

# 信息工程学院软件工程专业

## 《数据库系统原理》课程设计报告

设计名称： 相机租赁系统

学生学号： 20172430303

学生姓名： 陈振山

专业班级： 计算机科学与技术2班

设计地点： 信息工程学院三楼南机房

设计时间： 2019.08.26-2018.09.06

指导教师评价：	成绩：
1. 创新/特色 【好】 【较好】 【一般】 【无】	
2. 写作情况 【好】 【较好】 【一般】 【差】	
3. 需求分析 【好】 【较好】 【一般】 【差】	
4. 设计合理性 【好】 【较好】 【一般】 【差】	
5. 系统实现 【好】 【较好】 【一般】 【差】	
6. 总结 【好】 【较好】 【一般】 【差】	
7. 演示情况 【好】 【较好】 【一般】 【差】	

# 《数据库系统原理》课程设计

## 1、设计目的

《数据库系统原理》课程设计是软件工程和计算机科学与技术专业集中实践性环节之一，旨在让同学们加深对数据库基础理论和基本知识的理解，掌握设计数据库管理系统的基本方法，锻炼运用知识解决实际问题的动手能力。

## 2、任务与要求

要求学生们从给定的设计题目中进行选择，进行需求分析，概念设计、逻辑设计，数据库的设计过程，表结构、表之间的关联，给出视图的定义、触发器的定义、索引，安全性的实现，用 SQL 语句等在 SQL Server 2008 系统中实现数据库的数据输入，查询，更新和输出；给出实现效果截图及部分测试结果。

## 3、相机租赁系统课程设计

### 3.1 引言

现在手机的拍照质量已经能够满足人们对于拍照的需求了，但是共享相机应用的开发还是很有必要。手机拍照存在着很多的不足，在一些场合像是出外旅游，重要的场合拍照，这些都需要专业的拍照设备。只有相机才能拍出质量高，画面好的照片。不可否认的是手机拍照的优势，简单实用场景广泛。只需要轻轻地按一下拍照键，就能生成图片，还配有各种各样的功能和表情，丰富拍摄体验，应对日常的自拍等。但是对于有时候对于要求高一点的时候，手机拍照就显得不足。就需要专业的相机拍照来弥补。面对一些大场景的拍照，特写，手机拍出来的照片会比较的模糊，成像不清晰，噪点多等问题。而相机能够解决这些问题，通过拍摄技巧的拍摄参数的设定，便能解决这些问题。共享相机应用面对的市场需求是手机拍照功能的不足，相机流入渗透用户率低下下的市场的状况，所以共享相机应用的开发很有必要。

### 3.2 需求分析

#### 3.2.1 系统调查的方法：

系统需求的调查方法为访谈法和亲自参加业务活动。在做系统之前亲自询问了在网上租赁过相机的同学，了解相机租赁的基本流程和需要注意的步骤。自己也亲自上淘宝，京东等网上商店体验了一遍相机租赁的流程。

