

深圳大学实验报告

课程名称： 基于 UML 的面向对象系统分析与设计

实验项目名称： 实验 4 面向对象的系统分析与设计（二）

学院： 计算机与软件学院

专业： 软件工程

指导教师： 刘嘉祥

报告人： 郑彦薇 学号： 2020151022 班级： 软件工程 01 班

实验时间： 2022/11/09~2022/11/29

实验报告提交时间： 2022 年 11 月 29 日

教务部制

实验目的与要求：

熟悉使用 Enterprise Architect 工具绘制分析模型，并学会对实际案例进行用例分析。
具体包括：

- (1) 定义系统备选架构；
- (2) 识别分析类；
- (3) 构造用例实现-绘制顺序图；
- (4) 构造用例实现-绘制 VOPC 类图；
- (5) 完成实体类类图；
- (6) 对案例进行用例分析。

方法、步骤：

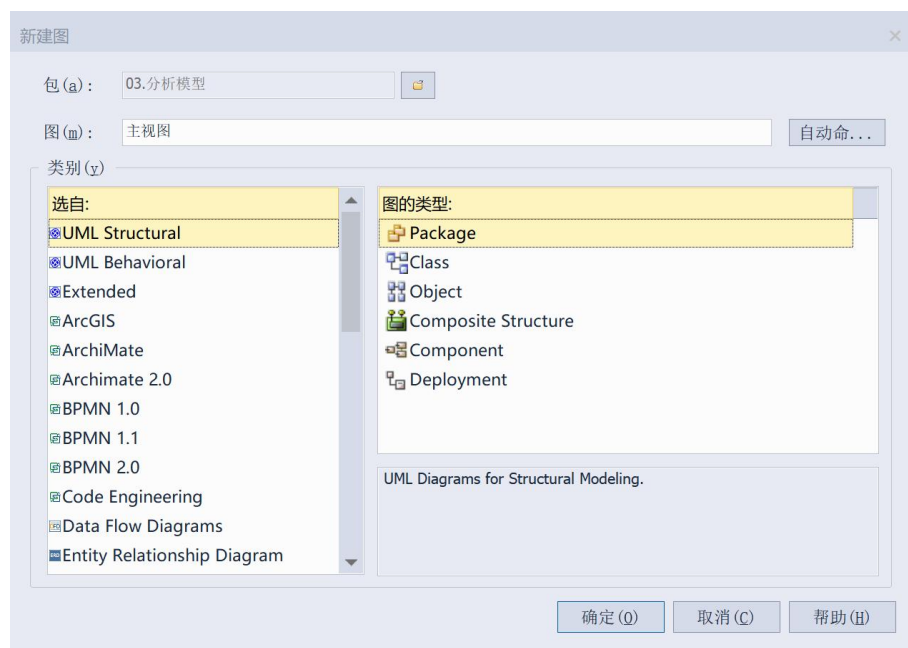
使用 EA 工具进行案例分析模型的绘制。对医院预约挂号系统案例进行用例分析。

实验过程及内容：

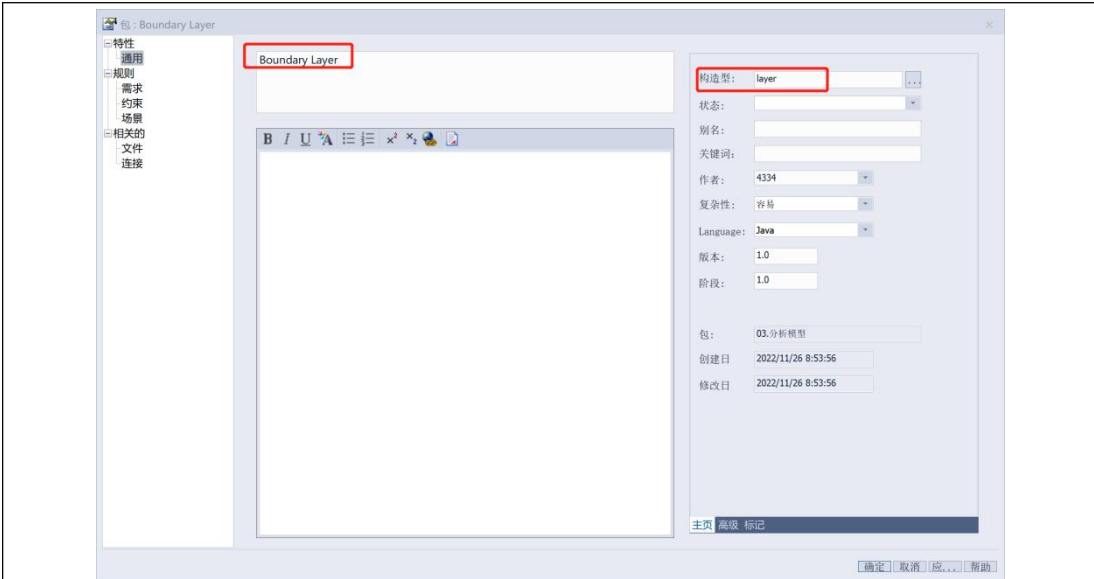
根据实验 4 的实验指导，完成以下 6 个练习。

练习 1：定义系统备选架构

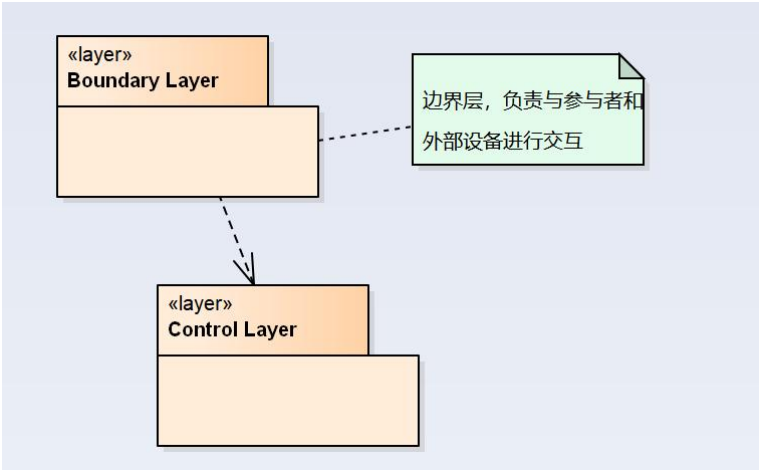
1.1. 打开项目旅游.cap，在分析模型中新建主视图，类型为包图（UML Structural → Package）。



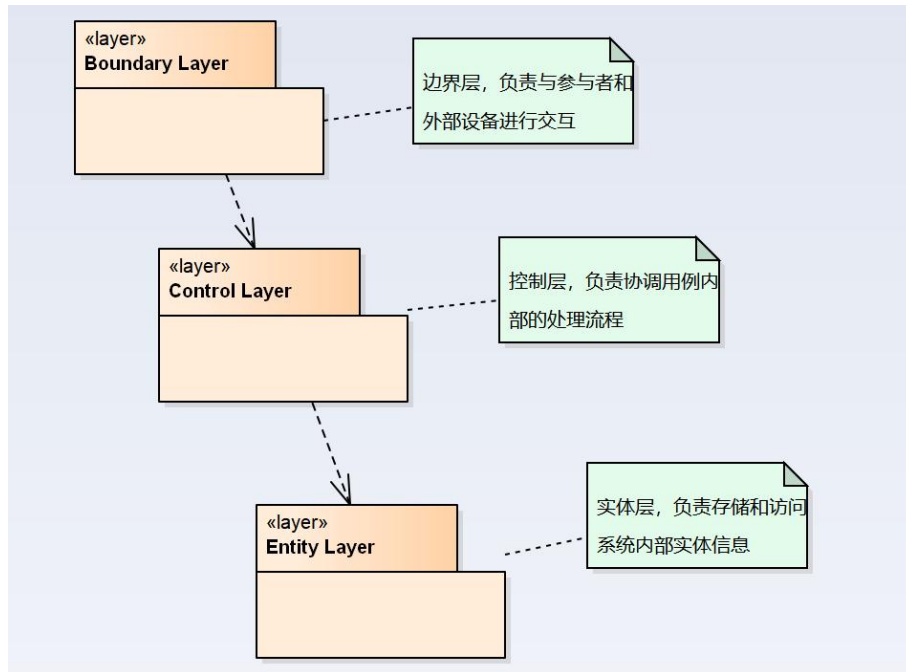
1.2. 打开上述新建的主视图，添加元素“Package”，命名为“Boundary Layer”。在弹出的特性窗口中，选择特性-通用，将构造型设置为 layer，点击确定完成添加。



