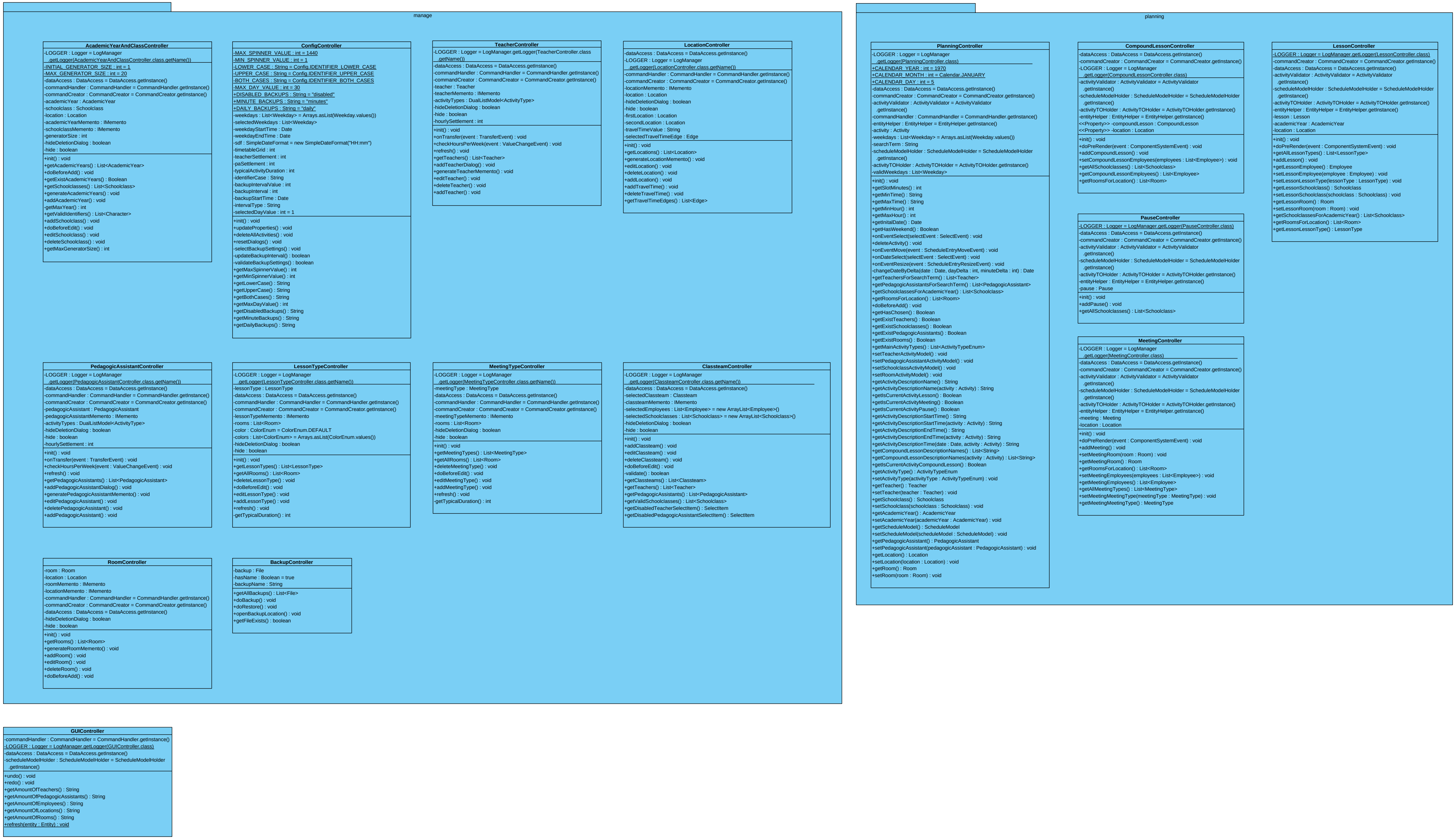

manage

AcademicYearAndClassController

```
-LOGGER : Logger = LogManager
   .getLogger(AcademicYearAndClassController.class.getName())
-INITIAL_GENERATOR_SIZE : int = 1
-MAX_GENERATOR_SIZE : int = 20
-dataAccess : DataAccess = DataAccess.getInstance()
-commandHandler : CommandHandler = CommandHandler.getInstance()
-commandCreator : CommandCreator = CommandCreator.getInstance()
-academicYear : AcademicYear
-schoolclass : Schoolclass
-location : Location
-academicYearMemento : IMemento
-schoolclassMemento : IMemento
-generatorSize : int
-hideDeletionDialog : boolean
-hide : boolean
+init() : void
+getAcademicYears() : List<AcademicYear>
+doBeforeAdd() : void
+getExistAcademicYears() : Boolean
+getSchoolclasses() : List<Schoolclass>
+generateAcademicYears() : void
+addAcademicYear() : void
-getMaxYear() : int
+getValidIdentifiers() : List<Character>
+addSchoolclass() : void
+doBeforeEdit() : void
+editSchoolclass() : void
+deleteSchoolclass() : void
+getMaxGeneratorSize() : int
```

ConfigController

```
-MAX_SPINNER_VALUE : int = 1440
-MIN_SPINNER_VALUE : int = 1
-LOWER_CASE : String = Config.IDENTIFIER_LOWER_CASE
-UPPER_CASE : String = Config.IDENTIFIER_UPPER_CASE
-BOTH_CASES : String = Config.IDENTIFIER_BOTH_CASES
-MAX_DAY_VALUE : int = 30
+DISABLED_BACKUPS : String = "disabled"
+MINUTE_BACKUPS : String = "minutes"
+DAILY_BACKUPS : String = "daily"
-weekdays : List<Weekday> = Arrays.asList(Weekday.values())
-selectedWeekdays : List<Weekday>
-weekdayStartTime : Date
-weekdayEndTime : Date
-sdf : SimpleDateFormat = new SimpleDateFormat("HH:mm")
-timetableGrid : int
-teacherSettlement : int
-paSettlement : int
-typicalActivityDuration : int
-identifierCase : String
-backupIntervalValue : int
-backupInterval : int
-backupStartTime : Date
-intervalType : String
-selectedDayValue : int = 1
+init() : void
