Hospital

Hospital.Warehouse

Campus
Warehouse warehouse

holds

Warehouse
List<Stock> stocks

get Warehouse from

«interface»
TimeObserver

1

holds

processes

get Stock from

...

Item

prototype

1

uses

Queue<I>

holds items

Stock<I extends Item>
I prototype
OrderPlacer placer
Queue<I> items
OrderList orders
List<ItemReservation> res
reserve(int count, Time at) : ItemReservation

n

holds orders

OrderList

manages

Order

ItemReservationCommand
execute()
use()

Command

makeReservation : ItemReservation

holds reservations

0..n

ItemReservation
use() : List<Item>

created reservation

1

StockConsistency
evaluateEvents(...) : boolean

check attempted reservation

simulates

StockChangeEvent
stockChange(int) : int
orderChange(int) : int
getTime() : Time
compare(HasTime time) : int

1

OrderPlacer
checkStock(int currentStock, int ordered, int maxStock) : int

holds

World
worldTime : WorldTime
...
getResourceByType(Class<S> clazz) : List<S>

holds

WorldTime
currentTime : Time
...

gets time from

current Time

uses

Time

Hospital.Schedule.ScheduleGroups

Schedulable

uses

«Interface»
ScheduleGroup
getSchedulables() : List<Schedulable>

0..n

0..n

selects attendees from

MultiScheduleGroup
Class<S extends Schedulable> type
World world
getSchedulables() : List<Schedulables>

SingleScheduleGroup
schedulable : Schedulable
getSchedulables() : List<Schedulable>

get schedulables from ▸

aggregates all groups

Hospital.Schedule.ConstraintSolver

JumpSolver

BruteForceSolver

AppointmentResult
getCampus() : Campus
getAttendees() : List<Schedulable>
getTime() : Time
getLength() : int

«interface»
AppointmentConstraintSolver
setTimeFrameConstraints(List<TimeFrameConstraint> tfC)
setCampusDecider(CampusDecider cd)
setDelayedTimeLength(DelayedTimeLength td)
setScheduleGroups(List<ScheduleGroup> groups)
getCampus() : Campus
getAttendees() : List<Schedulable>
getTime() : Time
solve()

1

creates

decide Campus

SolverAdapter
SINGLETON : SolverAdapter
solve(...) : AppointmentResult

get first Time from

Hospital.Schedule.GetCampus

CampusDecider
+getCampus() : Campus
setSchedulable(Schedulable s)
setStaff(Staff s)
setDoctor(Doctor d)
setNurse(Nurse n)
setPatient(Patient p)

SchedulableVisitor

NurseDecides
getCampus() : Campus

PatientDecides
getCampus() : Campus

Hospital.Schedules

must satisfy constraints

make fitting appointment

get result from

DelayedTimeLength
delay : int
length : int
worldTime : WorldTime
getLength() : int
getDelayedTime() : Time
setWorldTime(WorldTime wt)
satisfy constraints at

0..n

TimeFrameConstraint
+reset()
+isAccepted() : Time
(exception ScheduleConstraintException)
+setTime(Time time, int length)
setSchedulable(Schedulable s)
setStaff(Staff s)
setDoctor(Doctor d)
setNurse(Nurse n)
setPatient(Patient p)

AppointmentCommand
next : Command
execute() throws CannotDoException
undo()
isDone() : boolean

creates

holds creator

Appointment
attendees : List<Schedulable>
time : Time
length : int
campus : Campus
appCommand: AppointmentCommand
Appointment(AppointmentResult appResult, ...)

aggregates

makes Appointment for

Command

uses ▸

SchedulableVisitor
setSchedulable(Schedulable s)
setStaff(Staff s)
setDoctor(Doctor d)
setNurse(Nurse n)
setPatient(Patient p)

SchedulableVisitor

«Interface»
Appointable
setAppointment(Appointment app)
getAppointment() : Appointment
getConstraints() : List<TimeFrameConstraint>
getCampusDecider() : CampusDecider
getScheduleGroups() : List<ScheduleGroup>
getTimeFrameDelay() : TimeFrameDelay

Schedule
...
...

has one

1

«Interface»
Schedulable
getSchedule() : Schedule
visitConstraints(TimeFrameConstraint tfC)
getConstraints() : List<TimeFrameConstraint>

controls

controls

Machine
schedule
visitConstraints(TimeFrameConstraint tfC)
getConstraints() : List<TimeFrameConstraint>
...

XRayMachine

...

Person
schedule : Schedule
...
visitConstraints(TimeFrameConstraint tfC)
getConstraints() : List<TimeFrameConstraint>
...

add diagnosis

Patient

Staff

...

Doctor

has

creator

1

0..1

Treatment
...

...

«interface»
Result
enterResult(Argument[] args)
getEmptyArguments() : Argument[]
validate(Argument[] args) : boolean
isResultEntered() : boolean
getResultString() : String

MedicalTest
...

...

creates

n

DoctorPatientappointment

1

Diagnosis
...

aggregates
introduce to patient

n

creates

triggers execution

aggregates

creates

DiagnosisSecondOpinion

See scheduling...

creates

AppointmentCommand
...

triggers execution

approves

DiagnosisCommand
execute()
undo()
...

DiagnosisApproveCommand
execute()
undo()
...

TreatmentCommand
execute()
undo()
...

creates

MedicalTestCommand
execute()
undo()
...

create
execute

create
execute

create
execute

DiagnosisCreator
makeDiagnosis(...)

controls

controls

add treatmentCommand

TreatmentMaker
makeCast(...)
makeMedication(...)
makeSurgery(...)

MedicalTestCreator
makeXRay(...)
makeBloodAnalysis(...)
makeUltraSoundScan(...)

controls

controls

controller for

? extends Staff

creates

StaffICreator
create(...) : String

? extends Machine

creates

MachineAbstractFactory
make(...) : String

controls

controls

Hospital.Controllers

WarehouseController
listOrders(...)
processOrder(...)

StaffController
create(...) : String

MachineController
getMachineTypes() : String[]
makeMachine(...) : String

WorldController
login(...) : LoginController

LoginController

PatientController
registerPatient(String name) : String

DiagnosisController
enterDiagnosis(...) : String
approveDiagnosis(...) : String
disapproveDiagnosis(...) : String

TreatmentController
make...(...) : String

MedicalTestController
make...(...) : String

AdministratorController
advanceTime(Time t)

NurseControllerController
checkIn(String name) : String

DoctorController
openPatientFile(...)
dischargePatient(...)
undo(...)
redo(...)

TreatmentResultController
getResultArguments() : ...
enterResults(...) : String

MedicalTestResultController
getResultArguments() : ...
enterResults(...) : String