1.3. 绘制依赖关系、注释。

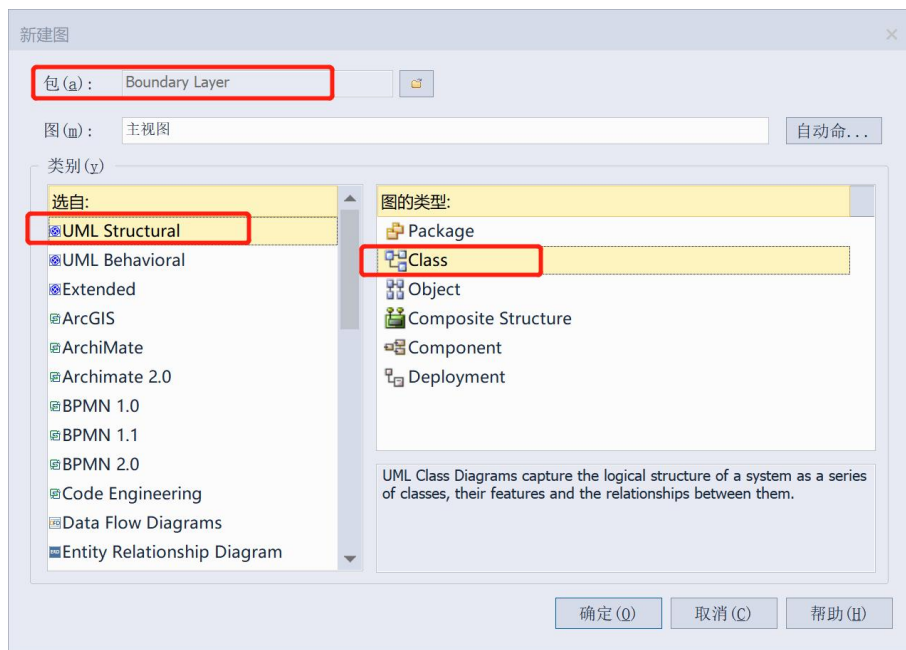


1.4. 最终绘制效果图如下图所示。

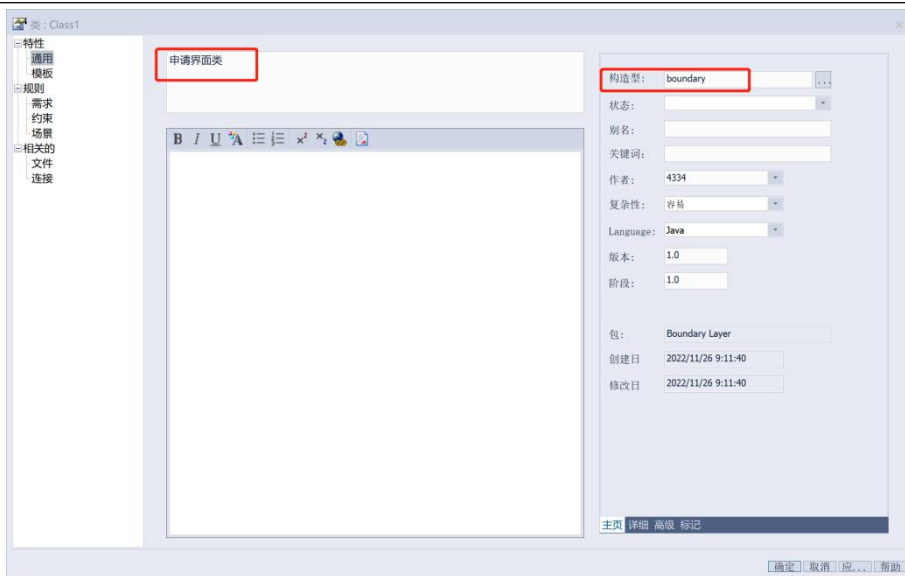


练习 2：识别分析类

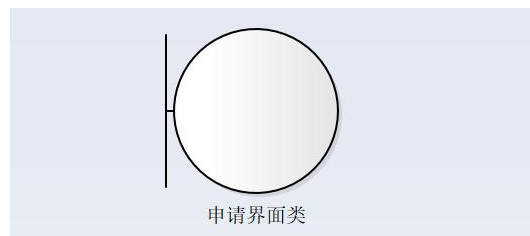
2.1. 在项目浏览器中点击练习 1 中创建的包“«layer» Boundary Layer”，选择新增主视图，类型为“类图”（即 UML Structural→Class）或“包图”。



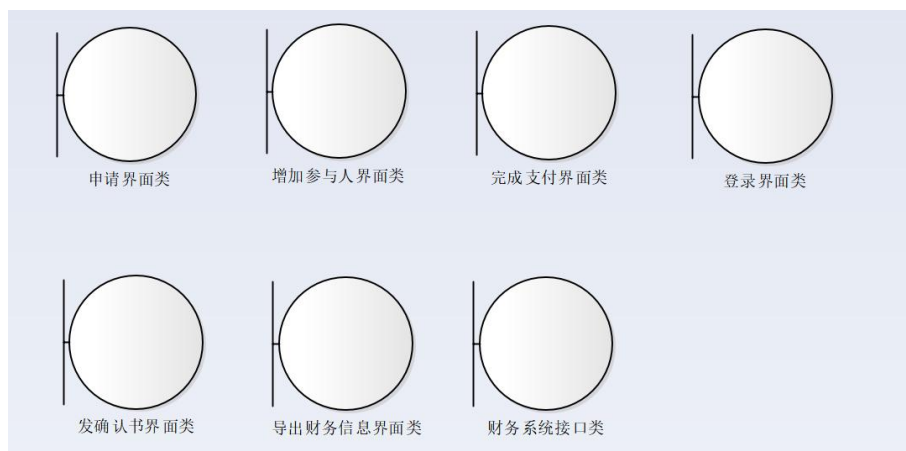
2.2. 双击主视图进行绘制。从工具箱中添加元素“Class”，将其命名为“申请界面类”，构造型设置为“boundary”，点击确定，完成创建。



