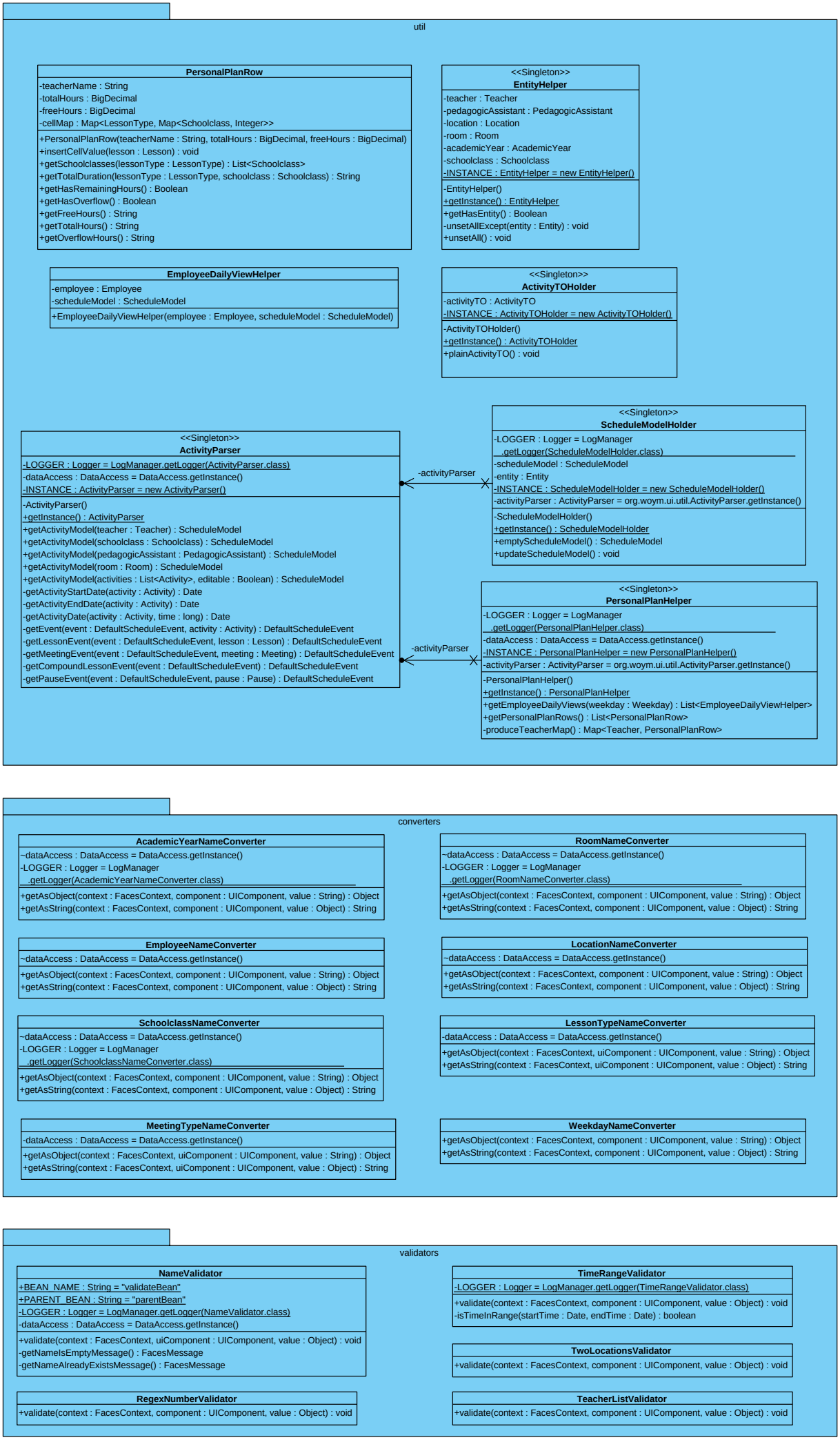
util
PersonalPlanRow
-teacherName : String
-totalHours : BigDecimal
-freeHours : BigDecimal
-cellMap : Map<LessonType, Map<Schoolclass, Integer>>
+PersonalPlanRow(teacherName : String, totalHours : BigDecimal, freeHours : BigDecimal)
+insertCellValue(lesson : Lesson) : void
+getSchoolclasses(lessonType : LessonType) : List<Schoolclass>
+getTotalDuration(lessonType : LessonType, schoolclass : Schoolclass) : String
+getHasRemainingHours() : Boolean
+getHasOverflow() : Boolean
+getFreeHours() : String
+getTotalHours() : String
+getOverflowHours() : String
<<Singleton>>
EntityHelper
-teacher : Teacher
-pedagogicAssistant : PedagogicAssistant
-location : Location
-room : Room
-academicYear : AcademicYear
-schoolclass : Schoolclass
-INSTANCE : EntityHelper = new EntityHelper()
-EntityHelper()
+getInstance() : EntityHelper
+getHasEntity() : Boolean
-unsetAllExcept(entity : Entity) : void
+unsetAll() : void
EmployeeDailyViewHelper
-employee : Employee
-scheduleModel : ScheduleModel
+EmployeeDailyViewHelper(employee : Employee, scheduleModel : ScheduleModel)
<<Singleton>>
ActivityTOHolder
-activityTO : ActivityTO
-INSTANCE : ActivityTOHolder = new ActivityTOHolder()
-ActivityTOHolder()
+getInstance() : ActivityTOHolder
+plainActivityTO() : void
<<Singleton>>
ActivityParser
-LOGGER : Logger = LogManager.getLogger(ActivityParser.class)
-dataAccess : DataAccess = DataAccess.getInstance()
-INSTANCE : ActivityParser = new ActivityParser()
-ActivityParser()
+getInstance() : ActivityParser
+getActivityModel(teacher : Teacher) : ScheduleModel
+getActivityModel(schoolclass : Schoolclass) : ScheduleModel
+getActivityModel(pedagogicAssistant : PedagogicAssistant) : ScheduleModel
+getActivityModel(room : Room) : ScheduleModel
+getActivityModel(activities : List<Activity>, editable : Boolean) : ScheduleModel
-getActivityStartDate(activity : Activity) : Date
-getActivityEndDate(activity : Activity) : Date
-getActivityDate(activity : Activity, time : long) : Date
-getEvent(event : DefaultScheduleEvent, activity : Activity) : DefaultScheduleEvent
-getLessonEvent(event : DefaultScheduleEvent, lesson : Lesson) : DefaultScheduleEvent
-getMeetingEvent(event : DefaultScheduleEvent, meeting : Meeting) : DefaultScheduleEvent
-getCompoundLessonEvent(event : DefaultScheduleEvent) : DefaultScheduleEvent
-getPauseEvent(event : DefaultScheduleEvent, pause : Pause) : DefaultScheduleEvent
-activityParser
<<Singleton>>
ScheduleModelHolder
-LOGGER : Logger = LogManager.getLogger(ScheduleModelHolder.class)
-scheduleModel : ScheduleModel
-entity : Entity
-INSTANCE : ScheduleModelHolder = new ScheduleModelHolder()
-activityParser : ActivityParser = org.woym.ui.util.ActivityParser.getInstance()
-ScheduleModelHolder()
+getInstance() : ScheduleModelHolder
+emptyScheduleModel() : ScheduleModel
+updateScheduleModel() : void
-activityParser
<<Singleton>>
PersonalPlanHelper
-LOGGER : Logger = LogManager.getLogger(PersonalPlanHelper.class)
-dataAccess : DataAccess = DataAccess.getInstance()
-INSTANCE : PersonalPlanHelper = new PersonalPlanHelper()
-activityParser : ActivityParser = org.woym.ui.util.ActivityParser.getInstance()
-PersonalPlanHelper()
+getInstance() : PersonalPlanHelper
