

北京邮电大学

Beijing University of Posts and Telecommunications

计算机学院（国家示范性软件学院）

软件工程



《静态结构设计》

304c 组		
姓名	班级	学号
谢睿	2018211304	2018211208
王倩	2018211304	2018211201
黄博伟	2018211304	2018211217
赵照	2018211304	2018211230
葛朔	2018211304	2018211242

2021 年 5 月 31 日

目录

○. 系统分层结构说明.....	3
0.1. 架构概述.....	3
0.2. 架构图示.....	3
一. 基于用例的静态结构.....	4
1.1 用例_1 使用空调.....	4
1.2 用例_2 空调调度.....	5
1.3 用例_3 管理空调.....	6
1.4 用例_4 账单生成.....	7
1.5 用例_5 查询报表.....	8
二. 系统级静态结构.....	9
三. 软件类的说明.....	9
3.1 用户界面层.....	9
3.1.1 consumer_window.....	9
3.1.2 reception_window.....	10
3.3.3 administrator_window.....	10
3.1.4 boss_window.....	11
3.2 控制器层.....	11
3.2.1 consumer_controller.....	11
3.2.2 reception_controller.....	12
3.3.3 administrator_controller.....	13
3.3.4 boss_controller.....	13
3.3 业务逻辑层.....	14
3.3.1 Scheduler.....	14
3.3.2 Service_Queue.....	14
3.3.3 Waitting_Queue.....	15
3.3.4 AirConditioner.....	16
3.3.5 Invoice.....	16
3.3.6 MonitoredObject.....	17
3.3.7 MainAirConditioner.....	17
3.3.8 Report.....	17
3.3.9 RecordDetail.....	17
3.3.10 Service.....	18
3.4 持久化层.....	18
3.4.1 DBFacade.....	18
3.5 数据库 SQLite.....	19
3.6 系统管理类.....	19
3.6.1 clientthread.....	19
四. 附录.....	20
4.1 系统级静态结构图.....	20

