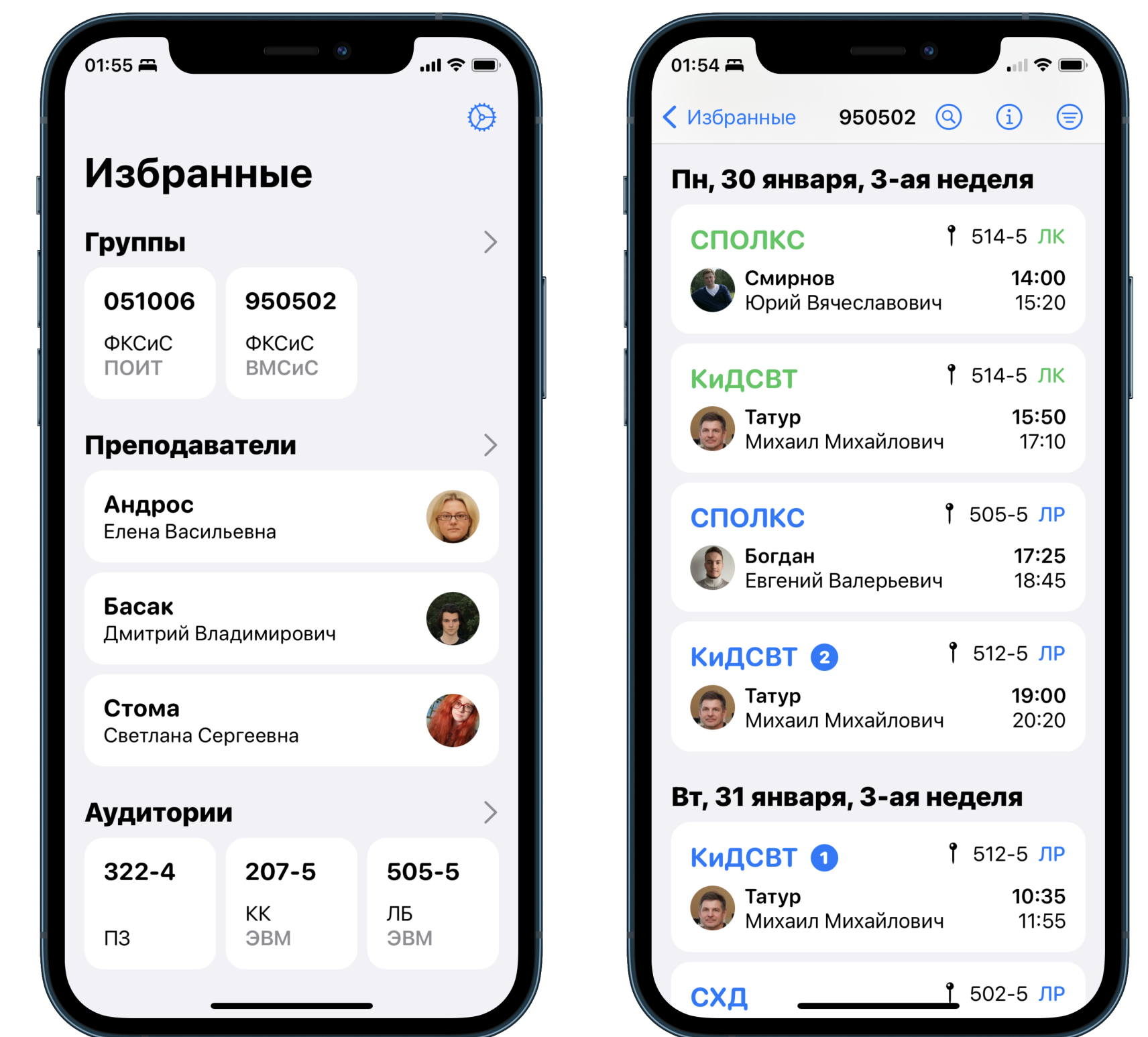


bsuirSchedule:

Кроссплатформенное приложение для iOS, macOS и iOS;
Расписание групп, преподавателей и кабинетов;
Максимум информации из API ИИС БГУИР;
Использование новейших Framework;
Адаптивный интерфейс.