系统的主要任务是用户进行浏览、下单，管理员管理用户、相机、订单，实现对其的增删改查，并把结果保存到数据库中。

完整性要求：防止不符合语义的数据进入数据库，定义断言实现。

安全性要求：需要防止客户对数据进行未经授权的访问，定义视图实现。

性能要求：最好能够实现并发访问，允许多个用户同时对数据库中的数据访问。

一致性要求：防止数据库进入不一致状态。

数据库要求：各种各样的故障都可能发生，出现意外时，尽可能的确保任何数据在任何情况下都不会丢失。使用数据库镜像技术实现。

### 3.2.2 业务流程

#### (1) 业务流程

用户登录系统，浏览相机信息，生成相应订单。

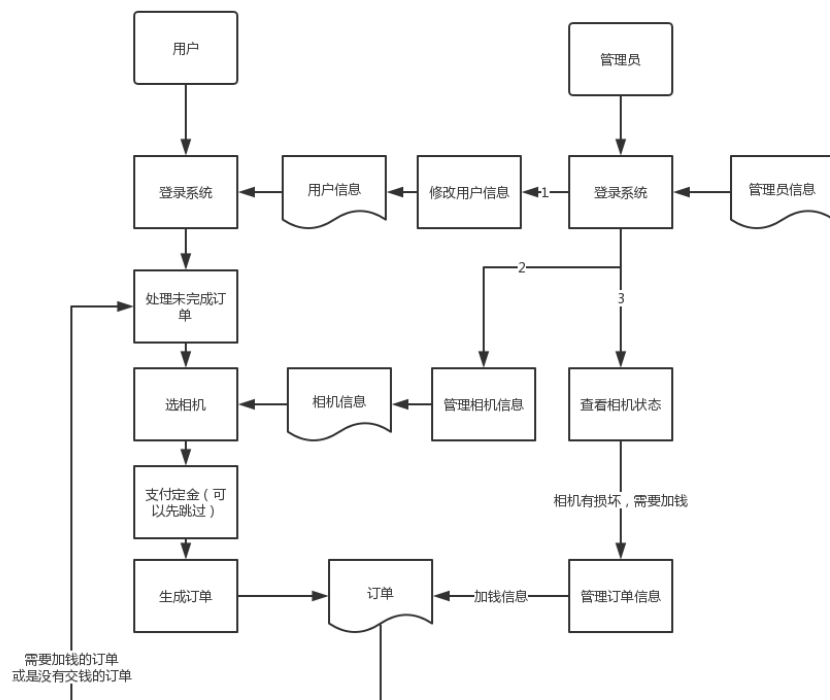
用户可以对订单进行结算。

管理员登录系统，管理管理员信息、管理用户信息、管理相机信息、管理订单信息。

管理员对相机进行检查，可对相应订单进行加钱处理。

管理员和用户之间可以相互切换。

#### (2) 业务数据流图



#### (3) 数据字典

数据项:

数据项名	含义说明	别名	数据类型	长度	取值范围	与其他数据项的逻辑关系
用户名	客户的唯一标识	客户名	字符型	45	字母、数字	无

用户密码	用户的密码	无	字符型	45	字母、数字	无
用户余额	用户账户剩余金额	无	双精度浮点型	20	数字、小数点、正负号	无
性别	用户的性别	无	字符型	2	Man、Woman	无
联系电话	用户的联系电话	无	字符型	45	数字	无
管理员名	管理员的唯一标识	无	字符型	45	字母、数字	无
管理员密码	管理员的密码	我	字符型	45	字母、数字	无
订单编号	订单的唯一标识	无	字符型	45	数字	无
租相机数量	租借相机的数量	租借相机的数量	整型	20	数字	无
总费用	租借相机的总费用	最终支付的费用	双精度浮点型	40	数字、小数点	可由租赁天数、租赁数量、相机单价、唯爱费用计算得出
相机单价	相机的单价，以天为单位	无	双精度浮点型	40	数字、小数点	无
相机编号	相机的唯一标识	无	字符型	45	数字	无
相机型号	相机的型号	无	字符型	45	英文字母、数字	无

数据处理过程:

处理过程名	输入{数据流}	输出{数据流}	处理	说明
登录系统	用户输入的账	当前用户的信	根据数据库的	对用户输入账

	号密码	息	信息验证账号 密码是否对应	号密码进行验证
生成订单	用户选择的相机信息	生成好的相机 订单信息	生成订单编号，打包成 sql 语句发送给数据库	系统根据用户 输入生成订单
修改用户信息	管理员对用户 信息的修改	修改好的用户 信息	将对用户信息的 处理包装成 sql 语句发送给 mysql	系统根据管理 员的操作，对用户信息进行 修改
管理订单信息	管理员对订单 信息的修改	修改好的订单 信息	将对订单信息的 处理包装成 sql 语句发送给 mysql	系统根据管理 员的操作，对订单信息进行 修改
管理相机信息	管理员对相机 信息进行修改	修改好的相机 信息	将对相机信息的 处理包装成 sql 语句发送给 mysql	系统根据管理 员的操作，对相机信息进行 修改