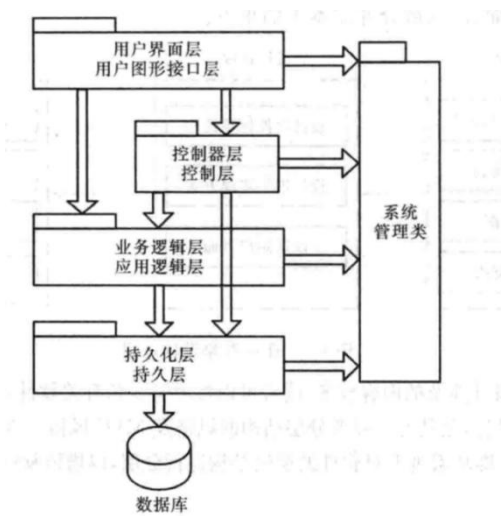
○. 系统分层结构说明

0.1. 架构概述

本系统采用了与课本相同的软件架构，分别是用户界面层、控制器层、应用层、持久化层和数据库；除此之外还有系统管理类。

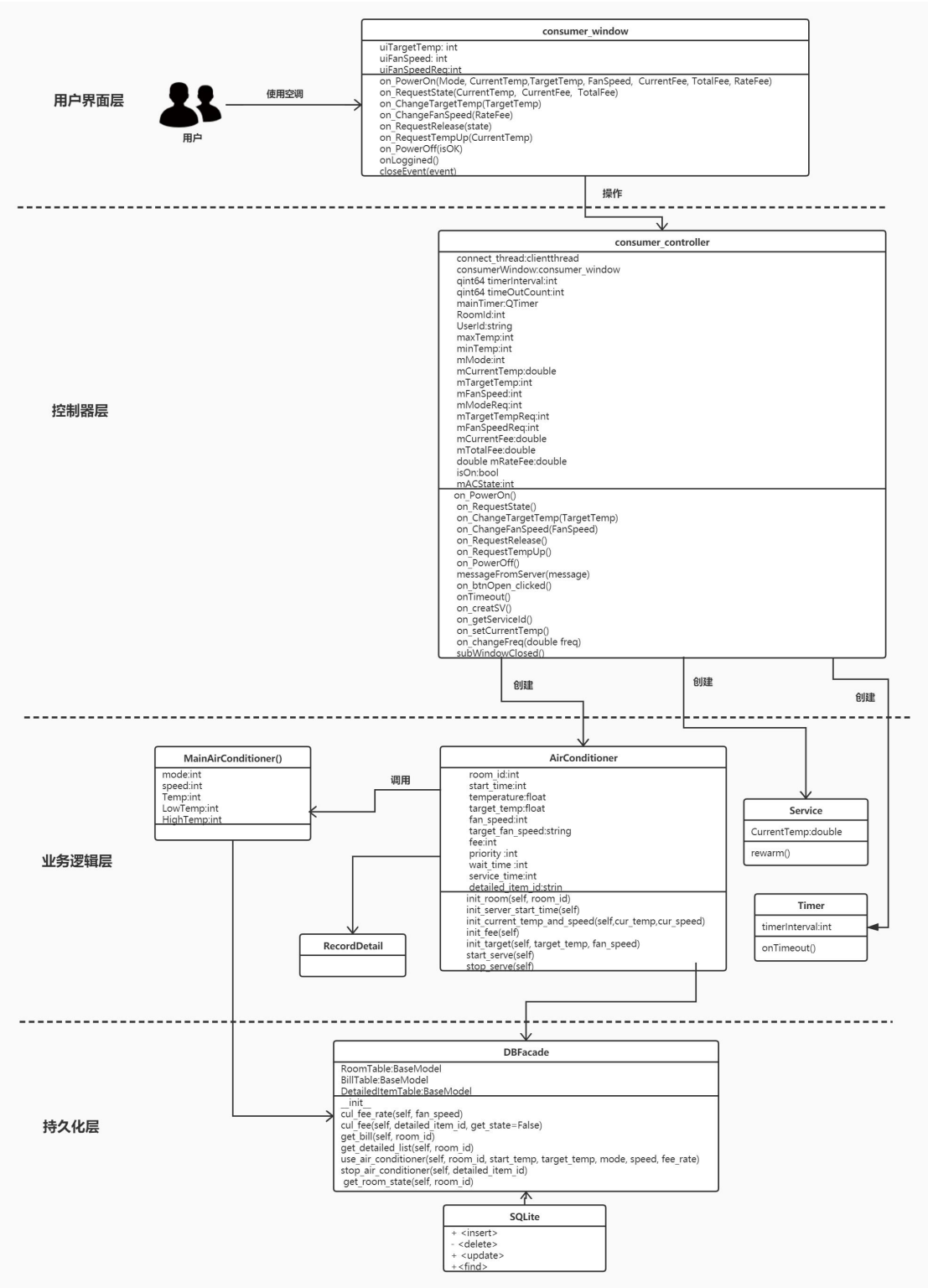
系统管理类实现软件的基本特性，在本系统中，其包含用于通信的类。

0.2. 架构图示

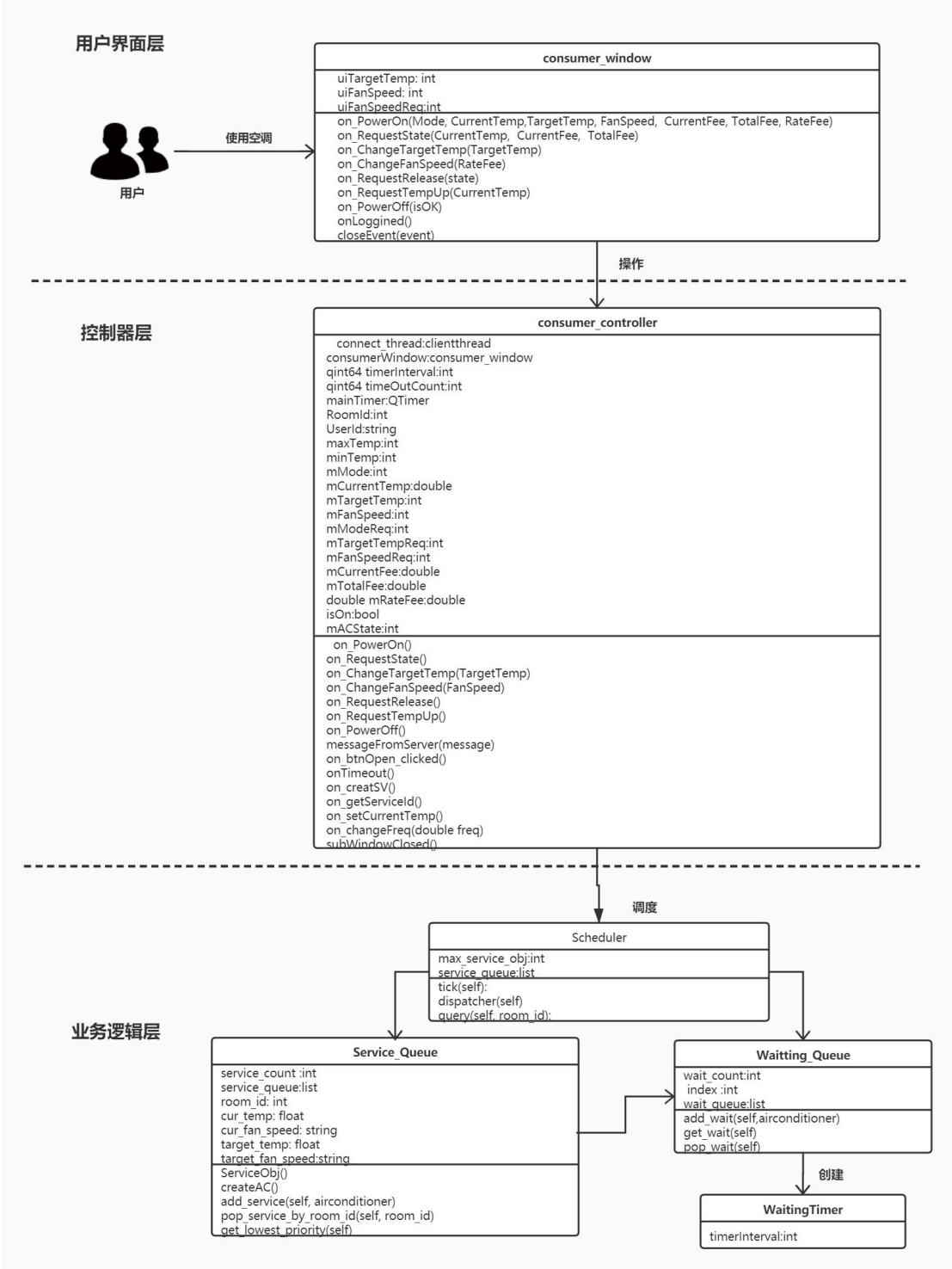


一. 基于用例的静态结构

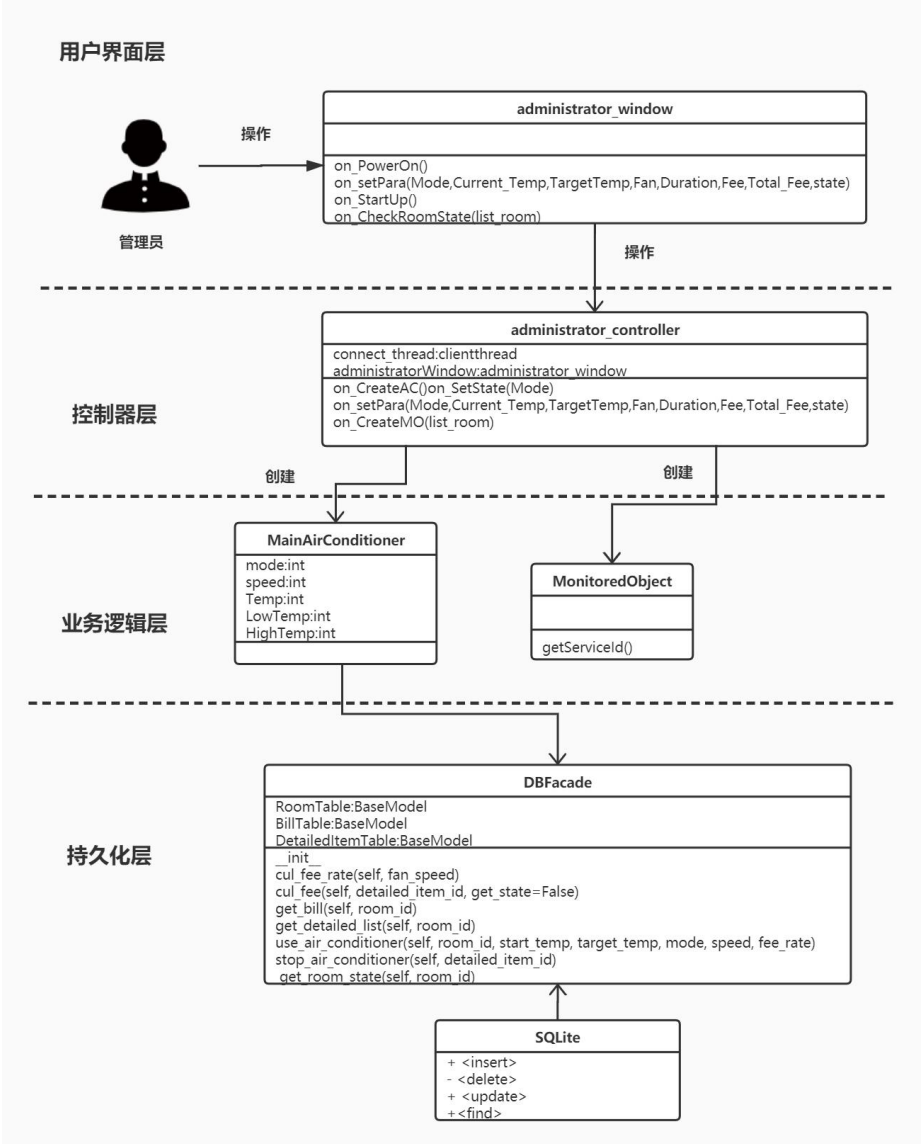
1.1 用例_1 使用空调



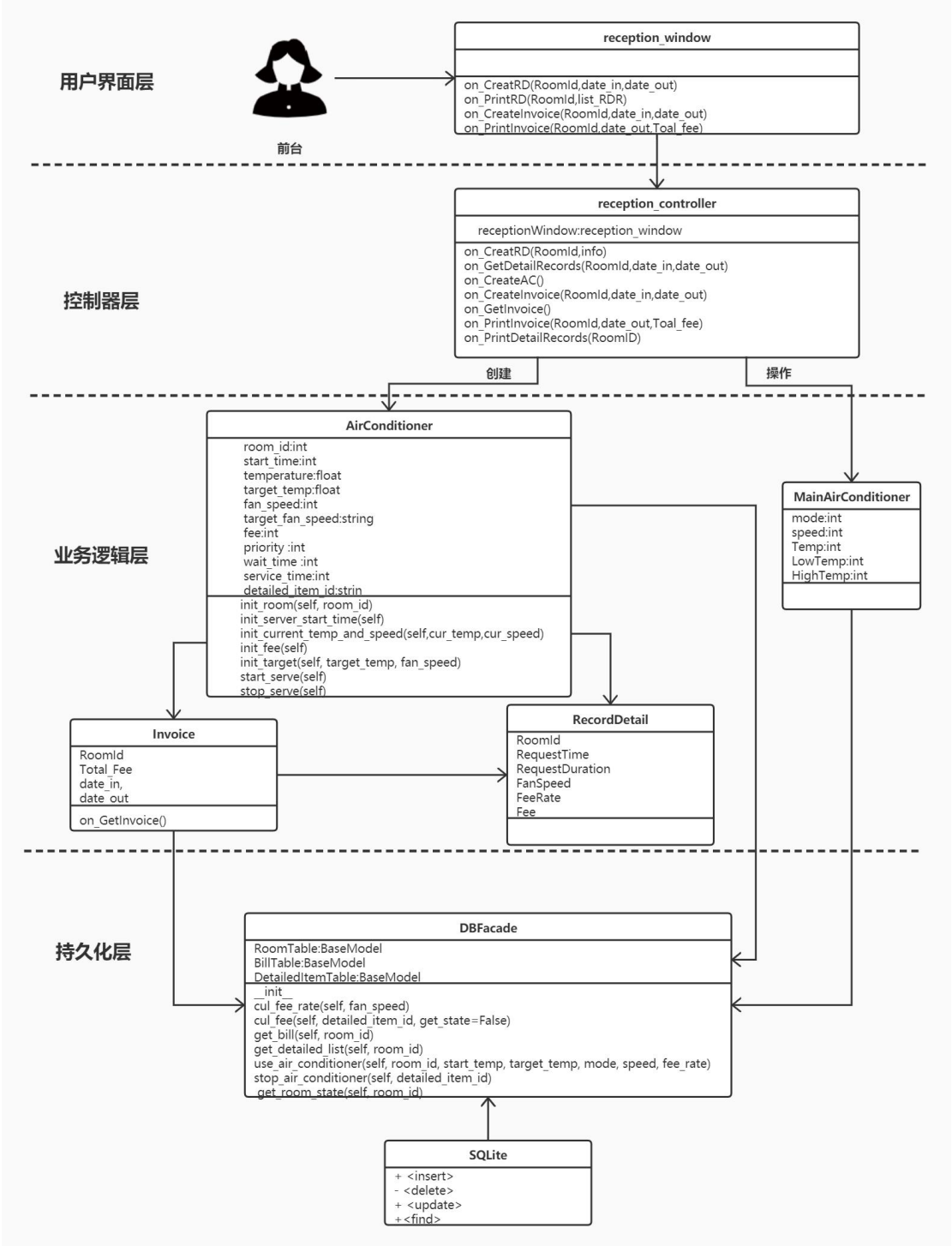
1.2 用例_2 空调调度



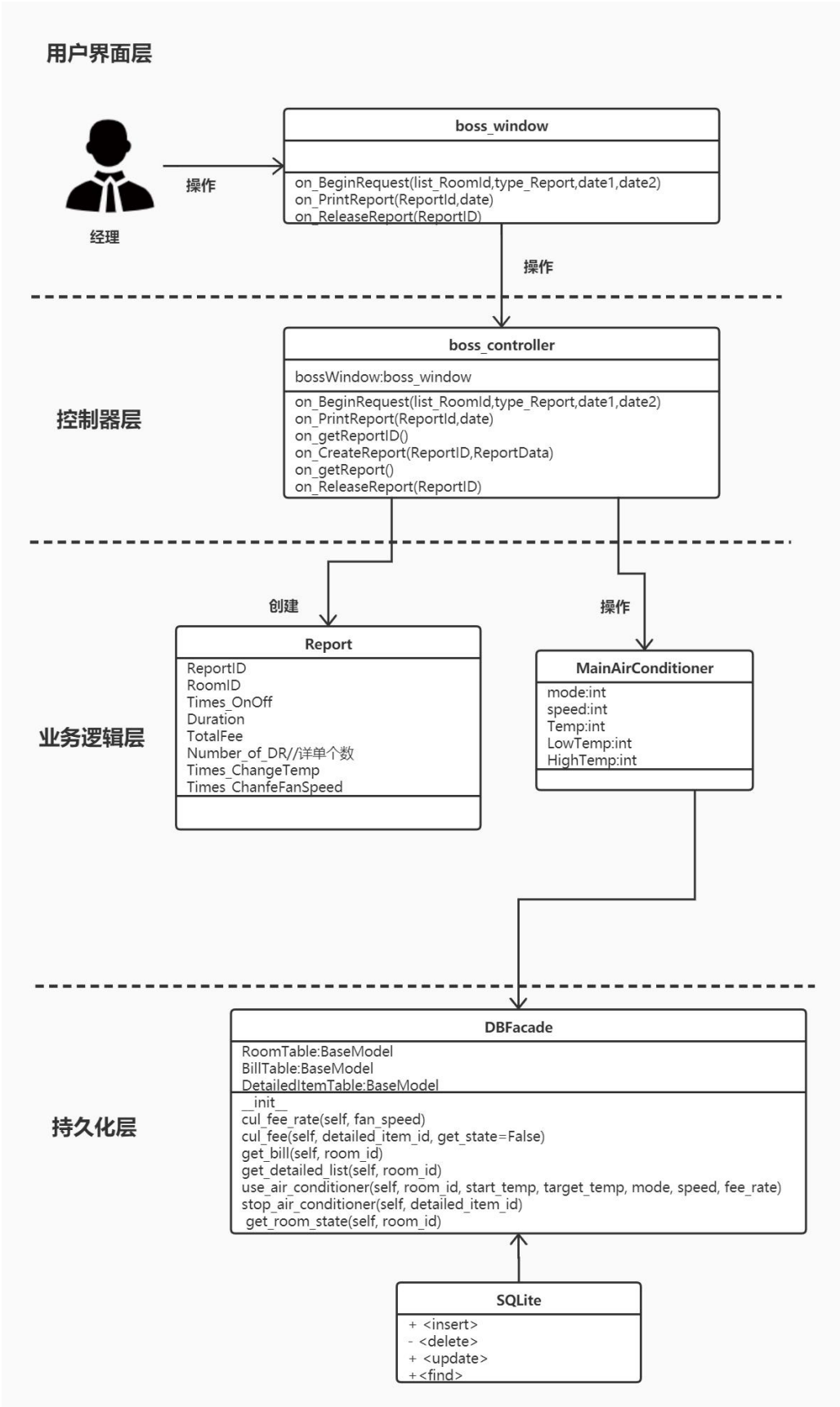
1.3 用例_3 管理空调



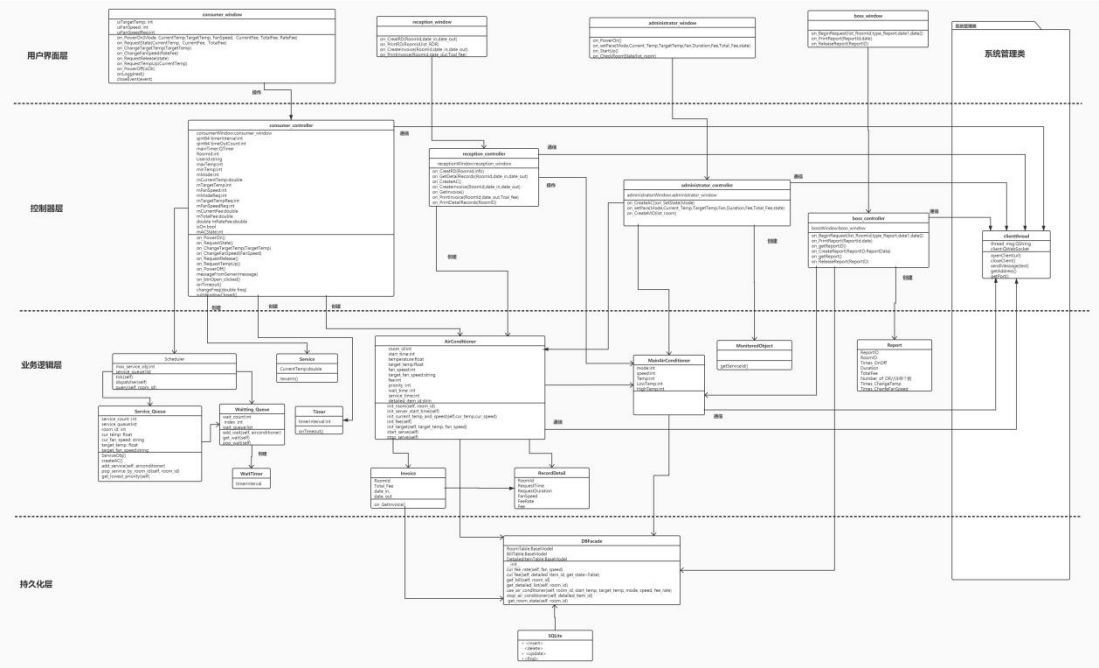
1.4 用例_4 账单生成



1.5 用例_5 查询报表



二. 系统级静态结构



上图较为复杂，我们在文档的附录内附上了系统级静态结构图的清晰版本：
[系统级静态结构图\(高清版\)](#)

三. 软件类的说明

3.1 用户界面层

3.1.1 consumer_window

属性	类型	说明
