管理人员对基本信息的管理, 包括对汽车类型信息的增加 删除 修改和查询;

对会员类型信息的增加 删除 修改和查询

实体：相机、客户、管理员、订单、事故（事故可以先不实现）

联系：租赁、管理、生成、审核

实体集：

客户：×用户名，姓名，性别，电话，密码，账户余额

相机：×相机 编号，相机 名字，总数量，已租赁数量，出租单价

管理员：×用户名，密码

联系集：

租赁：交易编号，客户用户名（外），汽车编号（外），租赁时间，租赁金额

管理：×（用户名，汽车编号）

概念设计阶段：设计分ER图，然后再合并成一个ER图。

**系统流程：**

**功能模块介绍：**

**客户：**

**1、取相机**

登录系统，检查是否有需要加钱的订单，如果有，必须先支付完费用。浏览相机信息（车基本信息（相机型号、相机功能），剩余几个，被租赁次数，每天单价），选相机，选择借相机起止时间，下订单，支付费用，得到提相机码（先不实现）

**2、还相机**

登录系统，浏览订单信息，选择一个订单进行清算。

如果有需要加钱的订单，需完成这个订单，否则不能**取相机**。

**3、需要修改或不足的地方**

还相机时间不一定固定，用户想什么时候换都行，在结算时清算费用。

**管理员**

**1、登录系统可以管理相机信息（对基本信息的增删改查）**

对相机信息进行更新，并及时更新到数据库里，让用户看到最新的相机信息。

**2、管理客户信息（对基本信息的增删改查）**

对客户信息进行更新，并及时更新到数据库里，防止意外事故发生。

**3、管理订单**

查看订单

**4、审核相机**

用户还车，对相机进行检查（需要管理员到实地检查相机，不在该系统内实现，只处理管理员输入的相机状态，是否需要加钱）。如果相机损坏，需要加钱，管理员管理订单添加**加钱**选项。

**数据抽象：**

用户、管理员、相机、订单

组成成分：

用户：用户标示，密码，

**概念设计阶段：**

分ER图，然后汇总。我写在纸上了。

ER图的优化：订单金额。可以由其他属性计算出来。

**逻辑结构设计阶段：**

ER图转为关系模式，规范、优化、为每个应用设计外模式。

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（×订单编号）

查看（×（客户名，相机编号））

管理（×（管理员名，订单编号））

更新（×（管理员名，客户名））

合并有相同码的关系模式。

将**生成**关系合并到**订单**里

最终的关系模式为：

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，单价，余量）

订单（×订单编号，客户名，租借数量，相机编号，起始日期，归还日期，额外加钱，是否结算）订单加个状态。是否结算。

查看（×（客户名，相机编号））

管理（×（管理员名，订单编号））

更新（×（管理员名，客户名））

规范化：

该模式出现的问题。

上述关系模式是BCNF。

外模式的设计；

对于**订单**关系模式，**客户**不能查看**额外加钱**这一属性，额外定义一个视图（体现在客户填写订单时没有额外加钱这一属性。），**起始日期**也没必要让客户看到（没有预约功能，暂时不需要）。

处理计算属性：客户结算时定义的**视图**里应该有**费用**属性，由归还日期减去起始日期乘上相机的单价得出。

下单的视图：（相机编号，数量，归还日期）

系统处理时填上：订单编号，客户名

管理员可能会加上**额外加钱**属性

客户确定时的视图：订单编号，客户名，租借数量，相机编号，起始日期，归还日期

客户归还时的视图：订单编号，客户名，租借数量，相机编号，起始日期，归还日期，**额外加钱**

**物理设计：**

为**客户**、**管理员**、**相机**、**订单**在主码上建立索引。

**额外的：**

1、加断言：相机余量不能超过总量，余量和总量不能小于0

mysql不支持断言。没有referencing 来标示旧值，利用触发器实现不出来，放一放。。。

实现了。

2、创建函数快速插入数据。

发现插入数据并不是按照顺序来的

3、为性别添加约束。

数据库约束不会，前台约束吧。

**界面**

**流程：**

**借**

1、登录，先检查有没有未交钱的订单（先不实现）（可能包含加钱的信息）。交完钱才能浏览相机信息。【检索indents表，是否有该用户名，检查是否到归还日期。到归还日期，弹出交钱按钮。】

