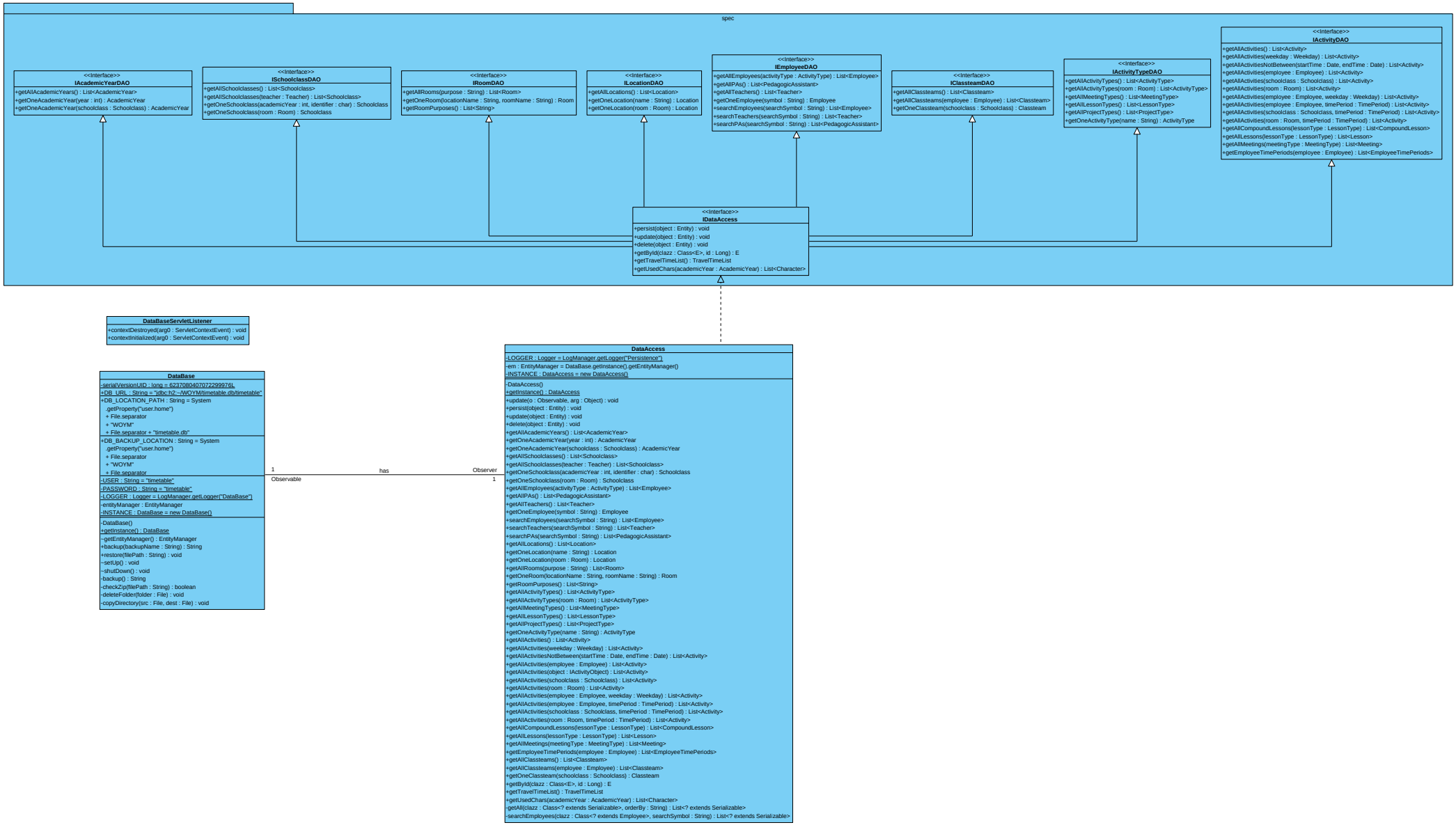
spec
<<Interface>>
IAcademicYearDAO
+getAllAcademicYears() : List<AcademicYear>
+getOneAcademicYear(year : int) : AcademicYear
+getOneAcademicYear(schoolclass : Schoolclass) : AcademicYear
<<Interface>>
ISchoolclassDAO
+getAllSchoolclasses() : List<Schoolclass>
+getAllSchoolclasses(teacher : Teacher) : List<Schoolclass>
+getOneSchoolclass(academicYear : int, identifier : char) : Schoolclass
+getOneSchoolclass(room : Room) : Schoolclass
<<Interface>>
IRoomDAO
+getAllRooms(purpose : String) : List<Room>
+getOneRoom(locationName : String, roomName : String) : Room
+getRoomPurposes() : List<String>
<<Interface>>
ILocationDAO
+getAllLocations() : List<Location>
+getOneLocation(name : String) : Location
+getOneLocation(room : Room) : Location
<<Interface>>
IEmployeeDAO
+getAllEmployees(activityType : ActivityType) : List<Employee>
+getAllPAs() : List<PedagogicAssistant>
+getAllTeachers() : List<Teacher>
+getOneEmployee(symbol : String) : Employee
+searchEmployees(searchSymbol : String) : List<Employee>
+searchTeachers(searchSymbol : String) : List<Teacher>
+searchPAs(searchSymbol : String) : List<PedagogicAssistant>
<<Interface>>
IClassteamDAO
+getAllClassteams() : List<Classteam>
+getAllClassteams(employee : Employee) : List<Classteam>
+getOneClassteam(schoolclass : Schoolclass) : Classteam
<<Interface>>
IActivityTypeDAO
+getAllActivityTypes() : List<ActivityType>
+getAllActivityTypes(room : Room) : List<ActivityType>
+getAllMeetingTypes() : List<MeetingType>
+getAllLessonTypes() : List<LessonType>
+getAllProjectTypes() : List<ProjectType>
+getOneActivityType(name : String) : ActivityType
<<Interface>>
IActivityDAO
+getAllActivities() : List<Activity>
+getAllActivities(weekday : Weekday) : List<Activity>
+getAllActivitiesNotBetween(startTime : Date, endTime : Date) : List<Activity>
+getAllActivities(employee : Employee) : List<Activity>
+getAllActivities(schoolclass : Schoolclass) : List<Activity>
+getAllActivities(room : Room) : List<Activity>
+getAllActivities(employee : Employee, weekday : Weekday) : List<Activity>
+getAllActivities(employee : Employee, timePeriod : TimePeriod) : List<Activity>
+getAllActivities(schoolclass : Schoolclass, timePeriod : TimePeriod) : List<Activity>
+getAllActivities(room : Room, timePeriod : TimePeriod) : List<Activity>
