



**UNIWERSYTET TECHNOLOGICZNO-PRZYRODNICZY  
IM. J. I J. ŚNIADECKICH W BYDGOSZCZY**

**WYDZIAŁ TELEKOMUNIKACJI, INFORMATYKI I ELEKTROTECHNIKI  
ZAKŁAD TECHNIKI CYFROWEJ**

**PROGRAMOWANIE W ŚRODOWISKU WINDOWS**

---

## **DOKUMENTACJA PROJEKTU**

**EWIDENCJA ŚRODKÓW TRWAŁYCH**

**AUTORZY :**  
**WITOLD TABACZYŃSKI**  
**SEBASTIAN STENCEL**  
**KIERUNEK :**  
**INFORMATYKA STOSOWANA**  
**GRUPA :**  
**II**

**SEMESTR :**  
**V**  
**ROK AKADEMICKI :**  
**2017/2018**  
**TRYB STUDIÓW :**  
**NIESTACJONARNY**

## WYMAGANIA

SYSTEM EWIDENCJI NA POTRZEBY UCZELNI. POZWALA NA:

- MOŻLIWOŚĆ DODANIA PRZEDMIOTU;
- MOŻLIWOŚĆ DODANIA OSOBY;
- MOŻLIWOŚĆ DODANIA MIEJSCA;
- MOŻLIWOŚĆ DODANIA JEDNOSTKI;
- MOŻLIWOŚĆ PRZYPISANIA MIEJSCA DO PRZEDMIOTU;
- MOŻLIWOŚĆ PRZYPISANIA OSOBY DO PRZEDMIOTU;
- MOŻLIWOŚĆ PRZYPISANIA LOKALIZACJI DO PRZEDMIOTU;
- MOŻLIWOŚĆ PODGLĄDU PRZEDMIOTÓW NALEŻĄCYCH DO JEDNOSTKI;
- MOŻLIWOŚĆ PODGLĄDU PRZEDMIOTU PRZY UŻYCIU JEGO KODU;
- MOŻLIWOŚĆ PODGLĄDU WSZYSTKICH PRZEDMIOTÓW;
- MOŻLIWOŚĆ PODGLĄDU WSZYSTKICH LOKALIZACJI;
- MOŻLIWOŚĆ PODGLĄDU WSZYSTKICH JEDNOSTEK;
- MOŻLIWOŚĆ PODGLĄDU OSOBY PRZY UŻYCIU JEJ ADRESU EMAIL;
- MOŻLIWOŚĆ PODGLĄDU OSÓB NALEŻĄCYCH DO DANEJ JEDNOSTKI;
- MOŻLIWOŚĆ ZMIANY DANYCH PRZEDMIOTU;
- MOŻLIWOŚĆ ZMIANY DANYCH MIEJSCA;
- MOŻLIWOŚĆ ZMIANY DANYCH OSOBY;
- MOŻLIWOŚĆ ZMIANY DANYCH JEDNOSTKI;
- MOŻLIWOŚĆ USUNIĘCIA PRZEDMIOTU;
- MOŻLIWOŚĆ USUNIĘCIA MIEJSCA;
- MOŻLIWOŚĆ USUNIĘCIA OSOBY;
- MOŻLIWOŚĆ USUNIĘCIA JEDNOSTKI;

## INTERFEJS UŻYTKOWNIKA

- INTERFEJS GŁÓWNY

### Create Read Update Delete

|                             |  |
|-----------------------------|--|
| Add item                    |  |
| Add person                  |  |
| Add location                |  |
| Add unit                    |  |
| Get items by unitShortName  | <input type="text"/> Show items of that unit     |
| Get item by code            | <input type="text"/> Get this item               |
| Get all items               | Get all items                                    |
| Get all units               | Get all units                                    |
| Get all locations           | Get all locations                                |
| Get person by unitShortName | <input type="text"/> Get person by unitShortName |
| Update item by code         | <input type="text"/> Update this item            |
| Update location by name     | <input type="text"/> Update this location        |
| Update person by email      | <input type="text"/> Update this person          |
| Update unit by shortName    | <input type="text"/> Update this unit            |
| Delete item by code         | <input type="text"/> Delete this by code         |
| Delete location by name     | <input type="text"/> Delete this by name         |
| Delete person by email      | <input type="text"/> Delete this by email        |
| Delete unit by shortName    | <input type="text"/> Delete this by shortName    |

- DODAWANIE/EDYCJA PRZEDMIOTU

### Add/Update

|               |                      |
|---------------|----------------------|
| name          | <input type="text"/> |
| code          | <input type="text"/> |
| description   | <input type="text"/> |
| unitShortName | <input type="text"/> |
| personEmail   | <input type="text"/> |
| locationName  | <input type="text"/> |
| Zapisz        |                      |