+updateProperties() : void
+deleteAllActivities() : void
+resetDialogs() : void
-selectBackupSettings() : void
-updateBackupInterval() : boolean
-validateBackupSettings() : boolean
+getMaxSpinnerValue() : int
+getMinSpinnerValue() : int
+getLowerCase() : String
+getUpperCase() : String
+getBothCases() : String
+getMaxDayValue() : int
+getDisabledBackups() : String
+getMinuteBackups() : String
+getDailyBackups() : String
```

TeacherController

```
-LOGGER : Logger = LogManager.getLogger(TeacherController.class
   .getName())
-dataAccess : DataAccess = DataAccess.getInstance()
-commandHandler : CommandHandler = CommandHandler.getInstance()
-commandCreator : CommandCreator = CommandCreator.getInstance()
-teacher : Teacher
-teacherMemento : IMemento
-activityTypes : DualListModel<ActivityType>
-hideDeletionDialog : boolean
-hide : boolean
-hourlySettlement : int
+init() : void
+onTransfer(event : TransferEvent) : void
+checkHoursPerWeek(event : ValueChangeEvent) : void
+refresh() : void
+getTeachers() : List<Teacher>
+addTeacherDialog() : void
+generateTeacherMemento() : void
+editTeacher() : void
+deleteTeacher() : void
+addTeacher() : void
```

LocationController

```
-dataAccess : DataAccess = DataAccess.getInstance()
-LOGGER : Logger = LogManager
   .getLogger(LocationController.class.getName())
-commandHandler : CommandHandler = CommandHandler.getInstance()
-commandCreator : CommandCreator = CommandCreator.getInstance()
-locationMemento : IMemento
-location : Location
-hideDeletionDialog : boolean
-hide : boolean
-firstLocation : Location
-secondLocation : Location
-travelTimeValue : String
-selectedTravelTimeEdge : Edge
+init() : void
+getLocations() : List<Location>
+generateLocationMemento() : void
+editLocation() : void
+deleteLocation() : void
+addLocation() : void
+addTravelTime() : void
+deleteTravelTime() : void
+getTravelTimeEdges() : List<Edge>
```