uiTargetTemp	int	目标温度
uiFanSpeed	int	风速
uiFanSpeedReq	int	风速请求
方法名称	参数	说明
on_PowerOn(Mode, CurrentTemp, TargetTemp, FanSpeed, CurrentFee, TotalFee, RateFee)	Mode:int CurrentTemp:double TargetTemp:int FanSpeed:int CurrentFee:double TotalFee:double RateFee:double 无返回值	开机
on_RequestState(CurrentTemp, CurrentFee, TotalFee)	CurrentTemp:double CurrentFee:double	请求状态

	TotalFee:double 无返回值	
on_ChangeTargetTemp(TargetTemp)	TargetTemp:int 无返回值	更改目标温度
on_ChangeFanSpeed(RateFee)	RateFee:int 无返回值	更改风速
on_RequestRelease(state)	State:int 无返回值	达到目标温度
on_RequestTempUp(CurrentTemp)	CurrentTemp:double 无返回值	回温
on_PowerOff(isOK)	isOK:double 无返回值	关机
onLoggedIn()	无返回值	窗口打开事件
closeEvent(event)	event:QCloseEvent 无返回值	窗口关闭事件

3.1.2 reception_window

方法名称	参数	说明
on_CreatRD(RoomId,date_in,date_out)	RoomId:int date_in:int date_out:int 无返回值	创建详单
on_PrintRD(RoomId,list_RDR)	RoomId:int list_RD:list 无返回值	打印详单
on_CreateInvoice(RoomId,date_in,date_out)	RoomId:int date_in:int date_out:int 无返回值	创建账单
on_PrintInvoice(RoomId,date_out,Toal_fee)	RoomId:int date_out:int Toal_fee:double 无返回值	打印账单

3.3.3 administrator_window

方法名称	参数	说明
on_PowerOn()	无返回值	开机
on_setPara(Mode,Temp_highLimit,Temp_lowLimit,default_TargetTemp,FreeRate_H,FreeRate_M,FreeRate_L)	Mode:int Temp_highLimit:int Temp_lowLimit:int	设置默认参数