### 3.2.3

#### (1) 系统功能描述

登录功能：用户输入账号密码，系统根据数据库信息进行验证，可用户可管理员交替登录

用户浏览功能：用户可查询、排序相机信息，进行下单

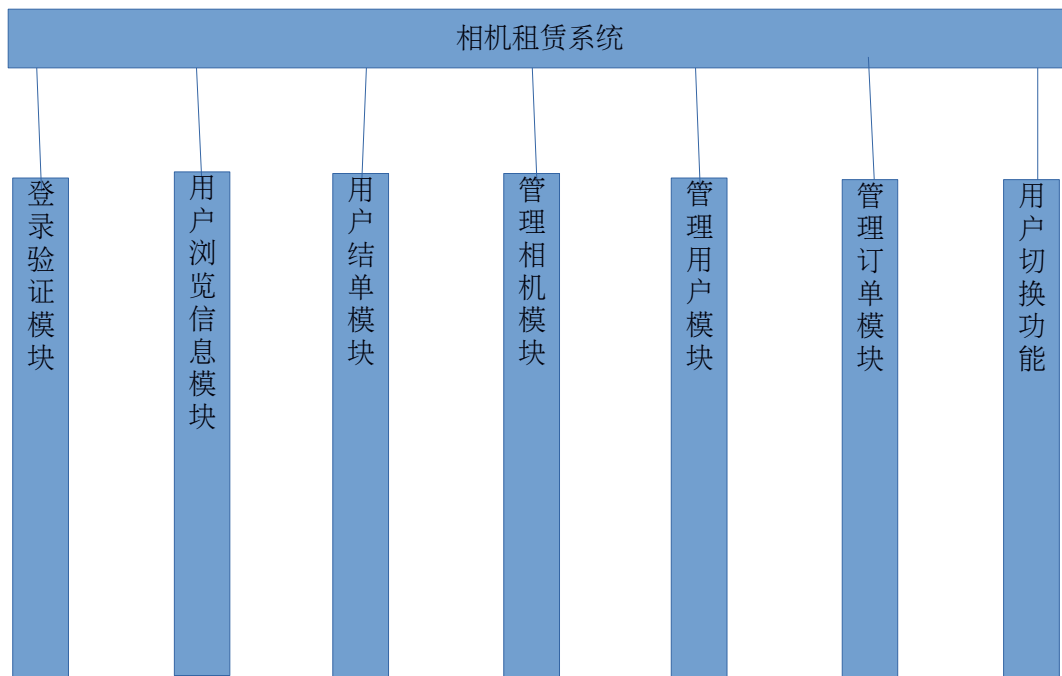
用户结单功能：用户可结算自己的订单

管理员管理用户功能：对用户进行增删改查

管理员管理相机功能：对相机信息进行增删改查

管理员管理订单功能：对订单进行增删改查

#### (2) 功能模块图



### 3.3 系统设计

#### (1) 数据库概念设计

##### 1) 基本思想:

将需求分析得到的用户需求抽象为信息结构，分析数据字典中数据字典间内在语义关联，并将其抽象表示成数据的概念模式。建立一个更真实、充分反应现实世界，易于理解、易于修改、易于向实际的 **DBMS** 支持的数据模型转换。

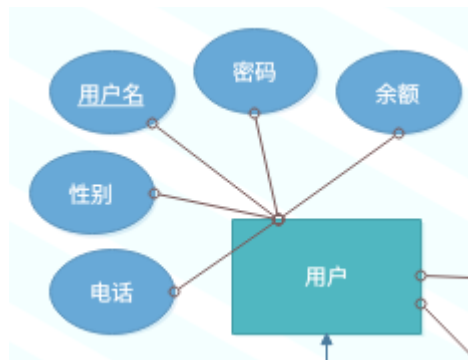
##### 2) 原理方法:

采用自底向上的方法。定义各局部应用的概念结构，然后逐步将他们集成起来，最终得到全局概念模式。

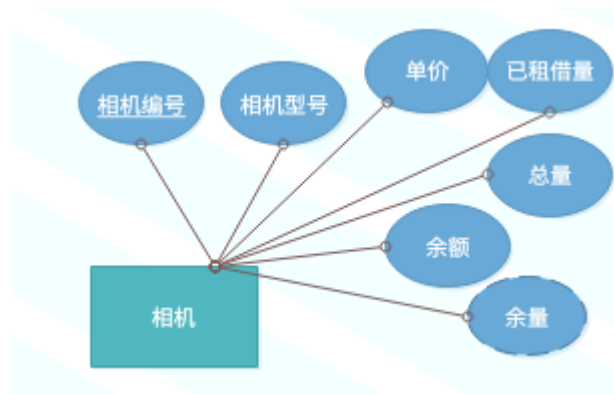
##### 3) 系统 E-R 图

局部 ER 图:

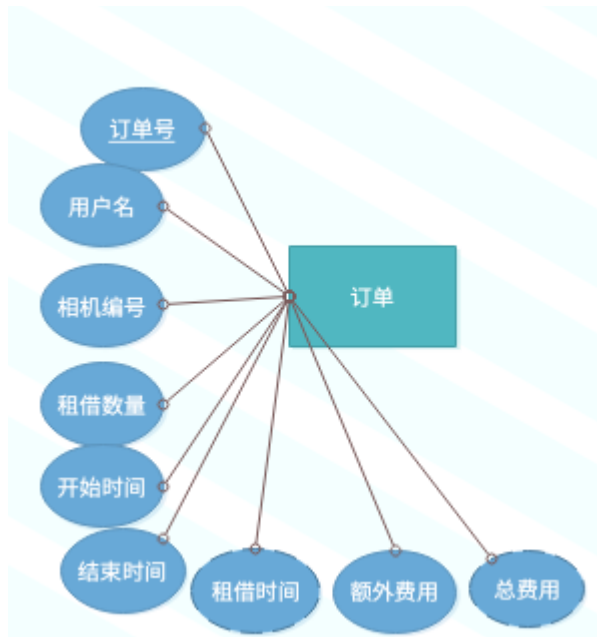
用户:



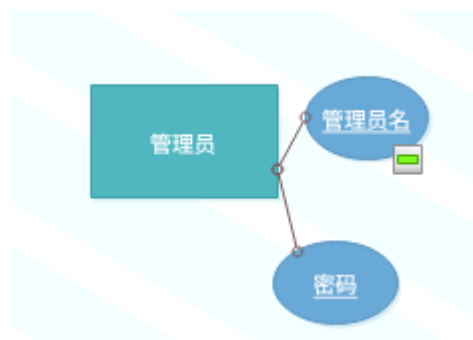
相机:



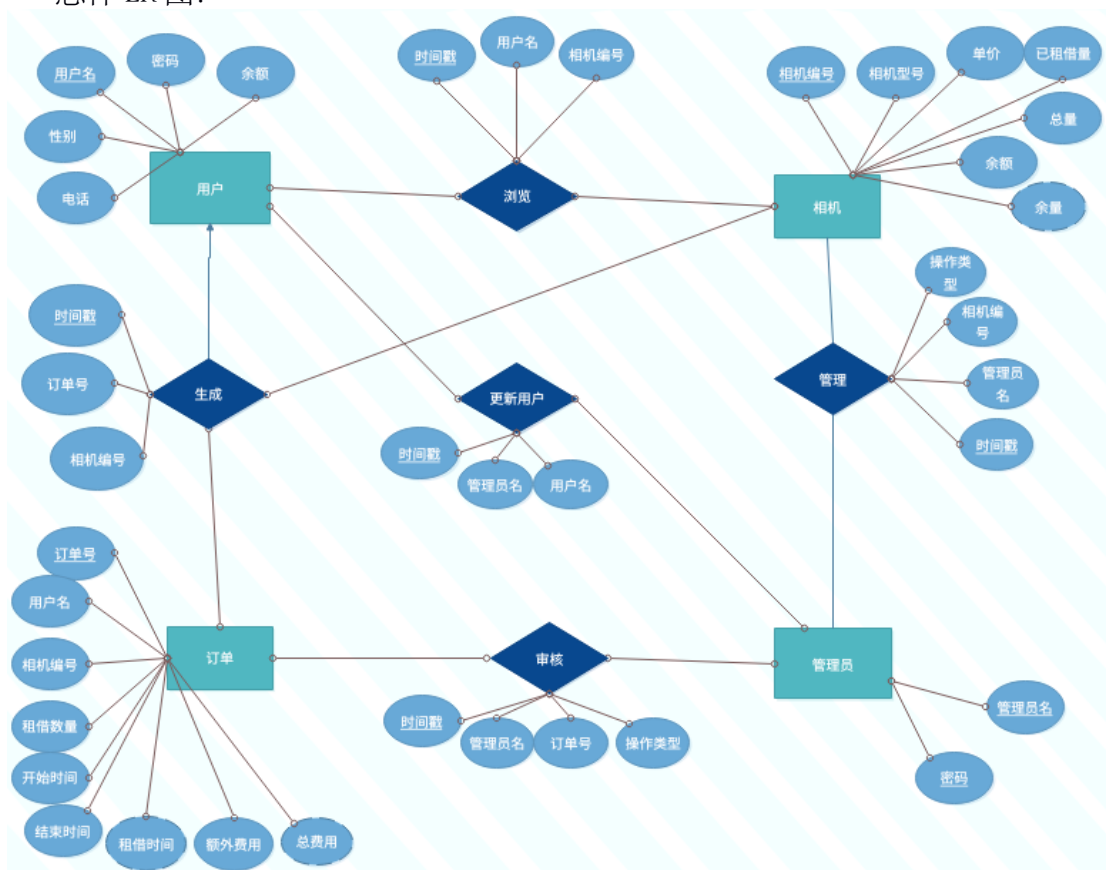
订单:



管理员:



总体 ER 图:



#### 4) 系统设计

系统应该具有权限设置、登录验证、相机信息浏览功能、以及管理各种信息功能。

#### 5) 功能设计

用户浏览功能: 用户可对相机信息进行查找、排序选择想要的信息。

用户结算功能: 用户点击相应订单进行结算。

管理员管理功能: 管理员可对数据库中的任何表做修改。

切换登录功能: 管理员和用户之间可以相互切换登录。

#### 6) 安全设计

用户权限: 只有管理员才能对该数据库中的对象进行任何操作, 为了能让其它用户使用该数据库, 需要进行权限设置。

访问权限: 为应用程序不同的部分建立不同的数据库账号, 使得它们能对数据库对象行使非常有限的权限。对这些账号应该只赋予最需要的权限, 同时应该防止相同的用户能够在不同的使用情况与数据库进行交流。



## (2) 数据库逻辑设计

基本思想：将概念设计阶段设计好的全局 ER 图抓换成 DBMS 产品所支持的数据模型，并进行规划化和优化，为每个应用设计外模式。

关系模式：（有×的属性是主码）

客户（×客户名，密码，余额，联系电话，性别）

管理员（×管理员名，密码）

相机（×相机编号，相机型号，单价，余量，总量）

订单（×订单编号，客户名，租借数量，相机编号，归还日期，额外加钱）

生成（×时间戳，订单编号，相机编号）

浏览（×时间戳，客户名，相机编号）

审核（×时间戳，管理员名，订单编号，操作类型）

更新用户（×时间戳，管理员名，客户名，操作类型）

管理（×时间戳，管理员命，相机编号，操作类型）

（说明：虽然生成、浏览、审核、更新用户、管理等关系主码一样，但它们是对应关系的主码，是不一样的，不能合并。）

对应的英文关系模式

```
customers(user_name,user_password,balance,phone,sex)
admins(admin_name, admin_password)
cameras(camera_no, name, total_num, rent_num, price)
indents(order_no, user_name, rent_num, camera_no, start_time,
rent_time, extra_charge)
manage(admin_name, camera_no, time, operate)
update_user(admin_name, custom_name, time, operate)
browse(user_name, camera_no, time)
```