Используемые технологии:



Swift

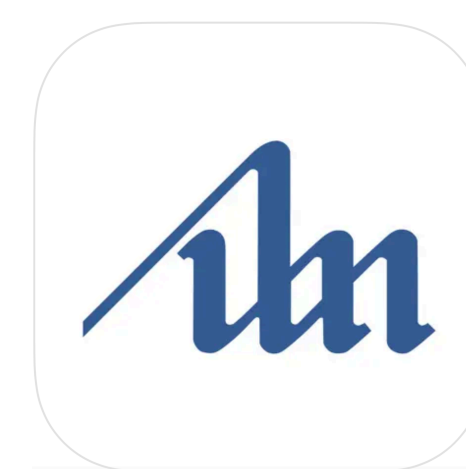


SwiftUI



CoreData

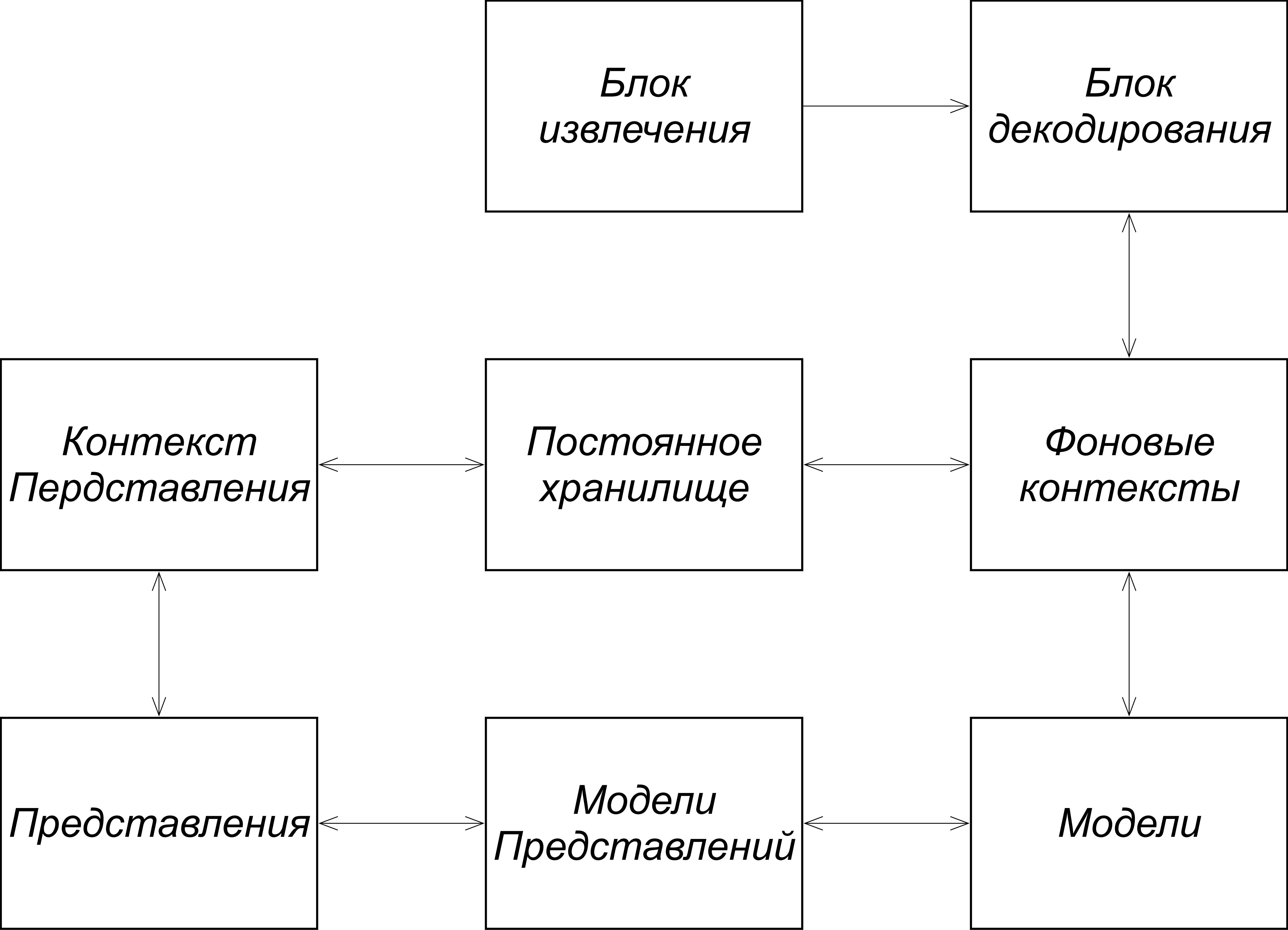
Аналоги:



Bsuir Schedule



BSUIR Timetable



Speciality

+ id : Int32

+ name : String

+ abbreviation : String

+ educationType : EducationType?