得到“申请边界类”如下：



2.3. 同理于上述操作，绘制所有的边界类如下图所示。

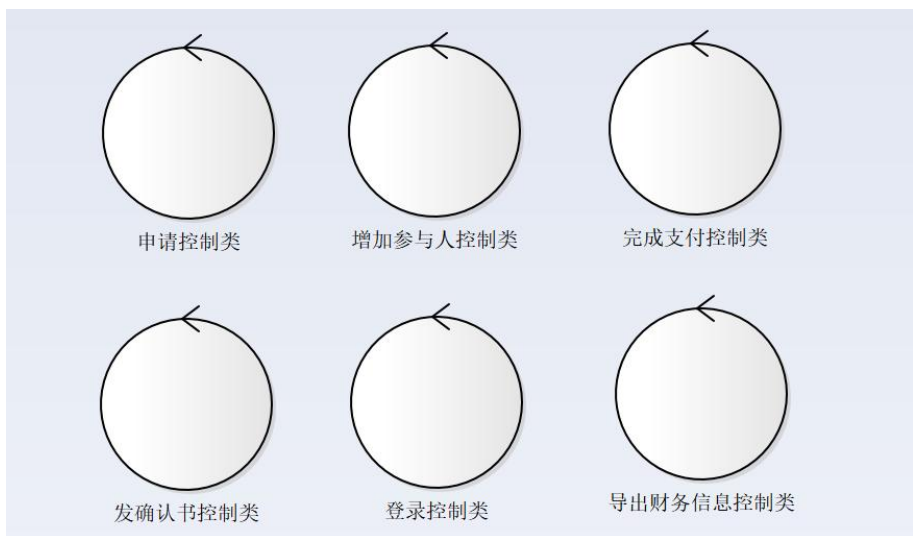


得到“《layer》Boundary Layer”包的结构如下：



2.4. 同理于上述绘图过程，在“《layer》Control Layer”包中新建主视图，并绘制所有构

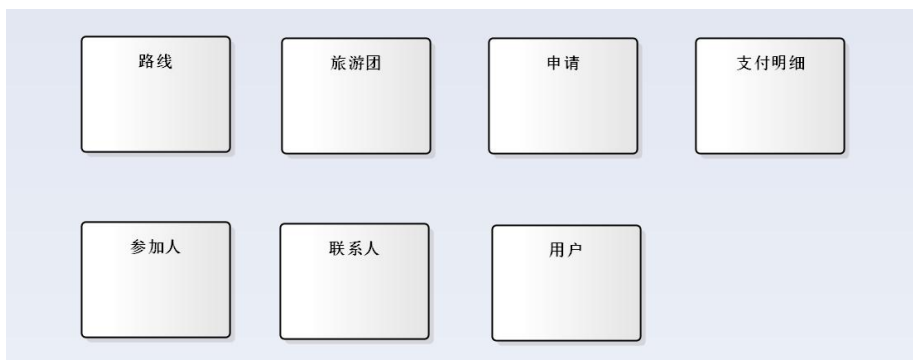
造型为“control”的控制类如下图所示：



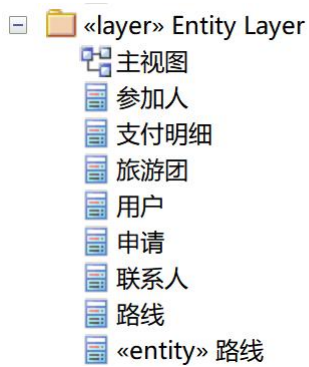
得到“《layer》Control Layer”包的结构如下：



2.5. 同理于上述绘图过程，在“《layer》Entity Layer”包中新建主视图，并绘制所有控制类如下图所示：

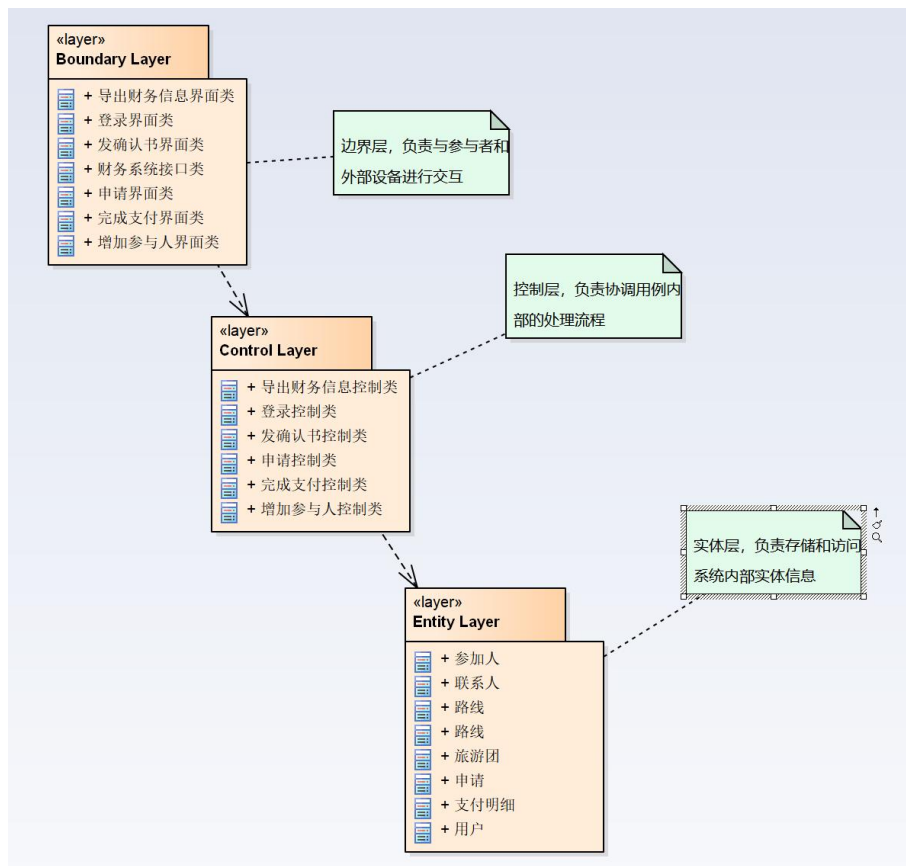


得到“《layer》Entity Layer”包的结构如下：



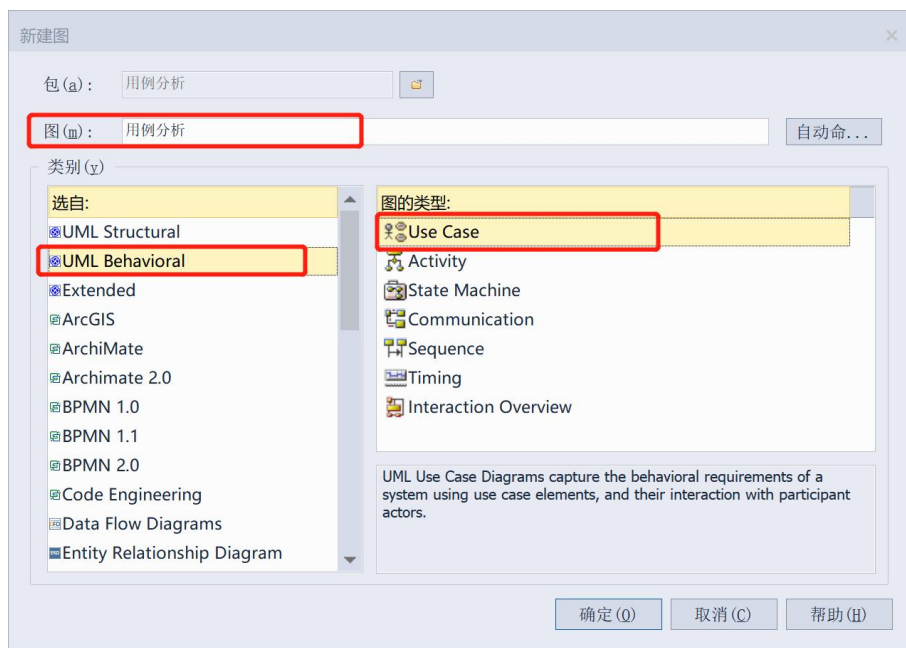
2.6. 完成上述 3 个分析类的绘制后，点开分析模型中的主视图，可以看到分析模型的包

已经得到细化。



练习 3：构造用例实现-绘制顺序图

3.1. 在分析模型中新建包，命名为“用例分析”。点击该包，选择新建图，命名为“用例分析”，类型选择“用例图”（即 UML Behavioral→Use Case）。

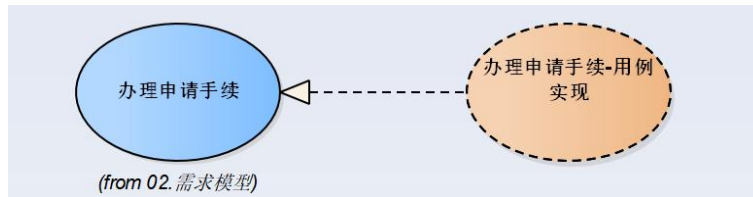


3.2. 双击打开上述创建的用例分析图，在项目浏览器中打开“需求模型”，选择用例“办理申请手续”将其拖入用例分析图。在弹出窗口的“放在图上作为”一栏选择“连接”，

点击确定，完成创建。



3.3. 从工具箱中选择元素“Collaboration”，将其命名为“办理申请手续-用例实现”，使用实现关系连接用例“办理申请手续”和它的用例实现。



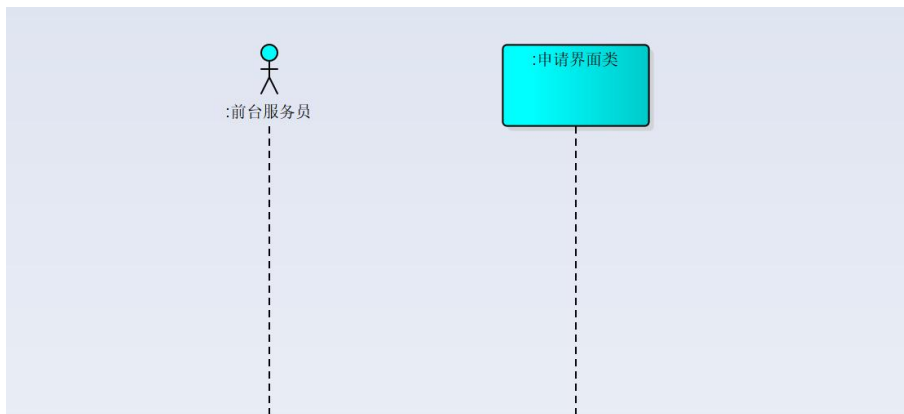
3.4. 右击上图 中“办理申请手续-用例实现”，选择“新建子图”，选择“交互元素-与顺序图”，新建复合元素名称为“基本场景”，点击确定，即完成“办理申请手续-用例实现”的顺序图。