	default_TargetTemp: int FreeRate_H:double FreeRate_M:double FreeRate_L:double 无返回值	
on_StartUp()	无返回值	修改空调对象 状态
on_CheckRoomState(list_room)	list_room:list<int> 无返回值	监控房间

3.1.4 boss_window

方法名称	参数	说明
on_BeginRequest(list_RoomId,type_Report, date1,date2)	list_RoomId:int type_Report:int date1:int date2:int 无返回值	报表查询请求
on_PrintReport(ReportId,date)	ReportId:int date:int 无返回值	打印报表
on_ReleaseReport(ReportID)	ReportID:int 无返回值	释放报表

3.2 控制器层

3.2.1 consumer_controller

属性	类型	说明
consumerWindow	consumer_window	界面
timerInterval	int	时间间隔
timeOutCount	int	定时器超时次数
mainTimer	QTimer	主定时器
RoomId	int	房间号
UserId	string	用户 ID
maxTemp	int	最高温度
minTemp	int	最低温度
mMode	int	模式
mCurrentTemp	double	当前温度
mTargetTemp	int	目标温度

mFanSpeed	int	风速
mModeReq	int	模式请求
mTargetTempReq	int	目标温度请求
mFanSpeedReq:	int	风速请求
mCurrentFee	double	当前费用
mTotalFee	double	总费用
mRateFee	double	费率
isOn	bool	状态
mACState	int	空调对象状态

方法名称	参数	说明
on_PowerOn()	无返回值	开机
on_RequestState()	无返回值	请求状态
on_ChangeTargetTemp(TargetTemp)	TargetTemp:int 无返回值	改变温度
on_ChangeFanSpeed(FanSpeed)	FanSpeed:int 无返回值	改变风速
on_RequestRelease()	无返回值	达到目标温度
on_RequestTempUp()	无返回值	回温
on_PowerOff()	无返回值	关机
messageFromServer(message)	message:string 无返回值	服务器传来消息
on_btnOpen_clicked()	无返回值	打开按钮
onTimeout()	无返回值	定时器超时处理
changeFreq(double freq)	无返回值	定时器间隔改变
subWindowClosed()	无返回值	子界面关闭