+ code : String?

+ groups : NSSet?

+ faculty : Faculty?

+ init(from: Decoder)

~ fetchAll()

+ addToGroups(NSSet)

+ removeFromGroups(NSSet)

+ addToGroups(Group)

+ removeFromGroups(Group)

~ formattedID() String

~ formattedDescription(Bool) String

~ update(Decoder)

~ updateFromGroupDecoder(Decoder)

EducationType

+ init(from:)

+ id : Int16

+ name : String

+ addToSpecialities(NSSet)

+ removeFromSpecialities(NSSet)

+ addToSpecialities(Speciality)

+ removeFromSpecialities(Speciality)

Group

~ title : String

~ flow : [Group]?

~ educationDegree : EducationDegree?

+ name : String

+ numberOfStudents : Int16

+ educationDegreeValue : Int16

+ course : Int16

+ favorite : Bool

+ lessonsUpdateDate : Date?

+ nickname : String?

+ speciality : Speciality?

+ educationStart : Date?

+ educationEnd : Date?

+ examsStart : Date?

+ examsEnd : Date?

+ lessons : NSSet?

~ employees : [Employee]?

+ init(from:)

+ addToLessons(NSSet)

+ removeFromLessons(NSSet)

+ addToLessons(Lesson)

+ removeFromLessons(Lesson)

~ update(Decoder)

- decodeGroup()

- decodeSpeciality()

- decodeLessons()

~ {static} defaultLessonSettings()

~ {static} fetchAll()

~ update()

~ {static} updateGroups(groups:)

- decodeEducationDates(Decoder)

~ fetchLastUpdateDate()

<<enumeration>> EducationDegree

~ bachelor

~ master

~ description : String

Employee

~ title : String

~ formattedName : String

~ departmentsArray : [Department]?

~ departmentsAbbreviations : [String]?

+ id : Int32

+ urlID : String?

+ firstName : String

+ middleName : String?

+ lastName : String

+ rank : String?

+ degree : String?

+ degreeAbbreviation : String?

+ departments : NSSet?

+ favorite : Bool

+ lessonsUpdateDate : Date?

+ educationStart : Date?

+ educationEnd : Date?

+ examsStart : Date?

+ examsEnd : Date?

+ photoLink : String?

+ photo : Data?

~ groups : [Group]?

+ lessons : NSSet?

+ init(from:)

~ {static} defaultLessonSettings()

~ {static} fetchAll()

~ update()

~ {static} updateEmployees([Employee])

~ fetchPhoto()

~ decodeEducationDates(Decoder)

~ fetchLastUpdateDate()

+ addToLessons(NSSet)

+ removeFromLessons(NSSet)

+ addToLessons(Lesson)

+ removeFromLessons(Lesson)

+ addToDepartments(NSSet)

+ removeFromDepartments(NSSet)

+ addToDepartments(Department)

+ removeFromDepartments(Department)

~ update(Decoder)

- decodeEmployee(Decoder)

- decodeLessons(Decoder)

- decodeDegree(Decoder)

Department

+ id : Int16

+ name : String?

~ formattedName : String

+ abbreviation : String

+ auditories : NSSet?

+ employees : NSSet?

+ init(Decoder)

+ init(Decoder, NSManagedObjectContext)

- formattedAbbreviationString() String

+ addToAuditories(NSSet)

+ removeFromAuditories(NSSet)

+ addToAuditories(Auditorium)

+ removeFromAuditories(Auditorium)

+ addToEmployees(NSSet)

+ removeFromEmployees(NSSet)

+ addToEmployees(Employee)

+ removeFromEmployees(Employee)

~ {static} fetchAll()

Auditorium

~ title : String

~ groups : [Group]?

+ floor : Int16

+ name : String

+ formattedName : String

+ note : String?

+ favorite : Bool

+ outsideUniversity : Bool

+ building : Int16

+ capacity : Int16

+ type : AuditoriumType?

+ department : Department?

+ lessons : NSSet?

+ init(from:)

+ init(from.in:)

~ {static} defaultLessonSettings()

+ addToLessons(NSSet)

+ removeFromLessons(NSSet)

+ addToLessons(Lesson)

+ removeFromLessons(Lesson)

~ {static} fetchAll()

~ update(Decoder)

- decodeBuilding(String)

- decodeName(String)

- formateName() String

AuditoriumType

+ id : Int16

+ name : String

+ abbreviation : String

+ auditories : NSSet?