PedagogicAssistantController

```
-LOGGER : Logger = LogManager
   .getLogger(PedagogicAssistantController.class.getName())
-dataAccess : DataAccess = DataAccess.getInstance()
-commandHandler : CommandHandler = CommandHandler.getInstance()
-commandCreator : CommandCreator = CommandCreator.getInstance()
-pedagogicAssistant : PedagogicAssistant
-pedagogicAssistantMemento : IMemento
-activityTypes : DualListModel<ActivityType>
-hideDeletionDialog : boolean
-hide : boolean
-hourlySettlement : int
+init() : void
+onTransfer(event : TransferEvent) : void
+checkHoursPerWeek(event : ValueChangeEvent) : void
+refresh() : void
+getPedagogicAssistants() : List<PedagogicAssistant>
+addPedagogicAssistantDialog() : void
+generatePedagogicAssistantMemento() : void
+editPedagogicAssistant() : void
+deletePedagogicAssistant() : void
+addPedagogicAssistant() : void
```

LessonTypeController

```
-LOGGER : Logger = LogManager
   .getLogger(LessonTypeController.class.getName())
-lessonType : LessonType
-dataAccess : DataAccess = DataAccess.getInstance()
-commandHandler : CommandHandler = CommandHandler.getInstance()
-commandCreator : CommandCreator = CommandCreator.getInstance()
-lessonTypeMemento : IMemento
-rooms : List<Room>
-color : ColorEnum = ColorEnum.DEFAULT
-colors : List<ColorEnum> = Arrays.asList(ColorEnum.values())
-hideDeletionDialog : boolean
-hide : boolean
+init() : void
+getLessonTypes() : List<LessonType>
+getAllRooms() : List<Room>
+deleteLessonType() : void
+doBeforeEdit() : void
+editLessonType() : void
+addLessonType() : void
+refresh() : void
-getTypicalDuration() : int
```

MeetingTypeController

```
-LOGGER : Logger = LogManager
   .getLogger(MeetingTypeController.class.getName())
-meetingType : MeetingType
-dataAccess : DataAccess = DataAccess.getInstance()
-commandHandler : CommandHandler = CommandHandler.getInstance()
-commandCreator : CommandCreator = CommandCreator.getInstance()
-meetingTypeMemento : IMemento
-rooms : List<Room>
-hideDeletionDialog : boolean
-hide : boolean
+init() : void
+getMeetingTypes() : List<MeetingType>
+getAllRooms() : List<Room>
+deleteMeetingType() : void
+doBeforeEdit() : void
+editMeetingType() : void
+addMeetingType() : void
+refresh() : void
-getTypicalDuration() : int
```

ClassteamController

```
-LOGGER : Logger = LogManager
   .getLogger(ClassteamController.class.getName())
-dataAccess : DataAccess = DataAccess.getInstance()
-selectedClassteam : Classteam
-classteamMemento : IMemento
-selectedEmployees : List<Employee> = new ArrayList<Employee>()
-selectedSchoolclasses : List<Schoolclass> = new ArrayList<Schoolclass>()
-hideDeletionDialog : boolean
-hide : boolean
+init() : void
+addClassteam() : void
+editClassteam() : void
+deleteClassteam() : void
+doBeforeEdit() : void
-validate() : boolean
+getClassteams() : List<Classteam>
+getTeachers() : List<Teacher>
+getPedagogicAssistants() : List<PedagogicAssistant>
+getValidSchoolclasses() : List<Schoolclass>
+getDisabledTeacherSelectItem() : SelectItem
+getDisabledPedagogicAssistantSelectItem() : SelectItem
```

RoomController

```
-room : Room
-location : Location
-roomMemento : IMemento
-locationMemento : IMemento
-commandHandler : CommandHandler = CommandHandler.getInstance()
-commandCreator : CommandCreator = CommandCreator.getInstance()
-dataAccess : DataAccess = DataAccess.getInstance()
-hideDeletionDialog : boolean
-hide : boolean
+init() : void
+getRooms() : List<Room>
+generateRoomMemento() : void
+addRoom() : void
+editRoom() : void
+deleteRoom() : void
+doBeforeAdd() : void
```