## (3) 数据库物理结构设计

物理设计的基本思想：根据逻辑设计出的逻辑模式，DBMS 及计算机系统所提供的手段和施加的限制，设计数据库的内模式，即文件结构，各种路径，控件分配，记录的存取方式等，为逻辑数据结构选取一个最合适的应用环境的物理结构。

用户基本信息：

数据项名	数据类型	说明
user_name	varchar(45)	用户的名称，是用户的唯一标识
user_password	varchar(45)	用户密码
balance	float	用户钱包里的余额

sex	varchar(45)	性别，取值为 man 或 woman
phone	varchar(45)	联系电话

管理员基本信息:

数据项名	数据类型	说明
admin_name	varchar(45)	管理员名，唯一标识
admin_password	varchar(45)	管理员密码

订单基本信息:

数据项名	数据类型	说明
order_no	varchar(45)	订单编号，唯一标识
user_name	varchar(45)	下该订单的用户名，是外键
rent_num	int(11)	租借相机的数量
camera_no	varchar(45)	该订单中相机的编号
start_time	int(11)	租赁相机开始的时间
rent_time	int(11)	租赁相机归还的时间
extra_charge	int(11)	额外支付的费用
rent_days	int(11)	租借天数，可由 start_time-rent_time 得出
final_charge	int(11)	最终用户支付的费用，是派生属性

相机基本信息:

数据项名	数据类型	说明
camera_no	varchar(45)	相机编号，相机的唯一标识
name	varchar(45)	相机品牌
total_num	int(11)	库存量
rent_num	int(11)	已租借量
price	int(11)	相机日租金

建立的索引:

#在 customs 表的 balance 属性上建立索引

create index customs\_balance\_index on customs(balance);

#在 cameras 表的 price 属性上建立索引

create index cameras\_price\_index on cameras(price);

## (4) 系统功能设计

(功能模块图及说明、安全设计、系统结构图及说明等)

### 3.4 数据库实施

数据库实施包括建立数据库模式、加载数据。

admins 表的定义:

```
CREATE TABLE `admins` (`admin_name` varchar(45) NOT NULL,
`admin_password` varchar(45) NOT NULL, PRIMARY KEY (`admin_name`),
UNIQUE KEY `admin_name_UNIQUE` (`admin_name`));
```

cameras 表的定义:

```
CREATE TABLE `cameras` (`camera_no` varchar(45) NOT NULL, `name`
varchar(45) DEFAULT NULL, `total_num` int(11) DEFAULT NULL,
`rent_num` int(11) DEFAULT NULL, `price` int(11) DEFAULT NULL,
PRIMARY KEY (`camera_no`), UNIQUE KEY `carid_UNIQUE` (`camera_no`),
KEY `cameras_price_index` (`price`));
```

customs 表的定义:

```
CREATE TABLE `customs` (`user_name` varchar(45) NOT NULL,
`user_password` varchar(45) NOT NULL, `balance` float DEFAULT NULL,
`sex` varchar(45) DEFAULT NULL, `phone` varchar(45) DEFAULT NULL,
PRIMARY KEY (`user_name`), UNIQUE KEY `user_name_UNIQUE`
(`user_name`), KEY `customs_balance_index` (`balance`));
```

indents 表的定义:

```
CREATE TABLE `indents` (`order_no` varchar(45) NOT NULL,
`user_name` varchar(45) NOT NULL, `rent_num` int(11) DEFAULT NULL,
`camera_no` varchar(45) NOT NULL, `start_time` int(11) DEFAULT NULL,
`rent_time` int(11) DEFAULT NULL, `extra_charge` int(11) DEFAULT
NULL, PRIMARY KEY (`order_no`), UNIQUE KEY `id_UNIQUE`
(`order_no`), KEY `fk_indent_1` (`user_name`), KEY `fk_indent_2`
(`camera_no`), CONSTRAINT `fk_indent_1` FOREIGN KEY (`user_name`)
REFERENCES `customs` (`user_name`) ON DELETE NO ACTION ON UPDATE NO
ACTION, CONSTRAINT `fk_indent_2` FOREIGN KEY (`camera_no`) REFERENCES
`cameras` (`camera_no`) ON DELETE NO ACTION ON UPDATE NO ACTION);
```

browse 表的定义:

```
CREATE TABLE `browse` (`user_name` varchar(45) DEFAULT NULL,
`camera_no` varchar(45) DEFAULT NULL, `time` int(11) NOT NULL,
PRIMARY KEY (`time`), UNIQUE KEY `time_UNIQUE` (`time`), KEY
`fk_browse_1_idx` (`camera_no`), KEY `fk_browse_2` (`user_name`),
```

```

CONSTRAINT `fk_browse_1` FOREIGN KEY (`camera_no`) REFERENCES
`cameras` (`camera_no`) ON DELETE NO ACTION ON UPDATE NO ACTION,
CONSTRAINT `fk_browse_2` FOREIGN KEY (`user_name`) REFERENCES
`customs` (`user_name`) ON DELETE NO ACTION ON UPDATE NO ACTION);