+ init(Decoder)

+ addToAuditories(NSSet)

+ removeFromAuditories(NSSet)

+ addToAuditories(Auditorium)

+ removeFromAuditories(Auditorium)

NSManagedObjectsSection

~ title : String

~ id : String?

~ items : [ObjectType]

~ init(String, Hashable, [NSManagedObject])

AuditoriesViewModel

~ searchText

~ selectedSectionType : AuditoriumSectionType

~ selectedAuditoriumType : AuditoriumType?

~ menuDefaultRules : [Bool]

~ update()

~ calculatePredicate()

GroupsViewModel

~ searchText: String

~ selectedSectionType : GroupSectionType

~ selectedFaculty : Faculty?

~ selectedSpecialtyEducationType : EducationType?

~ selectedEducationDegree : EducationDegree?

~ selectedCourse : Int16?

~ menuDefaultRules : [Bool]

~ {static} update()

~ calculatePredicate()

GroupViewModel

~ nicknameString : String

~ submitNickname(String)

EmployeesViewModel

~ {static} update()

~ calculatePredicate(String)

Faculty

+ id : Int16

+ name : String?

+ abbreviation : String?

+ specialities : NSSet?

+ init(Decoder)

+ init(Int16, String?, String?, NSManagedObjectContext)

+ addToSpecialities(NSSet)

+ removeFromSpecialities(NSSet)

+ addToSpecialities(Speciality)

+ removeFromSpecialities(Speciality)

~ {static} fetchAll()

~ update(Decoder)

~ updateFromGroupDecoder(Decoder)

LessonViewConfiguration

~ showAbbreviation : Bool

~ showGroups : Bool

~ showEmployees : Bool

~ showWeeks : Bool

~ showDates : Bool

~ showDate : Bool

~ init(Bool, Bool, Bool, Bool, Bool, Bool, Bool)

~ {static} ==(:, :)

ScheduleSection

~ id : String

~ date : Date?

~ educationWeek : Int

~ weekday : WeekDay

~ lessons : [Lesson]

~ title : String

~ dateTitle : String?

~ weekTitle : String

~ dateString : String?

~ description : String

~ init(Date, Int16, Int16, [Lesson])

~ {static} ==(ScheduleSection, ScheduleSection)

ScheduleViewModel

~ selectedSectionType : ScheduleSectionType

~ showScrollView

~ targetSection : ScheduleSection?

~ scrollWithAnimation

~ searchText

~ showSearchField

~ selectedDate

~ showDatePicker

~ selectedSubgroup : Int?

~ sections : [ScheduleSection]?

~ filteredSections : [ScheduleSection]?

~ cancellables

~ defaultRules : Bool

~ init()

~ addSelectedDateSubscriber()

~ addSearchTextSubscriber()

~ updateSections([Lessons])

~ updateFilteredSections()

~ scrollTo(ScheduleSection, ScrollViewProxy)

~ scrollToDate(Bool)

Lesson

+ description : String

+ subject : String?

+ abbreviation : String

+ note : String?

+ dateString : String

+ weekday : Int16

+ weeks : [Int]

~ weeksDescription : String?

+ startLessonDate : Date?

+ startLessonDateString : String

+ endLessonDate : Date?

+ timeStart : String

+ timeEnd : String

+ groups : NSSet?

+ subgroup : Int16

+ employees : NSSet?

+ employeesIDs : [Int32]

+ auditoriesNames : [String]

~ date : Date?

~ dateRange : ClosedRange<Date>?

~ timeRange : ClosedRange<Date>

+ type : LessonType?

+ auditories : NSSet?

+ init(from:)

- decodeLesson(Decoder)

- decodeDate(Decoder)

- decodeAnnouncement(Decoder)

- decodeAuditories(Decoder)

- decodeEmployees(Decoder)

- decodeGroups(Decoder)

+ addToGroups(NSSet)

+ removeFromGroups(NSSet)

+ addToGroups(Group)

+ removeFromGroups(Group)

+ addToEmployees(NSSet)

+ removeFromEmployees(NSSet)

+ addToEmployees(Employee)

+ removeFromEmployees(Employee)

+ addToAuditories(NSSet)

+ removeFromAuditories(NSSet)

+ addToAuditories(Auditorium)

+ removeFromAuditories(Auditorium)

~ id(String) String

LessonType

+ id : Int16

+ abbreviation : String

+ name : String

+ colorData : Data

+ color : Color

+ init(from:)

~ {static} initDefaultLessonTypes()

~ formattedName(Bool) String