BackupController

```
-backup : File
-hasName : Boolean = true
-backupName : String
+getAllBackups() : List<File>
+doBackup() : void
+doRestore() : void
+openBackupLocation() : void
+getFileExists() : boolean
```

planning

PlanningController

```
-LOGGER : Logger = LogManager
   .getLogger(PlanningController.class)
+CALENDAR_YEAR : int = 1970
+CALENDAR_MONTH : int = Calendar.JANUARY
+CALENDAR_DAY : int = 5
-dataAccess : DataAccess = DataAccess.getInstance()
-commandCreator : CommandCreator = CommandCreator.getInstance()
-activityValidator : ActivityValidator = ActivityValidator
   .getInstance()
-commandHandler : CommandHandler = CommandHandler.getInstance()
-entityHelper : EntityHelper = EntityHelper.getInstance()
-activity : Activity
-weekdays : List<Weekday> = Arrays.asList(Weekday.values())
-searchTerm : String
-scheduleModelHolder : ScheduleModelHolder = ScheduleModelHolder
   .getInstance()
-activityTOHolder : ActivityTOHolder = ActivityTOHolder.getInstance()
-validWeekdays : List<Weekday>
+init() : void
+getSlotMinutes() : int
+getMinTime() : String
+getMaxTime() : String
+getMinHour() : int
+getMaxHour() : int
+getInitalDate() : Date
+getHasWeekend() : Boolean
+onEventSelect(selectEvent : SelectEvent) : void
+deleteActivity() : void
+onEventMove(event : ScheduleEntryMoveEvent) : void
+onDateSelect(selectEvent : SelectEvent) : void
+onEventResize(event : ScheduleEntryResizeEvent) : void
-changeDateByDelta(date : Date, dayDelta : int, minuteDelta : int) : Date
+getTeachersForSearchTerm() : List<Teacher>
+getPedagogicAssistantsForSearchTerm() : List<PedagogicAssistant>
+getSchoolclassesForAcademicYear() : List<Schoolclass>
+getRoomsForLocation() : List<Room>
+doBeforeAdd() : void
+getHasChosen() : Boolean
+getExistTeachers() : Boolean
+getExistSchoolclasses() : Boolean
+getExistPedagogicAssistants() : Boolean
+getExistRooms() : Boolean
+getMainActivityTypes() : List<ActivityTypeEnum>
+setTeacherActivityModel() : void
+setPedagogicAssistantActivityModel() : void
+setSchoolclassActivityModel() : void
+setRoomActivityModel() : void
+getActivityDescriptionName() : String
+getActivityDescriptionName(activity : Activity) : String
+getIsCurrentActivityLesson() : Boolean
+getIsCurrentActivityMeeting() : Boolean
+getIsCurrentActivityPause() : Boolean
+getActivityDescriptionStartTime() : String
+getActivityDescriptionStartTime(activity : Activity) : String
+getActivityDescriptionEndTime() : String
+getActivityDescriptionEndTime(activity : Activity) : String
+getActivityDescriptionTime(date : Date, activity : Activity) : String
+getCompoundLessonDescriptionNames() : List<String>
+getCompoundLessonDescriptionNames(activity : Activity) : List<String>
+getIsCurrentActivityCompoundLesson() : Boolean
+getActivityType() : ActivityTypeEnum
+setActivityType(activityType : ActivityTypeEnum) : void
+getTeacher() : Teacher
+setTeacher(teacher : Teacher) : void
+getSchoolclass() : Schoolclass
+setSchoolclass(schoolclass : Schoolclass) : void
+getAcademicYear() : AcademicYear
+setAcademicYear(academicYear : AcademicYear) : void
+getScheduleModel() : ScheduleModel
+setScheduleModel(scheduleModel : ScheduleModel) : void
+getPedagogicAssistant() : PedagogicAssistant
+setPedagogicAssistant(pedagogicAssistant : PedagogicAssistant) : void
+getLocation() : Location
+setLocation(location : Location) : void
+getRoom() : Room
+setRoom(room : Room) : void
```