```

manage 表的定义:

```

CREATE TABLE `manage` (
  `admin_name` varchar(45) DEFAULT NULL,
  `camera_no` varchar(45) DEFAULT NULL,
  `time` int(11) NOT NULL,
  `operate` varchar(45) DEFAULT NULL,
  PRIMARY KEY (`time`),
  UNIQUE KEY `time_UNIQUE` (`time`),
  KEY `fk_manage_1` (`admin_name`),
  CONSTRAINT `fk_manage_1` FOREIGN KEY (`admin_name`) REFERENCES `admins`
  (`admin_name`) ON DELETE NO ACTION ON UPDATE NO ACTION);

```

update\_user 表的定义:

```

CREATE TABLE `update_user` (
  `admin_name` varchar(45) DEFAULT NULL,
  `custom_name` varchar(45) DEFAULT NULL,
  `time` int(11) NOT NULL,
  `operate` varchar(45) DEFAULT NULL,
  PRIMARY KEY (`time`),
  KEY `fk_update_1` (`admin_name`),
  CONSTRAINT `fk_update_1` FOREIGN KEY
  (`admin_name`) REFERENCES `admins` (`admin_name`) ON DELETE NO ACTION
  ON UPDATE NO ACTION);

```

#建立触发器，保证租借数量不大于总量。触发器实现断言功能

```

use czs;
drop trigger camera_total_gt_rent;
delimiter $$
create trigger camera_total_gt_rent
after update
on cameras for each row
begin
  DECLARE msg VARCHAR(200);
  if new.rent_num > new.total_num
  then
    set msg = concat(" 库存不足： 已租赁数量， 总量 ",
old.rent_num, new.total_num);
    signal sqlstate 'HY000' set message_text = msg;
  end if;
end;
$$
delimiter ;

```

#触发器

#当生成新的订单，更改 cameras 表中的 rent\_num 数量。

```

drop trigger update_camera_rent_num;

```

```

delimiter $$
create trigger update_camera_rent_num
after insert
on indents for each row
begin
    update cameras set rent_num = new.rent_num+cameras.rent_num
where camera_no = new.camera_no;
end;
$$
delimiter ;

```

#创建我的订单视图，显示的信息很全

```

drop view my_indent;
create view my_indent(order_no, user_name, camera_no, rent_num,
price, start_time, rent_time, rent_days, rent_charge, extra_charge,
finall_charge)
as
select      order_no,      user_name,      indents.camera_no,
indents.rent_num,  price,  start_time,  rent_time,  (rent_time-
start_time)/(24*60*60),      indents.rent_num*(rent_time-start_time)/
(24*60*60)*price,      extra_charge,      indents.rent_num*(rent_time-
start_time)/(24*60*60)*price + extra_charge
from indents, cameras
where indents.camera_no = cameras.camera_no;

```

### 3.6 系统调试和测试

登录界面:



The image shows a standard login window with a title bar containing the text '登录界面'. The window has a light gray background. It features two text input fields: the first is labeled '账户:' (Account) and the second is labeled '密码:' (Password). Below the password field, there is a checkbox followed by the text '管理员' (Administrator). At the bottom of the window, there is a button labeled '登录' (Login).

管理管理员信息界面：

管理员界面

管理管理员信息

管理用户信息

管理相机信息

管理订单信息

切换用户

	账户名	密码
1	1	1
2	Adi	111
3	admin_1	111
4	admin_2	111
5	czs	111

功能区：

增加管理员

删除管理员

修改管理员

查找管理员

管理用户信息界面：

管理员界面

管理管理员信息

管理用户信息

管理相机信息

管理订单信息

切换用户

	用户名	密码	余额	性别	联系方式
1	1	1	1920.02		
2	czs	111	652.071	man	13460699149
3	Lily	111	2000	woman	120
4	Mike	111	2000	man	13460699149
5	Tom	111	2000	woman	

增加用户

删除用户

修改用户

查找用户

## 管理相机信息界面：

管理员界面

管理管理员信息

管理用户信息

管理相机信息

管理订单信息

切换用户

	相机编号	相机型号	总量	已借出量	单价/天
1	Canon5D3	Canon	10	1	45
2	Canon5D4	Canon	10	5	50
3	Canon6D	Canon	10	2	100
4	Canon6D2	Canon	10	0	70
5	GoPro7	GoPro	10	0	40
6	leica0	leica	10	3	32
7	leica1	leica	10	6	32
8	leica2	leica	10	2	32
9	leica3	leica	10	0	32
10	leica4	leica	10	5	32
11	leica5	leica	10	0	32

功能区：

显示相机

增加相机

删除相机

修改相机

请根据下列属性查询：

☒ 相机编号

☐ 相机型号

查找相机

## 查询相机型号为 Canon 的相机信息：

管理员界面

管理管理员信息

管理用户信息

管理相机信息

管理订单信息

切换用户

	相机编号	相机型号	总量	已借出量	租金/天
1	Canon5D3	Canon	10	1	45
2	Canon5D4	Canon	10	5	50
3	Canon6D	Canon	10	2	100
4	Canon6D2	Canon	10	0	70