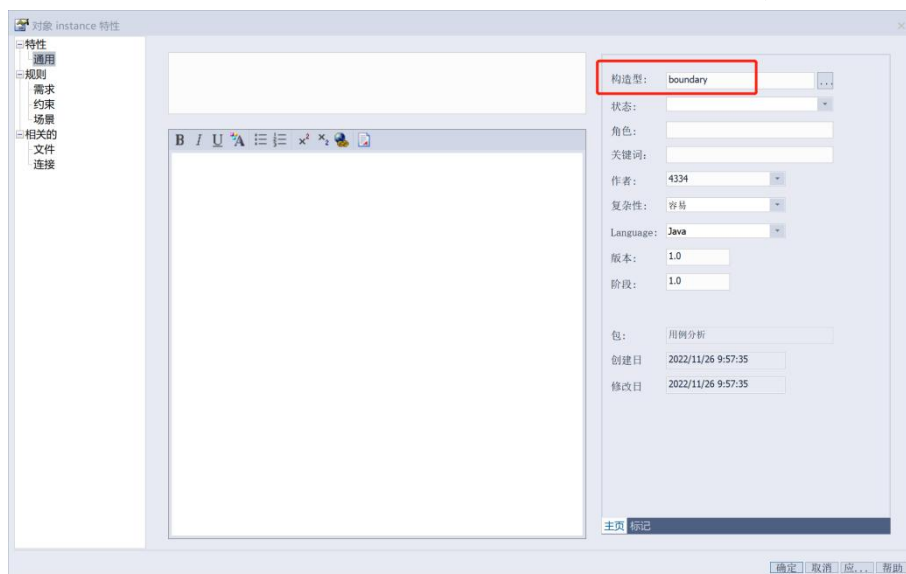
3.5. 打开“基本场景”，从“项目浏览器-需求模型”中，将参与者“前台服务员”拖入图中，并在弹出窗口中将“放在图上作为”设置为“生命线”，点击确定完成。



同理，将边界层“《layer》Boundary Layer”中的“申请界面类”拖入图中，同样选择生命线。



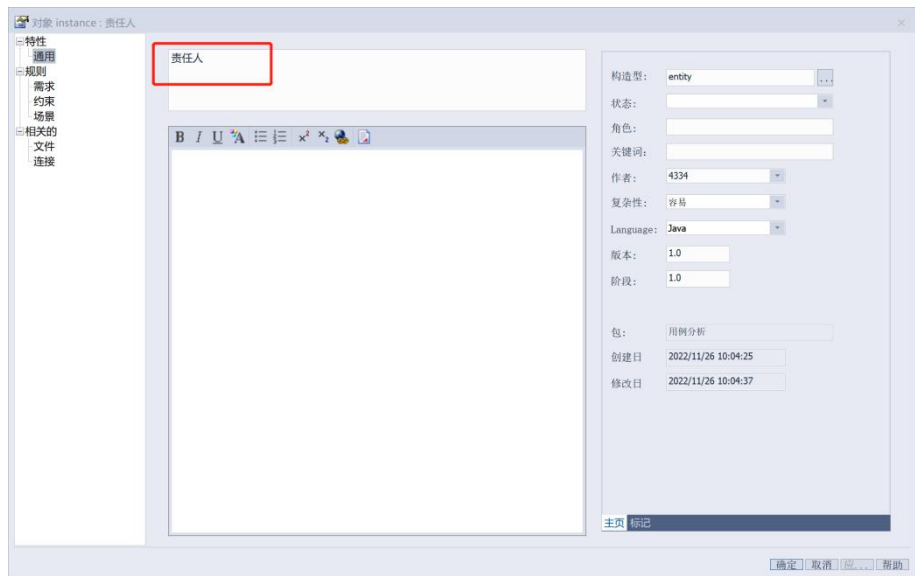
3.6. 双击“申请界面类”，打开特性窗口，在构造型中输入“boundary”



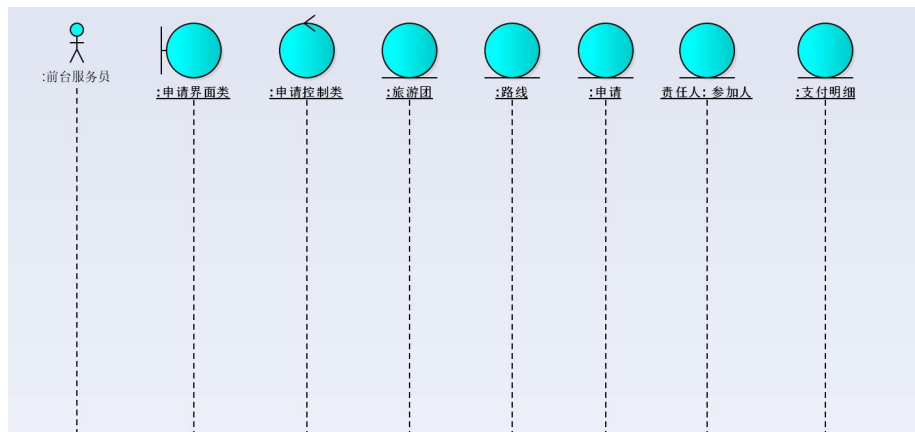
点击确定，得到以构造型特殊图形表示“基本场景”图如下图所示：



3.7. 同理，利用构造型“control”和“entity”完成顺序图里其他对象的绘制。其中，“参加人”类的对象具有对象名称“责任人”，在该对象的特性窗口中进行添加。



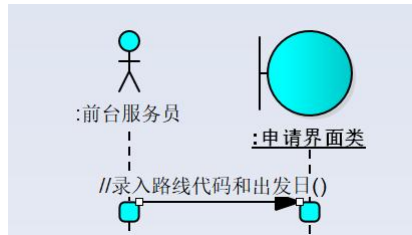
绘制的所有对象如下图所示：



3.8. 接下来，对顺序图中的消息进行绘制：

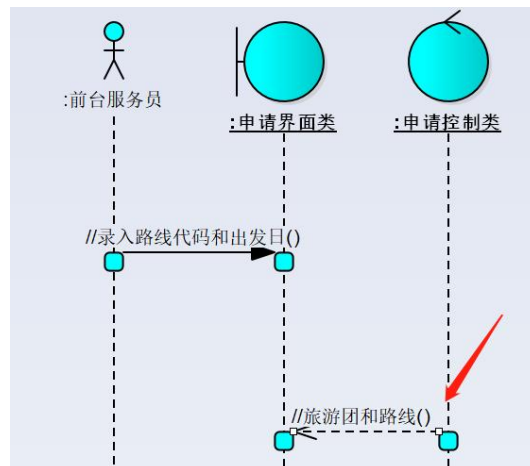
从工具箱中选择“Message”，在顺序图中，将“前台服务员”对象和“申请界面类”对象进行连接。双击消息箭头，在弹出的特性窗口的“消息”框中输入“//录入路线代码和出发日”，表示消息内容，点击确定完成。



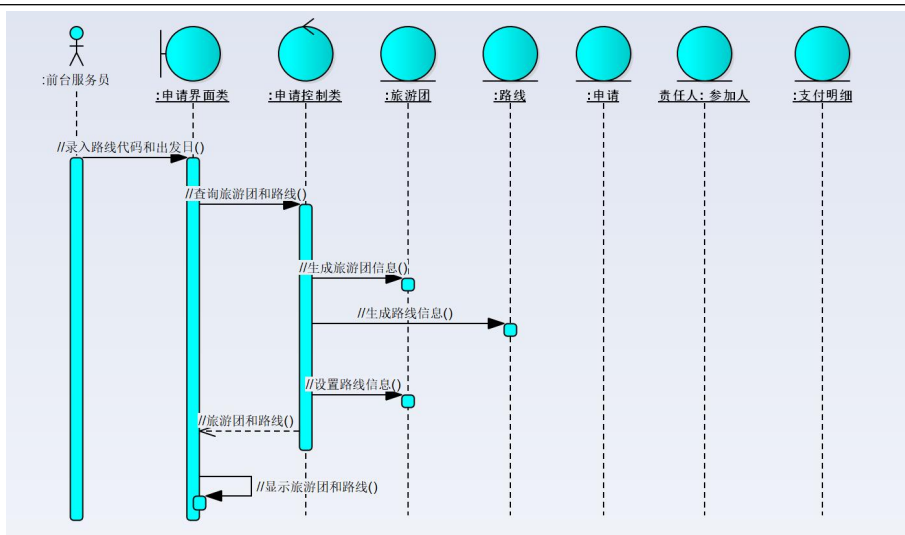


绘制“返回消息”：同理于上述操作绘制消息（连接“申请界面类”和“申请控制类”），然后在特性窗口中将“控制流类型”设置为“是返回”，点击确定即可创建“返回消息”（虚线表示）。