CompoundLessonController

```
-dataAccess : DataAccess = DataAccess.getInstance()
-commandCreator : CommandCreator = CommandCreator.getInstance()
-LOGGER : Logger = LogManager
   .getLogger(CompoundLessonController.class)
-activityValidator : ActivityValidator = ActivityValidator
   .getInstance()
-scheduleModelHolder : ScheduleModelHolder = ScheduleModelHolder
   .getInstance()
-activityTOHolder : ActivityTOHolder = ActivityTOHolder.getInstance()
-entityHelper : EntityHelper = EntityHelper.getInstance()
<<Property>> -compoundLesson : CompoundLesson
<<Property>> -location : Location
+init() : void
+doPreRender(event : ComponentSystemEvent) : void
+addCompoundLesson() : void
+setCompoundLessonEmployees(employees : List<Employee>) : void
+getAllSchoolclasses() : List<Schoolclass>
+getCompoundLessonEmployees() : List<Employee>
+getRoomsForLocation() : List<Room>
```

LessonController

```
-LOGGER : Logger = LogManager.getLogger(LessonController.class)
-commandCreator : CommandCreator = CommandCreator.getInstance()
-dataAccess : DataAccess = DataAccess.getInstance()
-activityValidator : ActivityValidator = ActivityValidator
   .getInstance()
-scheduleModelHolder : ScheduleModelHolder = ScheduleModelHolder
   .getInstance()
-activityTOHolder : ActivityTOHolder = ActivityTOHolder.getInstance()
-entityHelper : EntityHelper = EntityHelper.getInstance()
-lesson : Lesson
-academicYear : AcademicYear
-location : Location
+init() : void
+doPreRender(event : ComponentSystemEvent) : void
+getAllLessonTypes() : List<LessonType>
+addLesson() : void
+getLessonEmployee() : Employee
+setLessonEmployee(employee : Employee) : void
+setLessonLessonType(lessonType : LessonType) : void
+getLessonSchoolclass() : Schoolclass
+setLessonSchoolclass(schoolclass : Schoolclass) : void
+getLessonRoom() : Room
+setLessonRoom(room : Room) : void
+getSchoolclassesForAcademicYear() : List<Schoolclass>
+getRoomsForLocation() : List<Room>
+getLessonLessonType() : LessonType
```

PauseController

```
-LOGGER : Logger = LogManager.getLogger(PauseController.class)
-dataAccess : DataAccess = DataAccess.getInstance()
-commandCreator : CommandCreator = CommandCreator.getInstance()
-activityValidator : ActivityValidator = ActivityValidator
   .getInstance()
-scheduleModelHolder : ScheduleModelHolder = ScheduleModelHolder
   .getInstance()
-activityTOHolder : ActivityTOHolder = ActivityTOHolder.getInstance()
-entityHelper : EntityHelper = EntityHelper.getInstance()
-pause : Pause
+init() : void
+addPause() : void
+getAllSchoolclasses() : List<Schoolclass>
```

MeetingController

```
-LOGGER : Logger = LogManager
   .getLogger(MeetingController.class)
-dataAccess : DataAccess = DataAccess.getInstance()
-commandCreator : CommandCreator = CommandCreator.getInstance()
-activityValidator : ActivityValidator = ActivityValidator
   .getInstance()
-scheduleModelHolder : ScheduleModelHolder = ScheduleModelHolder
   .getInstance()
-activityTOHolder : ActivityTOHolder = ActivityTOHolder.getInstance()
-entityHelper : EntityHelper = EntityHelper.getInstance()
-meeting : Meeting
-location : Location
+init() : void
+doPreRender(event : ComponentSystemEvent) : void
+addMeeting() : void
+setMeetingRoom(room : Room) : void
+getMeetingRoom() : Room
+getRoomsForLocation() : List<Room>
+setMeetingEmployees(employees : List<Employee>) : void
+getMeetingEmployees() : List<Employee>
+getAllMeetingTypes() : List<MeetingType>
+setMeetingMeetingType(meetingType : MeetingType) : void
+getMeetingMeetingType() : MeetingType
```

GUIController

```
-commandHandler : CommandHandler = CommandHandler.getInstance()
-LOGGER : Logger = LogManager.getLogger(GUIController.class)
-dataAccess : DataAccess = DataAccess.getInstance()
-scheduleModelHolder : ScheduleModelHolder = ScheduleModelHolder
   .getInstance()
+undo() : void
+redo() : void
+getAmountOfTeachers() : String
+getAmountOfPedagogicAssistants() : String
+getAmountOfEmployees() : String
+getAmountOfLocations() : String
+getAmountOfRooms() : String
+refresh(entity : Entity) : void
```