- DODAWANIE/EDYCJA OSOBY

### Add/Update

|               |                      |
|---------------|----------------------|
| name          | <input type="text"/> |
| lastName      | <input type="text"/> |
| email         | <input type="text"/> |
| unitShortName | <input type="text"/> |
| Zapisz        |                      |

- DODAWANIE/EDYCJA MIEJSCA

# Add/Update

name

description

building

0.0

room

0.0

floor

0

Zapisz

- DODAWANIE/EDYCJA JEDNOSTKI

# Add/Update

name

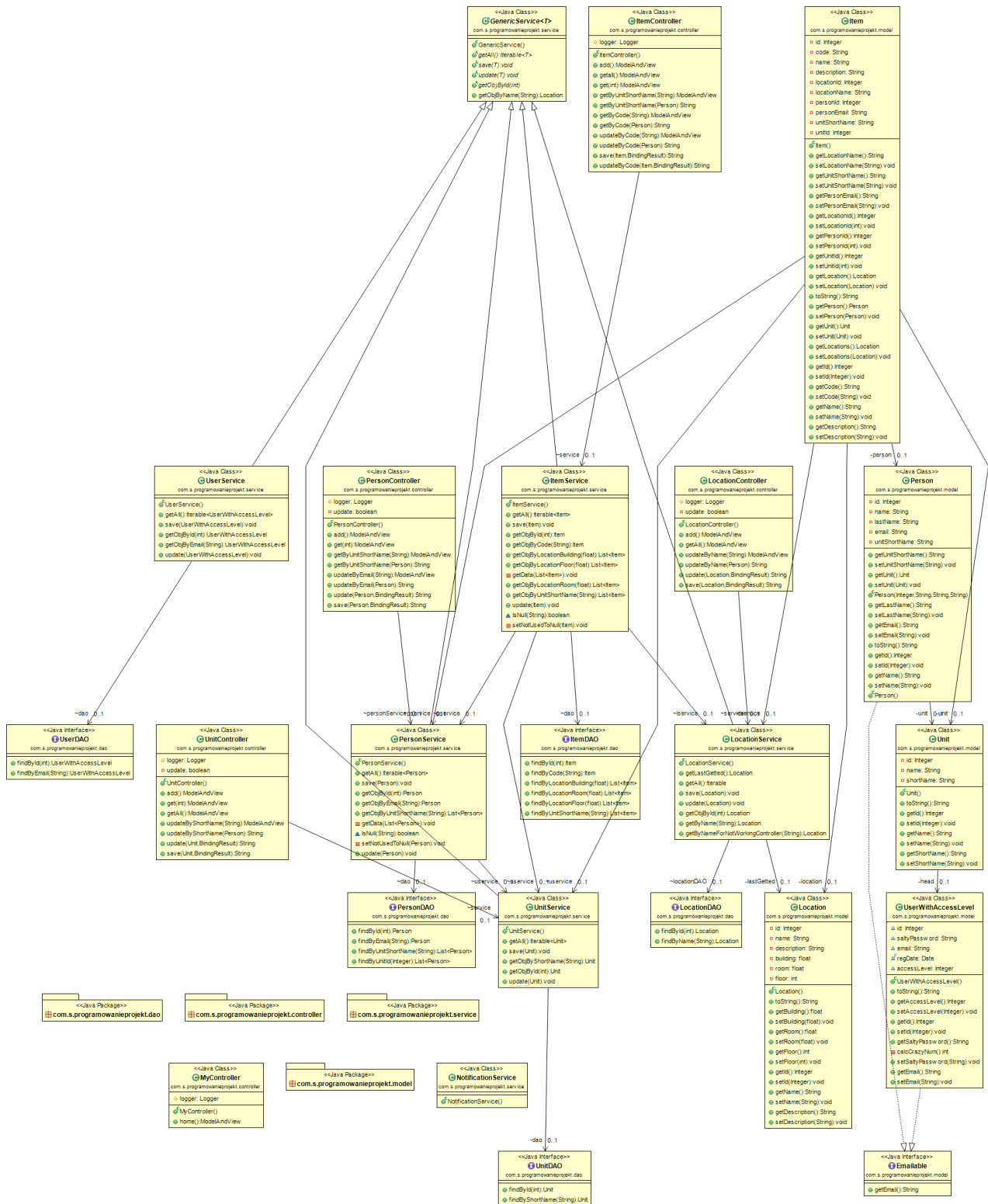
shortName

Zapisz

- PRZYKŁADOWY LISTING (PRZEDMIOTY JEDNOSTKI)