消息(s):	//旅游团和路线	操作
参数		
参数(u):		
返回值(V):	void	<input checked="" type="checkbox"/> 显示继承的方法(n)
分配给(g):		
构造型(p):		
别名(l):		
顺序表达式(E)		
条件(d):		
约束(r):		
<input type="checkbox"/> 重复(I)		
控制流类型(E):		
同步:	同步	生命周期:
类型:	调用	<input checked="" type="checkbox"/> 是返回

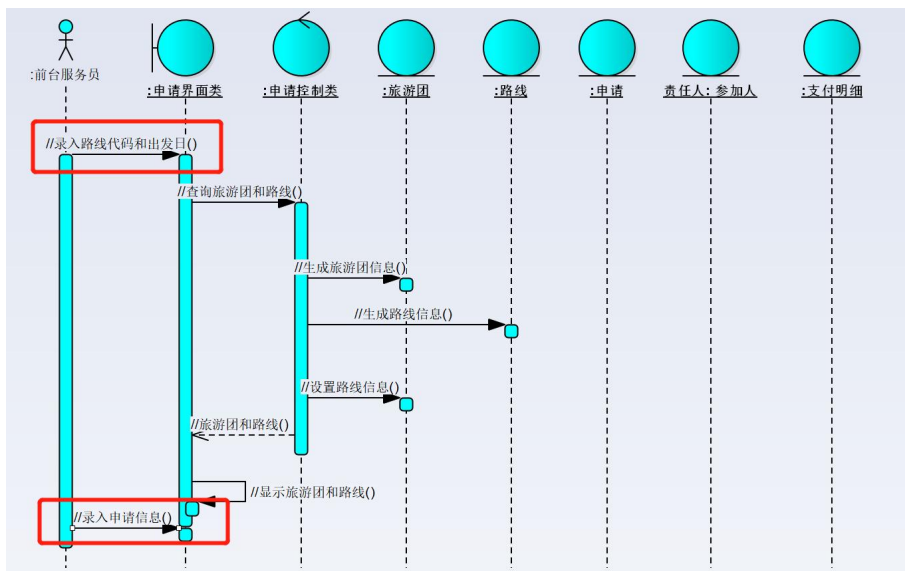


重复上述步骤，得到部分已添加消息的顺序图如下图所示：

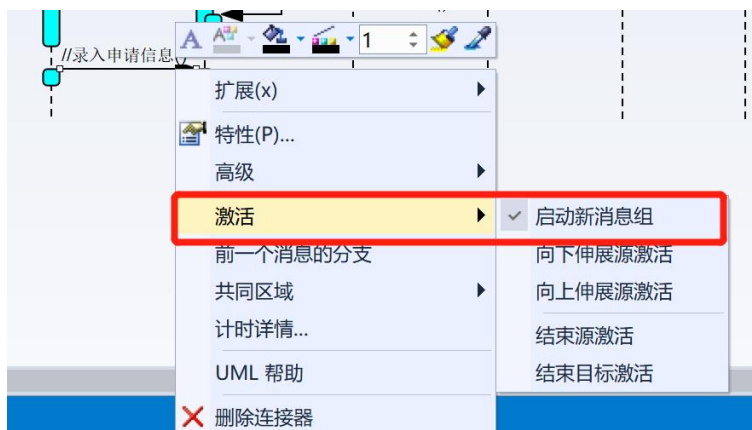


3.9. 接下来，绘制顺序图中的“执行”分段：

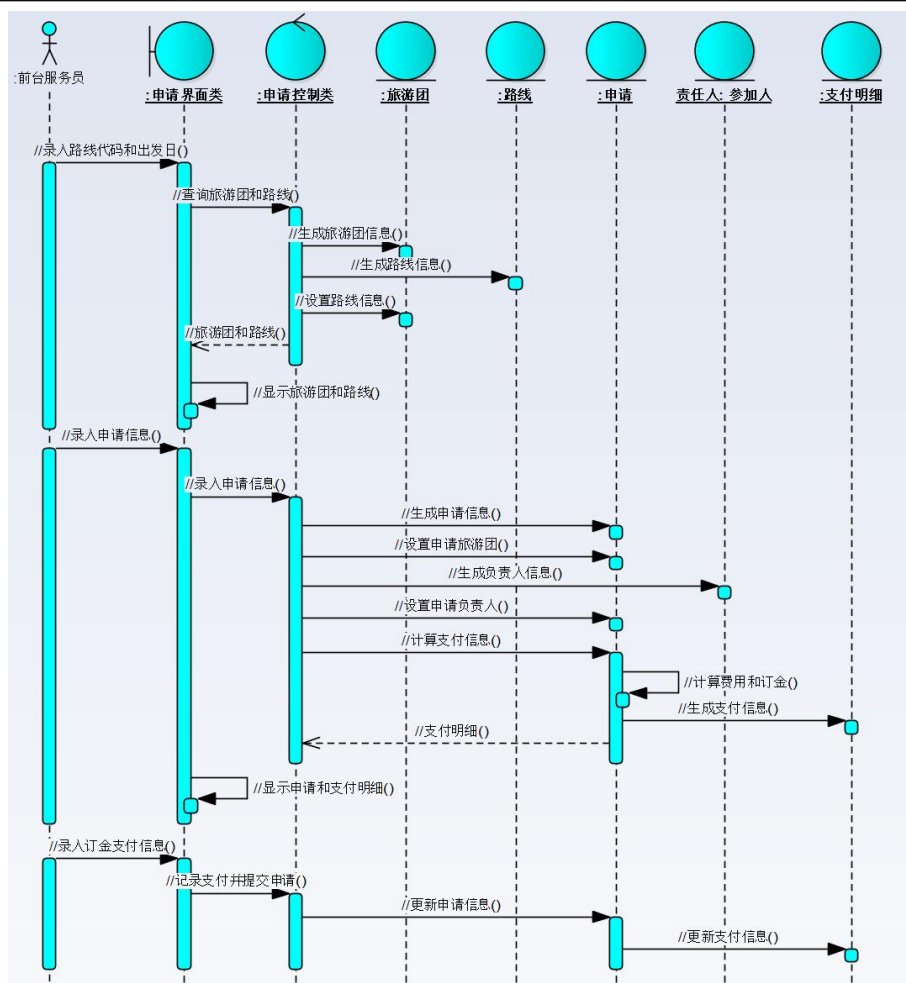
对于开启了新的交互的消息，需要将该“执行”进行分段。如下图中“录入路线代码和出发日”和“录入申请信息”属于不同的“执行”