3.2.2 reception_controller

属性	类型	说明
receptionWindow	reception_window	与 UI 交互
方法名称	参数	说明
on_CreatRD(RoomId,info)	RoomId:int info:string 无返回值	创建详单
on_GetDetailRecords(RoomId,date_in,date_out)	RoomId:int date_in:int	获得详单

	date_out:int 无返回值	
on_CreateAC()	无返回值	创建空调对象
on_CreateInvoice(RoomId,date_in,date_out)	RoomId:int date_in:int date_out:int 无返回值	创建账单
on_GetInvoice()	无返回值	获得账单
on_PrintInvoice(RoomId,date_out,Toal_fee)	RoomId:int date_out:int Total_fee:double 无返回值	打印账单
on_PrintDetailRecords(RoomID)	RoomID:int	打印详单

3.3.3 administrator_controller

属性	类型	说明
administratorWindow	administrator_window	与界面交互
方法名称	参数	说明
on_CreateAC()	无返回值	创建空调对象
on_SetState(state)	state:int 无返回值	设置状态
on_setPara(Mode,Temp_highLimit,Temp_lowLimit,default_TargetTemp,FreeRate_H,FreeRate_M,FreeRate_L)	Mode:int Temp_highLimit:int Temp_lowLimit:int default_TargetTemp:int FreeRate_H:double FreeRate_M:double FreeRate_L:double 无返回值	设置默认参数
on_CreateMO(list_room)	list_room:list<int> 无返回值	创建监控对象

3.3.4 boss_controller

属性	类型	说明
bossWindow	boss_window	与 UI 交互
方法名称	参数	说明

on_BeginRequest(list_RoomId,type_Report,date1,date2)	list_RoomId:int type_Report:int date1:int date2:int 无返回值	报表查询请求
on_PrintReport(ReportId,date)	ReportId:int date:int 无返回值	打印报表
on_getReportID()	返回值为 int	获得报表号
on_CreateReport(ReportID,ReportData)	ReportID:int ReportData:json	创建报表
on_getReport()	返回值:string	获取报表
on_ReleaseReport(ReportID)	ReportID:int 无返回值	释放报表

3.3 业务逻辑层

3.3.1 Scheduler

属性	类型	说明
max_service_obj	int	最大服务能力
service_queue	list	服务队列
方法名称	参数	说明
tick(self)	self 无返回值	更新相关时间的量
dispatcher(self)	self 无返回值	调度逻辑
query(self, room_id)	self room_id:int 无返回值	查询房间状态

3.3.2 Service_Queue

属性	类型	说明
service_count	int	服务数量
service_queue	list	服务队列
room_id	int	房间号
cur_temp	float	当前温度
cur_fan_speed	string	当前风速
target_temp	float	目标温度

target_fan_speed	string	目标风速
方法名称	参数	说明
ServiceObj()	无返回值	
createAC()	返回:AirConditioner	创建空调对象
add_service(self, airconditioner)	self airconditioner:AirConditioner 返回值:bool	将服务对象加入服务队列
pop_service_by_room_id(self, room_id: int)	self room_id: int 返回值:AirConditioner	通过房间号从服务队列取出服务对象(应对关机请求)
get_lowest_priority(self)	self 返回值:AirConditioner	从服务队列中找出优先级最低的服务对象, 若存在多个相同最低优先级, 则取出服务时间最长的(应对时间片轮转调度)

3.3.3 Waitting_Queue

属性	类型	说明
wait_count	int	等待数量
index	int	索引
wait_queue	list	等待队列
方法名称	参数	说明
add_wait(self, airconditioner)	self airconditioner:Airconditioner 返回值:bool	将服务对象加入等待队列
get_wait(self)	self 返回值:Airconditioner	从等待队列中取出优先级最高的服务对象
pop_wait(self)	self 返回值:Airconditioner	从等待/待调度队列弹出优先级最高服务对象

3.3.4 AirConditioner

属性	类型	说明
room_id	int	房间号
start_time	int	开始时间
temperature	float	温度
target_temp	float	目标温度
fan_speed	int	风速
target_fan_speed	string	目标风速
fee	int	费用
priority	int	优先级
wait_time	int	等待时间
service_time	int	服务时间
detailed_item_id	int	详单号
方法名称	参数	说明
init_room(self, room_id)	self room_id:int 返回值:bool	给房间号赋值
init_server_start_time(self)	Self 返回值:bool	初始化服务开始时间
init_current_temp_and_speed(self,cur_temp,cur_speed)	self cur_temp:double cur_speed:int 返回值:bool	初始化当前房间温度与当前风速
init_fee(self)	self 返回值:bool	重新初始化费用
init_target(self, target_temp, fan_speed)	self target_temp:double fan_speed:int 返回值:bool	设置目标温度与风速,并根据风速设置优先级
start_serve(self)	self 无返回值	开始服务,进行计费
stop_serve(self)	self 无返回值	停止一段服务