功能区：

显示相机

增加相机

删除相机

修改相机

请根据下列属性查询：

☐ 相机编号

☒ 相机型号

Canon

查找相机

## 管理订单界面：

管理订单界面

管理管理员信息

管理用户信息

管理相机信息

管理订单信息

切换用户

	订单编号	用户名	相机编号	租借数量	单价/天	开始时间	归还时间	租借天数	租借费用	额外费用	总费用
1	1567661437	1	SonyA7S2	2	30	1567661444	1567785600	1.437	86.2194	0	86.2194
2	1567661630	1	leica0	2	32	1567661636	1567872000	2.4348	155.8252	0	155.8252
3	1567662612	czs	leica1	1	32	1567662617	1567785600	1.4234	45.5493	50	95.5493
4	1567662703	Tom	leica2	1	32	1567662707	1567785600	1.4224	45.5159	0	45.5159
5	1567662712	Tom	leica1	1	32	1567662717	1567785600	1.4223	45.5122	0	45.5122

按下面属性排序：

☒ 租借数量

☐ 租借天数

☐ 租借费用

☐ 额外费用

☐ 总费用

排序

根据下面属性查找：

☒ 订单编号

☐ 用户名

☐ 相机编号

查找订单

功能区：

显示订单

修改订单

删除订单

## 对订单按总费用进行从高到低排序：

管理订单界面

管理管理员信息

管理用户信息

管理相机信息

管理订单信息

切换用户

	订单编号	用户名	相机编号	租借数量	单价/天	开始时间	归还时间	租借天数	租借费用	额外费用	总费用
1	1567662773	Tom	leica4	1	32	1567662775	1567699200	0.4216	13.4907	0	13.4907
2	1567662752	Tom	leica4	1	32	1567662754	1567699200	0.4218	13.4985	0	13.4985
3	1567662723	Tom	leica4	1	32	1567662729	1567785600	1.4221	45.5078	0	45.5078
4	1567662712	Tom	leica1	1	32	1567662717	1567785600	1.4223	45.5122	0	45.5122
5	1567662703	Tom	leica2	1	32	1567662707	1567785600	1.4224	45.5159	0	45.5159

按下面属性排序：

☐ 租借数量

☐ 租借天数

☐ 租借费用

☐ 额外费用

☒ 总费用

排序

根据下面属性查找：

☒ 订单编号

☐ 用户名

☐ 相机编号

查找订单

功能区：

显示订单

修改订单

删除订单



查找用户 czs 的所有订单信息：

管理员界面

管理管理员信息

管理用户信息

管理相机信息

管理订单信息

切换用户

	订单编号	用户名	相机编号	租借数量	单价/天	开始时间	归还时间	租借天数	租借费用	额外费用	总费用
1	1567662703	Tom	leica2	1	32	1567662707	1567785600	1.4224	45.5159	0	45.5159
2	1567662712	Tom	leica1	1	32	1567662717	1567785600	1.4223	45.5122	0	45.5122
3	1567662723	Tom	leica4	1	32	1567662729	1567785600	1.4221	45.5078	0	45.5078
4	1567662752	Tom	leica4	1	32	1567662754	1567699200	0.4218	13.4985	0	13.4985
5	1567662773	Tom	leica4	1	32	1567662775	1567699200	0.4216	13.4907	0	13.4907

按下面属性排序：

☐租借数量

☐租借天数

☐租借费用

☐额外费用

☒总费用

排序

根据下面属性查找：

☐订单编号

☒用户名

☐相机编号

Tom

查找订单

功能区：

显示订单

修改订单

删除订单

用户切换功能：

注意：切换用户不需要密码，切换管理员需要密码！

账号：

密码：

登录

用户按相机租金从小到大排序：

浏览相机

我的订单

切换用户

	相机编号	相机型号	总量	已借出量	租金/天
1	SonyA7S2	Sony	10	6	30
2	leica0	leica	10	3	32
3	leica1	leica	10	6	32
4	leica2	leica	10	2	32
5	leica3	leica	10	0	32
6	leica4	leica	10	5	32
7	leica5	leica	10	0	32
8	leica6	leica	10	0	32
9	leica7	leica	10	0	32
10	leica8	leica	10	0	32
11	leica9	leica	10	0	32
12	SonyA7R3	Sony	10	0	40
13	GoPro7	GoPro	10	0	40
14	SonyA7R2	Sony	10	0	40
15	Canon5D3	Canon	10	1	45
16	Canon5D4	Canon	10	5	50
17	SonyA7	Sony	10	0	60

功能区

根据下面属性排序：

租金

已借出量

排序

显示相机

下单

根据下面属性查找：

相机编号

相机型号

总量

已租借量

租金

查找

显示当前客户的订单：

浏览相机

我的订单

切换用户

当前账户余额：1920.02

结算

	订单号	用户名	相机编号	租赁数量	单价	开始时间	结束时间	租赁天数	租赁费用	额外费用	总费用
1	1567661437	1	SonyA7S2	2	30	2019/09/05	2019/09/07	1.437	86.2194	0	86.2194
2	1567661630	1	leica0	2	32	2019/09/05	2019/09/08	2.4348	155.8252	0	155.8252
3	1567667273	1	leica1	2	32	2019/09/05	2019/09/07	1.3694	87.64	0	87.64