则右击“录入申请信息”，选择“激活”→“启动新消息组”，完成“执行”分段。

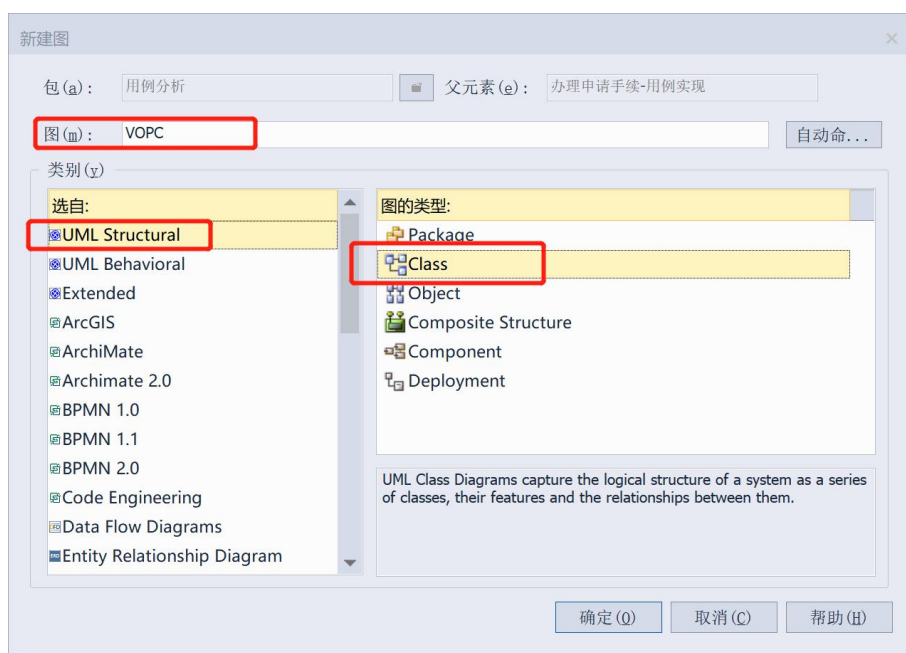


3.10 补全顺序图中所有消息并进行必要的“执行”分段，最终绘制效果图如图所示。

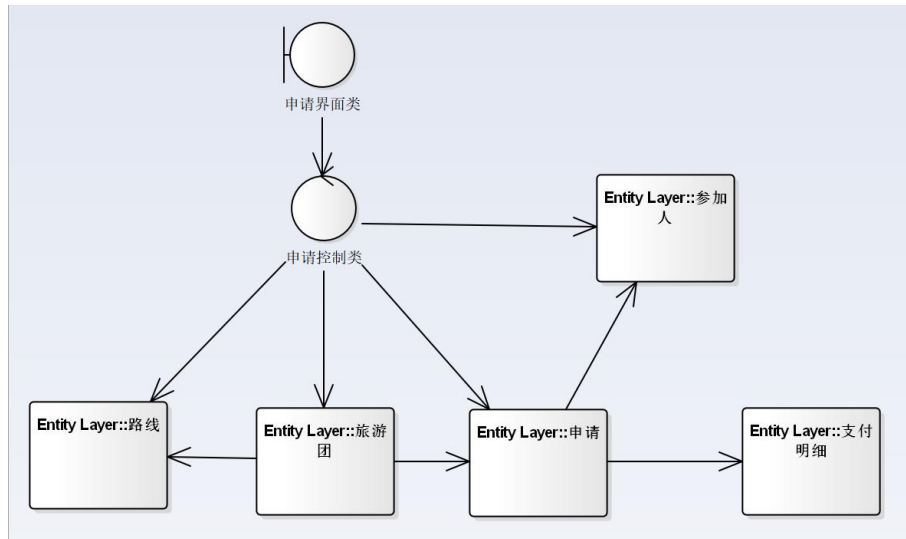


练习 4：构造用例实验-绘制 VOPC 类图

4.1. 在“项目浏览器”中右击用例实现“办理申请手续-用例实现”，选择“增加”→“添加图”，将其命名为“VOPC”，类型为“类图”（即 UML Structural-Class）。

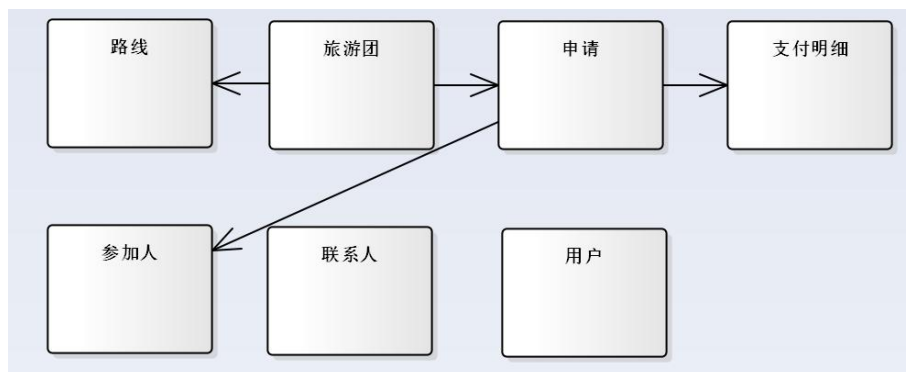


4.2. 将分析模型中分别在“Boundary Layer”、“Control Layer”以及“Entity Layer”中已定义的分析类拖入 VOPC 图中，并使用“关联”（Associate）关系连接每个类，最终绘制效果图如下图所示。

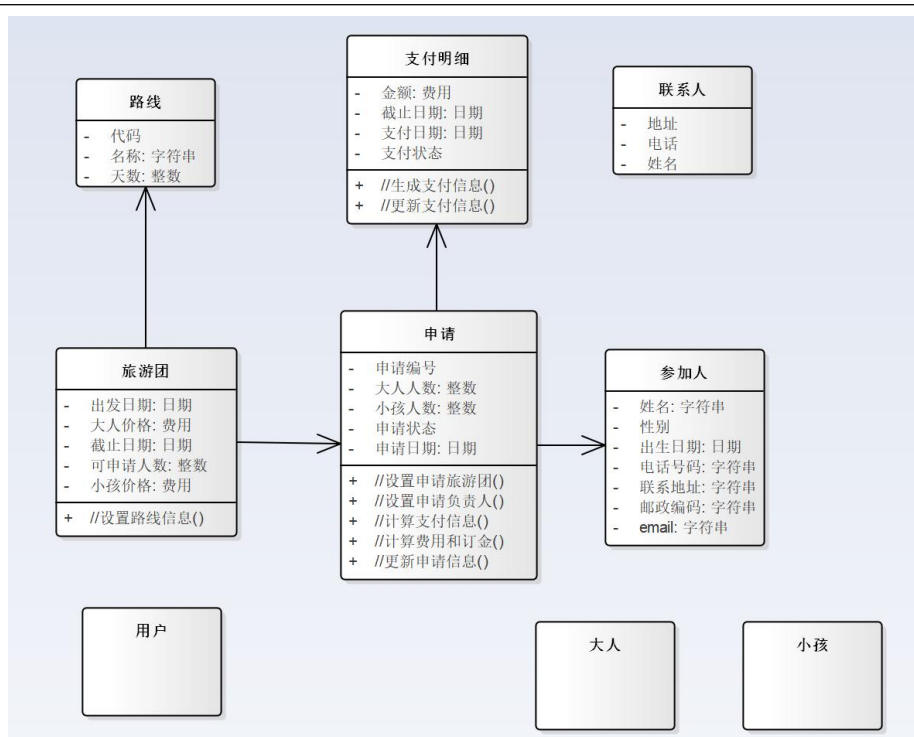


练习 5：完成实体类类图

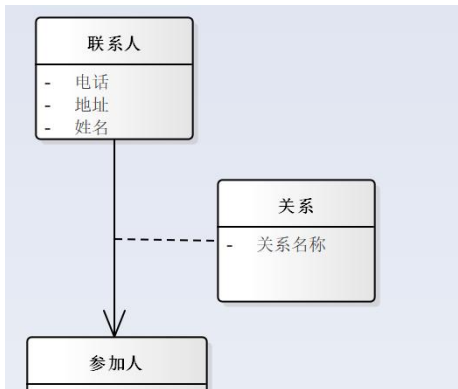
5.1. 经过练习 4 中 VOPC 图的绘制，在分析模型中“Entity Layer”主视图实体类关系如下图所示。



5.2. 为上图中的每一个实体类添加属性和操作，并添加“大人”和“小孩”两个实体类。

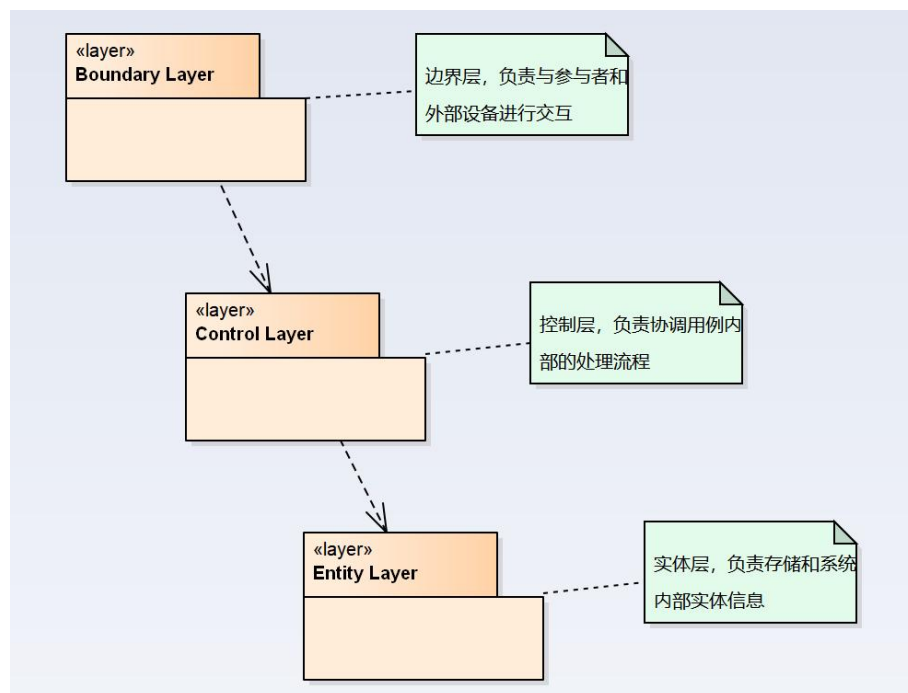


5.3. 对于“联系人”类与“参加人”类之间的关系，通过绘制“关联类”表示。在工具箱中选择关联类即“Association Class”，将“联系人”类与“参加人”类进行连接，将其命名为“关系”，并添加相关属性。