3.3.5 Invoice

属性	类型	说明
----	----	----

RoomId	int	房间号
Total_Fee	double	费用
date_in	int	开始时间
date_out	int	结束时间
方法名称	参数	说明
on_GetInvoice()	无返回值	获得账单

3.3.6 MonitoredObject

方法名称	参数	说明
getServiceId()	返回值:int	获得监督对象ID

3.3.7 MainAirConditioner

属性	类型	说明
mode	int	模式
speed	int	风速
Temp	int	温度
LowTemp	int	最低温度
HighTemp	int	最高温度

3.3.8 Report

属性	类型	说明
ReportID	int	报表号
RoomID	int	房间号
Times_OnOff	int	开关次数
Duration	int	时长
TotalFee	double	总费用
Number_of_DR	int	调度次数
Times_ChangeTemp	int	改变温度次数
Times_ChangeFanSpeed	int	改变风速

3.3.9 RecordDetail

属性	类型	说明
RoomId	int	房间号
RequestTime	int	请求时间

RequestDuration	int	请求时长
FanSpeed	int	风速
FeeRate	double	费率
Fee	double	费用

3.3.10 Service(用于回温的服务对象)

属性	类型	说明
CurrentTemp	double	当前温度
方法名称	参数	说明
rewarm()	CurrentTemp:double	模拟温度变化

3.4 持久化层

3.4.1 DBFacade

属性	类型	说明
RoomTable	BaseModel	房间表
BillTable	BaseModel	账单表
DetailedItemTable	BaseModel	详单表
方法名称	参数	说明
__init__	无返回值	初始化
cul_fee_rate(self, fan_speed)	self fan_speed:int 无返回值	当前费率
cul_fee(self, detailed_item_id, get_state=False)	self detailed_item_id:int get_state:bool 无返回值	当前费用
get_bill(self, room_id)	self room_id:int 无返回值	获得账单
get_detailed_list(self, room_id)	self room_id:int 无返回值	获得详单
use_air_conditioner(self, room_id, start_temp, target_temp, mode, speed, fee_rate)	self room_id:int start_temp:double target_temp:double mode:int speed:int fee_rate:double	使用空调对象

	无返回值	
stop_air_conditioner(self, detailed_item_id)	self detailed_item_id:int 无返回值	停止空调对象
get_room_state(self, room_id)	self room_id:int 无返回值	获得房间状态

3.5 数据库 SQLite

方法名称	说明
insert	增
delete	删
update	改
find	查

3.6 系统管理类（负责通信）

3.6.1 clientthread

属性	类型	说明
thread_msg	QString	消息
client	QWebSocket	socket
方法名称	参数	说明
openClient(url)	Url	打开通信
closeClient()		关闭通信
sendMessage(text)	text:string	发送消息
getAddress()		获得 IP 地址
getPort()		获得端口号

The diagram illustrates the system architecture, organized into four main layers from top to bottom:

- 用户界面 (User Interface):** Contains windows for 'consumer', 'template', 'admin', 'base', and 'system'. Each window is associated with specific UI components and data handling logic.
- 控制层 (Control Layer):** Contains controllers that manage the flow of data and logic.
 - Consumer Controller:** Manages consumer data and interacts with the 'Consumer Controller' service.
 - Template Controller:** Manages template data and interacts with the 'Template Controller' service.
 - Admin Controller:** Manages administrative data and interacts with the 'Admin Controller' service.
 - Base Controller:** Manages base data and interacts with the 'Base Controller' service.
- 业务逻辑层 (Business Logic Layer):** Contains the core business logic and services.
 - Consumer Controller:** A service that handles consumer-related logic, including scheduling, waiting queues, and timers.
 - Template Controller:** A service that handles template-related logic, including monitoring and reporting.
 - Admin Controller:** A service that handles administrative logic, including monitoring and reporting.
 - Base Controller:** A service that handles base-level logic, including monitoring and reporting.
 - Services:** Various utility services like 'Scheduler', 'Waiting Queue', 'Timer', 'Monitor', 'Report', and 'Diffable' that support the core logic.
- 持久化层 (Persistence Layer):** The bottom layer, which handles data storage and retrieval, including 'Database' and 'Cache' components.

The diagram uses a color-coded system: blue for the User Interface, green for the Control Layer, yellow for the Business Logic Layer, and red for the Persistence Layer. Arrows indicate the flow of data and control between these components.