 功能1 | 登录、注册 |
| 功能2 | 定位用户位置 |
| 功能3 | 查看电影信息 |
| 功能4 | 查看影院信息 |
| 功能5 | 查看自己的信息 |
| 功能6 | 发表评论 |
| 功能7 | 购买电影票 |

**7.1功能1设计**

7.1.1功能1设计描述

功能1主要是用户的注册和登录。新用户需要注册一个账号才能使用软件，老用户可以输入账号，密码登录使用。

7.1.2功能1界面描述

（1登录界面采用“EditView”(编辑框)

（2与常见的登录界面相似，一个框为输入账号，下面一个框为输入密码

**7.2功能2设计**

7.2.1

功能2是根据用户当前的位置，给出定位信息。如果用户有需要，可以手动更改上面的位置信息。

7.2.2功能2界面描述

（1点击左上角“位置”下拉菜单

（2选择城市

**7.3功能3设计**

7.3.1功能3设计描述

功能3是根据用户的位置，提供附近影院上映电影的列表，以及根据用户的观影爱好，个性化推荐电影。包括电影海报、电影名称、电影简介等的信息。

7.3.2功能3界面描述

（1点击“电影列表项”

（2进入电影详细介绍页

**7.4功能4设计**

7.4.1功能4设计描述

功能4是根据定位，给出附近电影院的列表。包括电影院的名称、地址、与用户的距离和票价等的信息

7.4.2功能4界面描述

（1点击底部栏的“影院”图标

（2进入“影院列表”页

（3点击影院列表项，进入“影院详细信息”页

**7.5功能5设计**

7.5.1功能5设计描述

功能5是查看个人的信息，包括是否为会员、订单记录、还可以进行设置。

7.5.2功能5界面描述

（1点击底部栏的“我的”图标

（2进入“个人信息”页

**7.6功能6设计**

7.6.1功能6设计描述

功能6是允许用户在观看过的电影后，发表自己对这部电影的评价。

7.6.2功能6界面描述

（1进入电影详细介绍页

（2浏览到评论区，点击“发短评”按钮发送评价

**7.7功能7设计**

7.7.1功能7设计描述

功能7是当用户看到喜欢的电影时，可以通过这个软件购买附近影院的电影票。

7.7.2功能7界面描述

有两种购票方式：

（1从电影详细页点击“立即购票”→进入“选择影院”页面→进入“影院详细信息”页→选择观看的时间段，点击“选座购”按钮→进入“选座”页面→确认合适的位置之后，可进行支付

（2从“电影院列表”页点击“电影院列表”项→进入“影院详细信息”页→选择观看的电影→选择观看的时间段，点击“选座购”按钮→进入“选座”页面→确认合适的位置之后，可进行支付

## 附录

**8.1使用的开源代码说明**

无

**8.2使用的其他课程项目说明**

|  |  |  |
| --- | --- | --- |
| **项目名称** | **课程名称** | **用途** |
| 电影购票系统 | 系统分析与设计 | 整个系统 |

**8.3参考资料**

|  |  |  |
| --- | --- | --- |
| **名称** | **来源** | **备注** |
| 《UML和模式应用》 | 机械工业出版社 | 第三版 |