5.4. 关联关系端点名及多重性的指定：对于“路线”类上的自反关联关系，双击该关联关系，在弹出的特性窗口中选择“角色”，在“起始”和“目标”区域分别定义角色名以及“基数”。

打开“医院.cap”文件，选择“分析模型”，按照练习 1 的方法，建立系统备选分层架构 B-C-E 如下图所示：



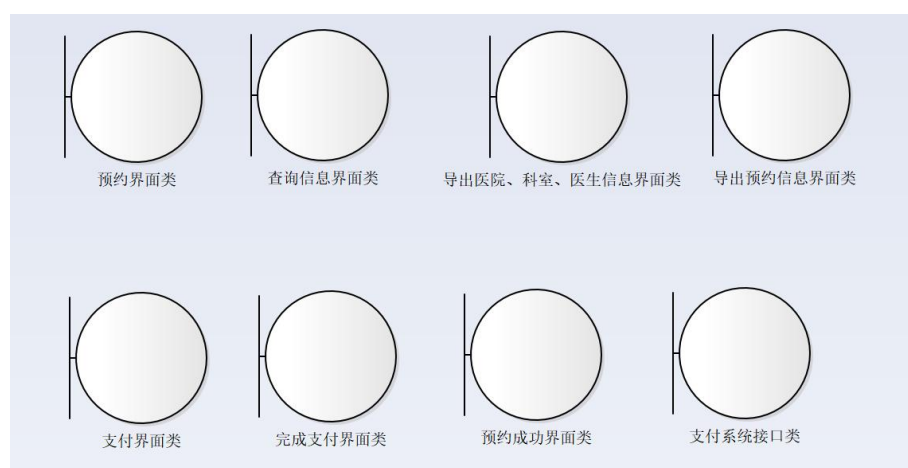
6.2. “预约挂号”用例-识别分析类

6.2.1. 边界类的识别

根据对象系统中“分析类”的定义和特点，以及“预约挂号”用例文档，可以知道用户在进行预约挂号的时候，需要通过“支付系统”进行支付，由此识别出一个边界类为“支付系统接口类”。

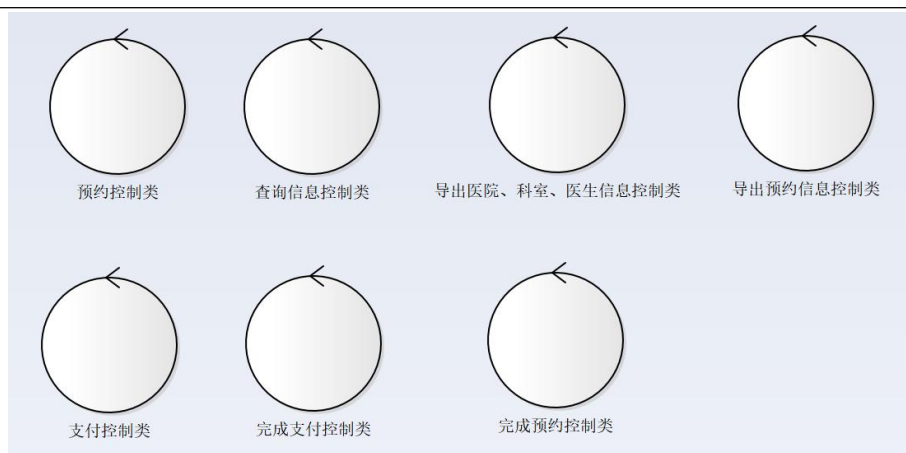
再根据用例基本事件流可以得到“边界类”应该包括“预约界面类”、“查询信息界面类”等系统以满足需求的共 7 个界面类（具体见附图所示）。

于是包括接口类在内的共 8 个边界类如下图所示：



6.2.2. 控制类的识别

根据上述识别的边界类，定义 7 个控制类分别用于封装相关用例业务流程和逻辑规则如下图所示：

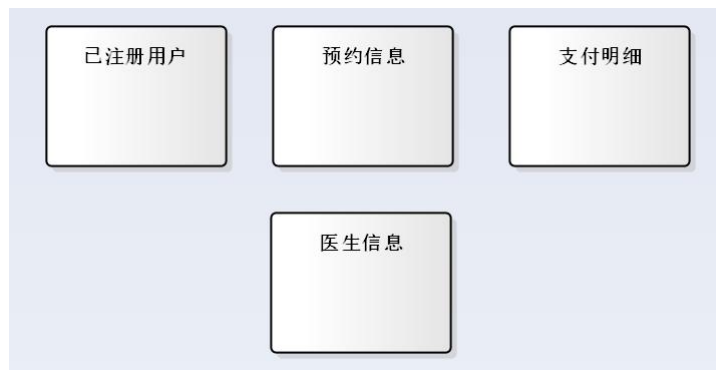


6.2.3. 实体类的识别

在系统分析中，实体类作为数据管理和业务逻辑处理层面上存在的类别，用于存储和管理系统内部的信息。

在“预约挂号”用例中，需要获取已注册用户得信息、录入存储用户的预约信息、支付信息，同时，由于系统对于医生的接诊人数有限制，在该过程中，用户预约信息会使某个医生的可接诊人数发生改变，故医生信息也需要进行存储。

根据实体类定义以及“预约挂号”系统中涉及的需要存储的信息，识别得 4 个实体类如下图所示：



6.3. “预约挂号”用例-构造用例实现（绘制顺序图和 VOPC 图）

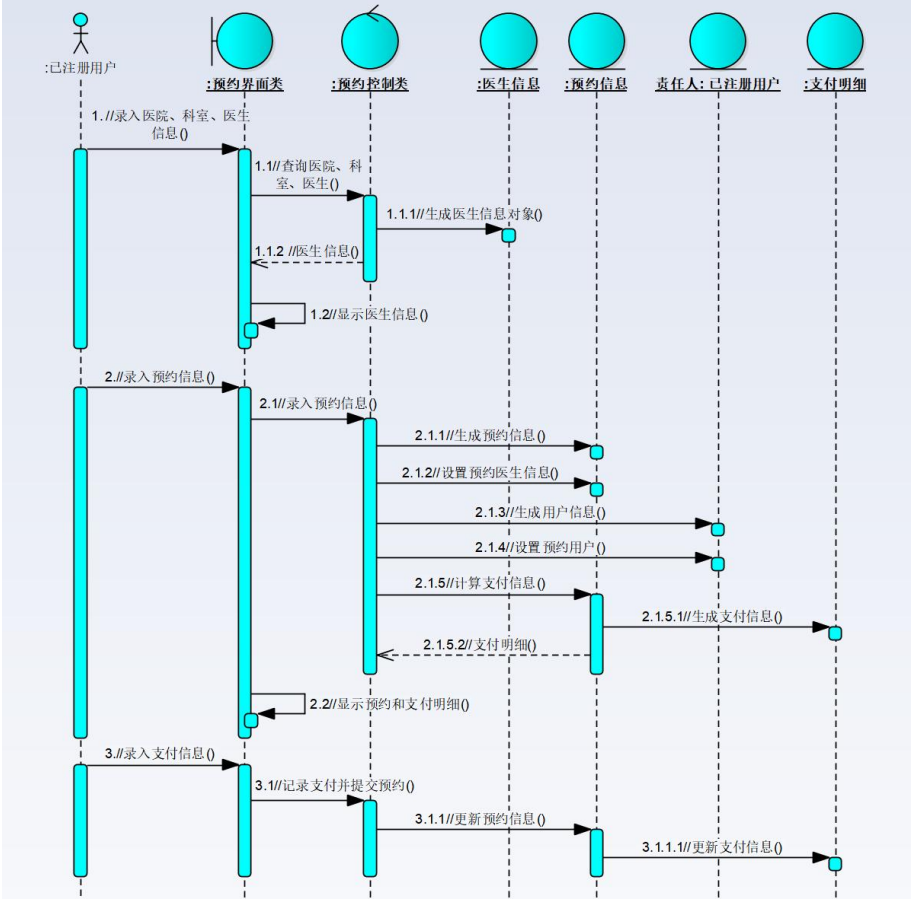
6.3.1. “预约挂号-用例实现”的基本场景顺序图描述的基本流程如下：

已注册用户首先在查询信息界面类录入医院、科室、医生信息（消息 1），界面类根据信息向控制类查询所要查询的医院、科室和医生信息（消息 1.1），控制类执行查询请求，根据查询结果生成相应的医生信息对象（创建消息 1.1.1），其中医生信息包含所在医院和科室，并返回该对象（返回消息 1.1.2）；界面对象接收到返回结果后，进行刷新，从而显示所查询到的医院、科室、医生信息（消息 1.2）（基本事件流 1~4 完成）。