结算成功界面：

浏览相机

我的订单

切换用户

当前账户余额：1920.02

结算

正确！

你已成功支付费用！

OK

	订单号	用户名	相机编号	租赁数量	结束时间	租赁天数	租赁费用	额外费用	总费用		
1	1567661437	1	SonyA7S2	2	30	2019/09/05	2019/09/07	1.437	86.2194	0	86.2194
2	1567661630	1	leica0	2	32	2019/09/05	2019/09/08	2.4348	155.8252	0	155.8252
3	1567667273	1	leica1	2	32	2019/09/05	2019/09/07	1.3694	87.64	0	87.64

满足需求分析的要求！

## 4.系统评价

### 4.1 系统特色

#### 1、用户切换功能

有了这个功能就可以不重新打开界面就切换用户登录或者管理员登录

#### 2、查询功能

用户可以直接输入想要租借的相机类型，系统直接查询出所有该类型的相机信息，方便用户查看、比较、选择。

#### 3、排序功能

用户可以排序，方便用户选择相机。比如可以对相机的单价进行排序，选择出最便宜的相机点击订单进行下载。

#### 4、自动刷新功能

当用户完成订单后、管理员对用户信息、相机信息、订单信息进行增删改查后，及时更新显示界面，不用再手动刷新了。

#### 5、后台自动记录浏览记录功能

当用户完成下单后，触发器自动往用户浏览记录表中插入用户下单的信息，当用户结单后，订单信息自动删除后，也能在浏览记录中查到已结算的订单；当管理员对用户信息、相机信息、订单信息进行增删改查后，自动保存修改记录，记录信息有时间戳、旧信息、信息信息、操作类型等。让所有操作都有迹可循，防止用户的非法输入以及管理员的错误操作。

#### 6、用时间戳作为主码

时间戳是 1970 年 1 月 1 日到现在的秒数。这个数是唯一的，用它作为订单的主码很合适。只要订单不是在同一秒同生成的，那么他的订单编号就是不同的，因为时间戳不同。用时间戳作为主码还很方便转换为时间，C++有直接的转换函数，把时间戳转换成自己想要的时间显示格式。

### 4.2 系统不足及改进

不足之处：前台没有显示用户及管理严的操作记录。不过也好做，只需在前台显示记录表的信息就可以了（不能开放删除、修改功能，操作记录只能增加，不能删除、修改）

## 5.设计心得

### 1、mysql 无法实现断言功能

这个是 mysql 的特性，mysql 设计时都没有设计断言，不过可以用触发器来实现断言功能。比如我上面写的用触发器实现防止已租借数量大于总数量的断言功能。

代码如下:

#建立触发器, 保证租借数量不大于总量。触发器实现断言功能

```
use czs;
drop trigger camera_total_gt_rent;
delimiter $$
create trigger camera_total_gt_rent
after update
on cameras for each row
begin
    DECLARE msg VARCHAR(200);
    if new.rent_num > new.total_num
    then
        set msg = concat(" 库存不足: 已租赁数量, 总量 ",
old.rent_num, new.total_num);
        signal sqlstate 'HY000' set message_text = msg;
    end if;
end;
$$
delimiter ;
```

2、Qt Creator 无法连接到数据库。报错: QSqlDatabase: QMYSQL driver not loaded

Qt Creator 编译器默认没有 mysql 数据库的驱动器, 如果自己下载 Qt 源码, 可以自己编译 mysql 的驱动器, 如果是 Windows 系统可以上网查 mysql 的动态链接库, 放到 Qt Creator 的目录下。

3、往 mysql 发送语句, 没有执行成功

如果要查询的信息有 varchar 类型, 命令要加双引号, 还要用\转义下。如下: `QString command = QString("select user_password from customs where user_name = \"%1\").arg(user_name);`

4、无法对视图进行更新:

如果视图是由多个基本表构建的, 则不允许更新该视图, 只能更新基本表。对基本表的修改, 会体现在视图上。想想也是, 视图中的总金额是由单价和天数得出来的。如果更新总金额, 那基本表中的单价和天数应该如何更新? 一个数进行因式分解不止有一个可能, 所以禁止更新视图。

5、时间显示问题

数据库中的 **date** 类型和 Qt 中的 **QDateTime** 类型, 不太一样, 直接转换效果不太好。用时间戳作为媒介就完整解决了这个问题。系统可以获取当前的时间戳, 它是无符号整型, 数据库中的时间也设置成整型, 这样就完美的解决了时间不兼容的问题。从数据库中读取整型类型的时间戳, 在前端系统中用 `.from_t()` 函数将其转换为 **QDateTime** 类型的数据, 即时间类型的数据, 接下来就很方便在前段显示了。