+getAllCompoundLessons(lessonType : LessonType) : List<CompoundLesson>
+getAllLessons(lessonType : LessonType) : List<Lesson>
+getAllMeetings(meetingType : MeetingType) : List<Meeting>
+getEmployeeTimePeriods(employee : Employee) : List<EmployeeTimePeriods>
<<Interface>>
IDataAccess
+persist(object : Entity) : void
+update(object : Entity) : void
+delete(object : Entity) : void
+getById(clazz : Class<E>, id : Long) : E
+getTravelTimeList() : TravelTimeList
+getUsedChars(academicYear : AcademicYear) : List<Character>
DataBaseServletListener
+contextDestroyed(arg0 : ServletContextEvent) : void
+contextInitialized(arg0 : ServletContextEvent) : void
DataBase
-serialVersionUID : long = 6237080407072299976L
+DB_URL : String = "jdbc:h2:~/WOYM/timetable.db/timetable"
+DB_LOCATION_PATH : String = System
.getProperty("user.home")
+ File.separator
+ "WOYM"
+ File.separator + "timetable.db"
+DB_BACKUP_LOCATION : String = System
.getProperty("user.home")
+ File.separator
+ "WOYM"
+ File.separator
-USER : String = "timetable"
-PASSWORD : String = "timetable"
-LOGGER : Logger = LogManager.getLogger("DataBase")
-entityManager : EntityManager
-INSTANCE : DataBase = new DataBase()
-DataBase()
+getInstance() : DataBase
-getEntityManager() : EntityManager
+backup(backupName : String) : String
+restore(filePath : String) : void
~setUp() : void
~shutDown() : void
-backup() : String
-checkZip(filePath : String) : boolean
-deleteFolder(folder : File) : void
-copyDirectory(src : File, dest : File) : void
1
has
Observer
Observable
1
DataAccess
-LOGGER : Logger = LogManager.getLogger("Persistence")
-em : EntityManager = DataBase.getInstance().getEntityManager()
-INSTANCE : DataAccess = new DataAccess()
-DataAccess()
+getInstance() : DataAccess
+update(o : Observable, arg : Object) : void
+persist(object : Entity) : void
+update(object : Entity) : void
+delete(object : Entity) : void
+getAllAcademicYears() : List<AcademicYear>
+getOneAcademicYear(year : int) : AcademicYear
+getOneAcademicYear(schoolclass : Schoolclass) : AcademicYear
+getAllSchoolclasses() : List<Schoolclass>
+getAllSchoolclasses(teacher : Teacher) : List<Schoolclass>
+getOneSchoolclass(academicYear : int, identifier : char) : Schoolclass
+getOneSchoolclass(room : Room) : Schoolclass
+getAllEmployees(activityType : ActivityType) : List<Employee>