|  |
|--|
| Item{id=64, code=KOMPUTER000003, name=komputer, description=zwykly laptop, location=null, person=Person{id=5, name=name, lastName=lastName, email=email@email.email, unit=Unit{id=11, name=Wydział Telekomunikacji, Informatyki i Elektrotechniki, shortName=WTIE}}, unit=Unit{id=11, name=Wydział Telekomunikacji, Informatyki i Elektrotechniki, shortName=WTIE}}  |
| Item{id=63, code=KOMPUTER000002, name=komputer, description=zwykly laptop, location=Location{id=53, name=administracja, description=administracja, building=2.2, room=407.0, floor=3}, person=Person{id=5, name=name, lastName=lastName, email=email@email.email, unit=Unit{id=11, name=Wydział Telekomunikacji, Informatyki i Elektrotechniki, shortName=WTIE}}, unit=Unit{id=11, name=Wydział Telekomunikacji, Informatyki i Elektrotechniki, shortName=WTIE}}                         |
| Item{id=68, code=KOMPUTER000005, name=komputer, description=zwykly PC, upgraded: + 16 GB RAM, location=Location{id=53, name=administracja, description=administracja, building=2.2, room=407.0, floor=3}, person=Person{id=5, name=name, lastName=lastName, email=email@email.email, unit=Unit{id=11, name=Wydział Telekomunikacji, Informatyki i Elektrotechniki, shortName=WTIE}}, unit=Unit{id=11, name=Wydział Telekomunikacji, Informatyki i Elektrotechniki, shortName=WTIE}}      |
| Item{id=70, code=KOMPUTER000099, name=laptop, description=I9 + 16 GB, location=null, person=null, unit=Unit{id=11, name=Wydział Telekomunikacji, Informatyki i Elektrotechniki, shortName=WTIE}}   |
| Item{id=62, code=KOMPUTER000001, name=komputer, description=zwykly PC, upgraded: wincej rdzeniuff, location=Location{id=53, name=administracja, description=administracja, building=2.2, room=407.0, floor=3}, person=Person{id=5, name=name, lastName=lastName, email=email@email.email, unit=Unit{id=11, name=Wydział Telekomunikacji, Informatyki i Elektrotechniki, shortName=WTIE}}, unit=Unit{id=11, name=Wydział Telekomunikacji, Informatyki i Elektrotechniki, shortName=WTIE}} |

## DIAGRAM KLAS



## PRZYKŁADOWY KOD

### KONTROLER

```
@Controller
@RequestMapping("/ITEM")
public class ItemController {
    @Autowired //@Qualifier("ITEMSERVICE")
    ITEMService service;
    @RequestMapping("/ADD")
    public ModelAndView add() {
        ModelAndView model = new
        ModelAndView("ADDITEM");
        model.addObject("ITEM", new ITEM());
        return model;
    }
    @RequestMapping("/GET/ALL")
    public ModelAndView getAll() {
        ModelAndView model = new
        ModelAndView("RESPONSE");
        model.addObject("OBJECT", service.getAll());
        return model;
    }
    @RequestMapping("/GET/{ID}")
    public ModelAndView get(@PathVariable("ID") int id)
    {
        ModelAndView model = new
        ModelAndView("RESPONSE");
        model.addObject("OBJECT", service.getObjById(id));
        return model;
    }
    @RequestMapping("/GETBYUNITSHORTNAME/{SHORTNAME}")
    public ModelAndView
    getByUnitShortName(@PathVariable("SHORTNAME")
    String shortName) {
        ModelAndView model = new
        ModelAndView("RESPONSE");
        model.addObject("OBJECT",
        service.getObjByUnitShortName(shortName));
        return model;
    }
    @RequestMapping("/GETBYUNITSHORTNAME")
    public String getByUnitShortName(Person p) {
        String name = p.getName();
        return
        "REDIRECT:/ITEM/GETBYUNITSHORTNAME/"+name+".htm";
    }
    @RequestMapping("/GET/BYCODE/{CODE}")
    public ModelAndView
    getByCode(@PathVariable("CODE") String code) {
        ModelAndView model = new
        ModelAndView("SINGLERESPONSE");
        model.addObject("OBJECT",
        service.getObjByCode(code));
        return model;
    }
    @RequestMapping("/GET/BYCODE")
    public String getByCode(Person p) {
        String name = p.getName();
        return "REDIRECT:/ITEM/GET/BYCODE/"+name+".htm";
    }
}
```

```
@RequestMapping("/UPDATE/BYCODE/{CODE}")
public ModelAndView
updateByCode(@PathVariable("CODE") String code){
    ModelAndView model = new
    ModelAndView("ADDITEM");
    ITEM item = service.getObjByCode(code);
    model.addObject("ITEM", item);
    return model;
}
@RequestMapping("/UPDATE/BYCODE")
public String updateByCode(Person p) {
    String name = p.getName();
    return
    "REDIRECT:/ITEM//UPDATE/BYCODE/"+name+".htm";
}
@RequestMapping(value = "/SAVE", method =
RequestMethod.POST)
public String save(@Valid Item p,BindingResult
BindingResult) {
    if(BindingResult.hasErrors())
    {
        for (ObjectError e :
        BindingResult.getAllErrors())
            System.out.println(e);
        return "ADDITEM";
    }
    service.save(p);
    return "REDIRECT:/HOME.HTM";
}
@RequestMapping(value = "/UPDATE/BYCODE/SAVE",
method = RequestMethod.POST)
public String updateByCode(@Valid Item
p,BindingResult BindingResult) {
    if(BindingResult.hasErrors())
    {
        for (ObjectError e :
        BindingResult.getAllErrors())
            System.out.println(e);
        return "ADDITEM";
    }
    service.update(p);
    return "REDIRECT:/HOME.HTM";
}
@RequestMapping("/DELETEBYCODE")
public String deleteByCode(Person p) {
    String name = p.getName();
    service.deleteByCode(name);
    return "REDIRECT:/HOME.HTM";
}
}
```