+getEmployeeDailyViews(weekday : Weekday) : List<EmployeeDailyViewHelper>
+getPersonalPlanRows() : List<PersonalPlanRow>
-produceTeacherMap() : Map<Teacher, PersonalPlanRow>
converters
AcademicYearNameConverter
~dataAccess : DataAccess = DataAccess.getInstance()
-LOGGER : Logger = LogManager.getLogger(AcademicYearNameConverter.class)
+getAsObject(context : FacesContext, component : UIComponent, value : String) : Object
+getAsString(context : FacesContext, component : UIComponent, value : Object) : String
RoomNameConverter
~dataAccess : DataAccess = DataAccess.getInstance()
-LOGGER : Logger = LogManager.getLogger(RoomNameConverter.class)
+getAsObject(context : FacesContext, component : UIComponent, value : String) : Object
+getAsString(context : FacesContext, component : UIComponent, value : Object) : String
EmployeeNameConverter
~dataAccess : DataAccess = DataAccess.getInstance()
+getAsObject(context : FacesContext, component : UIComponent, value : String) : Object
+getAsString(context : FacesContext, component : UIComponent, value : Object) : String
LocationNameConverter
~dataAccess : DataAccess = DataAccess.getInstance()
+getAsObject(context : FacesContext, component : UIComponent, value : String) : Object
+getAsString(context : FacesContext, component : UIComponent, value : Object) : String
SchoolclassNameConverter
~dataAccess : DataAccess = DataAccess.getInstance()
-LOGGER : Logger = LogManager.getLogger(SchoolclassNameConverter.class)
+getAsObject(context : FacesContext, component : UIComponent, value : String) : Object
+getAsString(context : FacesContext, component : UIComponent, value : Object) : String
LessonTypeNameConverter
-dataAccess : DataAccess = DataAccess.getInstance()
+getAsObject(context : FacesContext, uiComponent : UIComponent, value : String) : Object
+getAsString(context : FacesContext, uiComponent : UIComponent, value : Object) : String
MeetingTypeNameConverter
-dataAccess : DataAccess = DataAccess.getInstance()
+getAsObject(context : FacesContext, uiComponent : UIComponent, value : String) : Object
+getAsString(context : FacesContext, uiComponent : UIComponent, value : Object) : String
WeekdayNameConverter
+getAsObject(context : FacesContext, component : UIComponent, value : String) : Object
+getAsString(context : FacesContext, component : UIComponent, value : Object) : String
validators
NameValidator
+BEAN_NAME : String = "validateBean"
+PARENT_BEAN : String = "parentBean"
-LOGGER : Logger = LogManager.getLogger(NameValidator.class)
-dataAccess : DataAccess = DataAccess.getInstance()
+validate(context : FacesContext, uiComponent : UIComponent, value : Object) : void
-getNameIsEmptyMessage() : FacesMessage
-getNameAlreadyExistsMessage() : FacesMessage
TimeRangeValidator
-LOGGER : Logger = LogManager.getLogger(TimeRangeValidator.class)
+validate(context : FacesContext, component : UIComponent, value : Object) : void
-isTimeInRange(startTime : Date, endTime : Date) : boolean
TwoLocationsValidator
+validate(context : FacesContext, component : UIComponent, value : Object) : void
RegexNumberValidator
+validate(context : FacesContext, component : UIComponent, value : Object) : void
TeacherListValidator
+validate(context : FacesContext, component : UIComponent, value : Object) : void