+getAllPAs() : List<PedagogicAssistant>
+getAllTeachers() : List<Teacher>
+getOneEmployee(symbol : String) : Employee
+searchEmployees(searchSymbol : String) : List<Employee>
+searchTeachers(searchSymbol : String) : List<Teacher>
+searchPAs(searchSymbol : String) : List<PedagogicAssistant>
+getAllLocations() : List<Location>
+getOneLocation(name : String) : Location
+getOneLocation(room : Room) : Location
+getAllRooms(purpose : String) : List<Room>
+getOneRoom(locationName : String, roomName : String) : Room
+getRoomPurposes() : List<String>
+getAllActivityTypes() : List<ActivityType>
+getAllActivityTypes(room : Room) : List<ActivityType>
+getAllMeetingTypes() : List<MeetingType>
+getAllLessonTypes() : List<LessonType>
+getAllProjectTypes() : List<ProjectType>
+getOneActivityType(name : String) : ActivityType
+getAllActivities() : List<Activity>
+getAllActivities(weekday : Weekday) : List<Activity>
+getAllActivitiesNotBetween(startTime : Date, endTime : Date) : List<Activity>
+getAllActivities(employee : Employee) : List<Activity>
+getAllActivities(object : IActivityObject) : List<Activity>
+getAllActivities(schoolclass : Schoolclass) : List<Activity>
+getAllActivities(room : Room) : List<Activity>
+getAllActivities(employee : Employee, weekday : Weekday) : List<Activity>
+getAllActivities(employee : Employee, timePeriod : TimePeriod) : List<Activity>
+getAllActivities(schoolclass : Schoolclass, timePeriod : TimePeriod) : List<Activity>
+getAllActivities(room : Room, timePeriod : TimePeriod) : List<Activity>
+getAllCompoundLessons(lessonType : LessonType) : List<CompoundLesson>
+getAllLessons(lessonType : LessonType) : List<Lesson>
+getAllMeetings(meetingType : MeetingType) : List<Meeting>
+getEmployeeTimePeriods(employee : Employee) : List<EmployeeTimePeriods>
+getAllClassteams() : List<Classteam>
+getAllClassteams(employee : Employee) : List<Classteam>
+getOneClassteam(schoolclass : Schoolclass) : Classteam
+getById(clazz : Class<E>, id : Long) : E
+getTravelTimeList() : TravelTimeList
+getUsedChars(academicYear : AcademicYear) : List<Character>
-getAll(clazz : Class<? extends Serializable>, orderBy : String) : List<? extends Serializable>
-searchEmployees(clazz : Class<? extends Employee>, searchSymbol : String) : List<? extends Serializable>