2、选择相机信息（点中单元格），按下下单按钮，显示下单界面。显示信息有：订单编号，当前用户名，相机编号，用户选择租借数量。【根据上面信息，在indent表中生成一条记录，让extra\_charge=0】

3、点击我的点单按钮，显示和当前客户有关的订单。

不知道tabWidget控件被点击时发出的信号是什么，放到该界面的构造函数里。

4、生成新的订单后，要及时刷新我的订单的显示。

**还**

1、进入我的订单，选择相应订单（最好是点中order\_no），点结算按钮，自动扣钱，不在弹出什么界面了。。（扣余额，余额不足，弹出警告）【进入界面，在indent表根据用户名列出所有订单记录，根据选中的order\_no结算。】

2、我的订单界面里开始时间和结束时间是字符串型的时间戳。

先转换成unsigned int类型，在转换为年月日展示给用户。

存在隐患，int→unsigned int。不过目前为止没有什么问题。

**管理员：3个tab页**

1、管理管理员信息

列出所有管理员信息，选中，点修改按钮，弹出管理员详细信息对话框，修改后，点保存或取消，重新写会数据库。

2、管理客户信息

3、管理相机信息

**界面设计：**

3、订单界面

4、交钱界面

**编程阶段：**

1、注意查询varchar类型，命令要加双引号，还要用\转义下。

QString command = QString("select user\_password from customs where user\_name = \"%1\"").arg(user\_name);

2、点击下单按钮，获取选择的相机信息。

下面生成indent表里的一条记录。

根据时间戳生成订单编号，记录当前用户名，开始时间，用户输入归还时间，相机编号。

往数据库里insert 订单编号，用户名，相机编号，开始时间，归还时间，不需要加钱。

实现：

获取相机编号，触发信号（传递相机编号），显示订单界面，客户确认订单信息，提交或取消。

3、增强鲁棒性

1、如果没有选择相机编号就下单，系统会崩溃。加一个判断，当数据路返回结果为0时，弹出一个警告框。

2、按下提交按钮，就出现indent\_show界面，如果没有选中相机编号也会出现这个界面。难受。

3、从主界面往子界面传递数据

在子界面写一个public函数（带参数），在主界面生成子界面的对象，调用这个public函数，就实现了参数的传递。

4、普通界面实现模式界面的阻塞其他界面响应的效果。

加一setWindowModality(Qt::ApplicationModal);//定义为模式界面，当有此界面时，阻塞其他界面的响应。

5、用户名的传递。定义一个静态成员变量。

6、内存溢出了。。。。

好像是声明了Date类型的变量，但是没有初始化造成的。

关闭indent界面还是会溢出，不过没啥影响。现在又没这个问题了，先不管（隐藏了一个大BUG，出问题再解决吧。）

运行一段时间程序自动崩溃了？？！！

我把模式对话框去掉了，貌似解决这个问题了。

6、date类型数据处理

转换成时间戳，再转换成字符串存到数据库里。懒得转换了。

7、明明已经插入数据库，但是query.exec()还是返回false！难受

8、计算金额的，前段实现有点麻烦，直接定义个视图吧。

indent表和cameras表

create view my\_indent(order\_no, user\_name, camera\_no, rent\_num, price, start\_time, rent\_time,rent\_days, charge, extra\_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 as charge, extra\_charge

from indents, cameras

where indents.camera\_no = cameras.camera\_no;

9、点击提交按钮，获取表格中的final\_charge。获取用户的余额，扣除，重新写入。update

10、你下单了，需要更新cameras表中的rent\_num项。