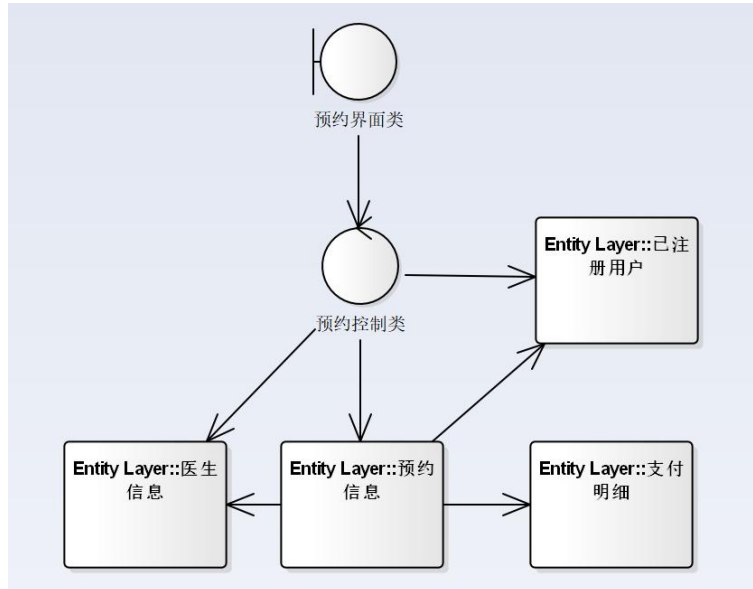
之后，用户确认预约该医院-科室-医生以及预约时间段，已注册用户向界面录入预约信息（消息 2），界面类将预约内容提交给控制类（消息 2.1），控制类针对预约信息交由相应的实体类进行处理：首先生成一个预约对象（消息 2.1.1），并与医生对象关联（消息 2.1.2），表明该预约信息所对应的医院、科室和医生；之后，生成一个参加人对象（消息 2.1.3），来存储预约的用户信息，并在预约对象中关联用户信息（消息 2.1.4）；最后，控制类要求预约对象计算本次预约有关的支付信息（消息 2.1.5），并生成支付明细对象来保存相应结果（消息 2.1.5.1），并将结果返回给控制类（消息 2.1.5.2）。控制对象将本次预约的明细返回给界面后，界面类进行刷新显示（消息 2.2）（基本事件流 5~6 完成）。

最后，用户根据系统计算出来的费用，通过支付界面类进行支付并录入支付信息（消息 3），界面将支付结果提交给控制类（消息 3.1），控制类根据支付结果更新预约对象的状态（消息 3.1.1），同时预约对象也会把支付情况记录到支付明细对象中（消息 3.1.1.1）（基本事件流 7~9 完成）。

根据练习 3 操作步骤以及上述流程描述，绘制得顺序图为：

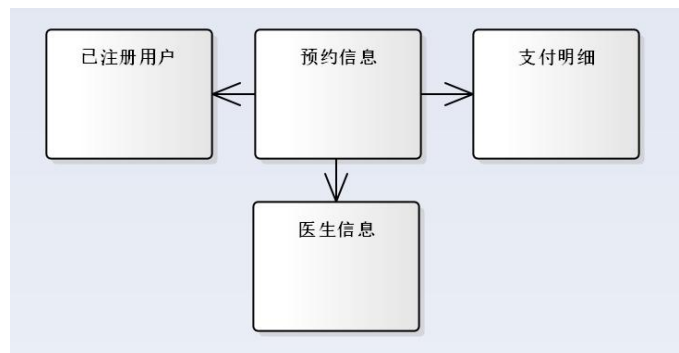


6.3.2. 按照练习 4-4.1 方式新建类图 VOPC 图，将分析模型中已定义的分析类拖入 VOPC 图中，并使用“关联”（Associate）关系连接每个类，最终绘制效果图如下图所示。

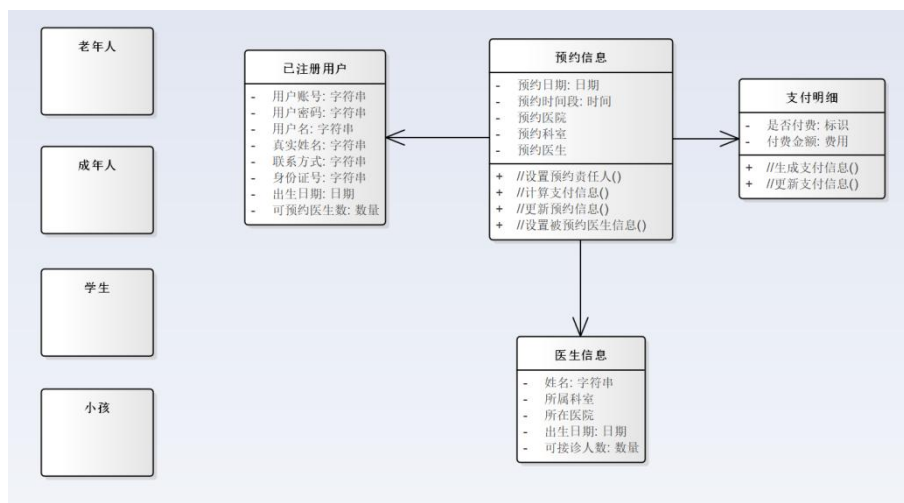


6.4. “预约挂号”用例-完成实体类类图

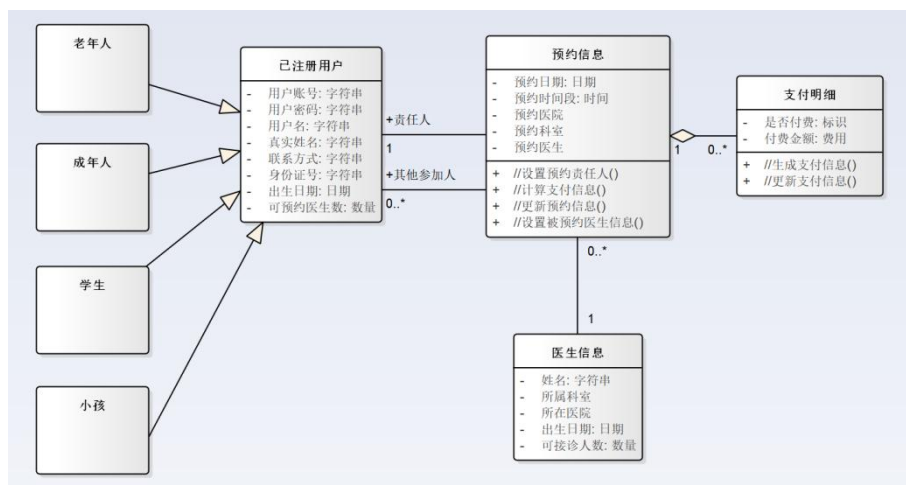
6.4.1. 经过上述 VOPC 图的绘制，我们可以看到“Entity Layer”的主视图的实体类关系发生改变如下：



6.4.2. 对上图实体类增加属性和操作，其中，对于用户以及医生的年龄信息，为方便数据库管理，添加属性“出生日期”，通过计算得到年龄信息；对于已注册用户的不同身份，可以添加“老年人”、“成年人”、“学生”、“小孩”四个实体类。得到增加属性和操作后的类图如下图所示：



6.4.3. 进一步对类之间的关系进行完善，进行关联关系端点名及多重性的指定。完成图中所有关系，得到最终效果图如下：



实验结论：

通过本次实验，学习了如何在 EA 中定义系统备选架构、识别分析类、构造分析模型中的用例实现等用例分析步骤；学会如何根据系统案例描述对指定系统以 B-C-E 架构对分析模型进行组织，对用例进行用例分析，进一步完善案例建模。

指导教师批阅意见：

成绩评定：

指导教师签字：

年 月 日

备注：

注：1、报告内的项目或内容设置，可根据实际情况加以调整和补充。

2、教师批改学生实验报告时间应在学生提交实验报告时间后 10 日内。