## DAO

```
PUBLIC INTERFACE ITEMDAO EXTENDS CRUDREPOSITORY<ITEM, INTEGER>{  
    PUBLIC ITEM FINDBYID(INT ID);  
    PUBLIC ITEM FINDBYCODE(String CODE);  
    PUBLIC List<ITEM> FINDBYLOCATIONBUILDING(FLOAT BUILDING);  
    PUBLIC List<ITEM> FINDBYLOCATIONROOM(FLOAT ROOM);  
    PUBLIC List<ITEM> FINDBYLOCATIONFLOOR(FLOAT FLOOR);  
    PUBLIC List<ITEM> FINDBYUNITSHORTNAME(String SHORTNAME);  
}
```

## MODEL

```
PUBLIC CLASS ITEM {

    @AutOWired @Transient
    PERSONSERVICE pSERVICE;
    @AutOWired @Transient
    UNITSERVICE uSERVICE;
    @AutOWired @Transient
    LOCATIONSERVICE lSERVICE;

    @Id @GeneratedValue(strategy=GenerationType.AUTO)
    PRIVATE INTEGER ID;
    @NotEmpty @Column(unique=true)
    PRIVATE STRING CODE;
    @NotEmpty
    PRIVATE STRING NAME;
    @NotEmpty
    PRIVATE STRING DESCRIPTION;
    @ManyToOne
    PRIVATE LOCATION LOCATION;
    @ManyToOne
    PRIVATE PERSON PERSON;
    @ManyToOne
    PRIVATE UNIT UNIT;
    PRIVATE INTEGER LOCATIONID;
    PRIVATE STRING LOCATIONNAME;
    PUBLIC STRING getLocationName() {
        RETURN LOCATIONNAME;
    }
    PUBLIC VOID setLocationName(STRING LOCATIONNAME) {
        IF(LOCATION==NULL)
            LOCATION=NEW LOCATION();
        LOCATION.setName(LOCATIONNAME);
        THIS.LOCATIONNAME = LOCATIONNAME;
    }
    PRIVATE INTEGER PERSONID;
    @Email
    PRIVATE STRING PERSONEMAIL;
    PRIVATE STRING UNITSHORTNAME;
    PUBLIC STRING getUnitShortName() {
        RETURN UNITSHORTNAME;
    }
    PUBLIC VOID setUnitShortName(STRING UNITSHORTNAME) {
        IF(UNIT== NULL)
            UNIT=NEW UNIT();
        UNIT.setShortName(UNITSHORTNAME);
        THIS.UNITSHORTNAME = UNITSHORTNAME;
    }
    PUBLIC STRING getPersonEmail() {
        RETURN PERSONEMAIL;
    }
    PUBLIC VOID setPersonEmail(STRING PERSONEMAIL) {
        IF(PERSON==NULL)
            PERSON=NEW PERSON();
        PERSON.setEmail(PERSONEMAIL);
        THIS.PERSONEMAIL = PERSONEMAIL;
    }
    PRIVATE INTEGER UNITID;
    PUBLIC INTEGER getLocationId() {
        RETURN LOCATIONID;
    }
    PUBLIC VOID setLocationId(INT LOCATIONID) {
        IF(LOCATION==NULL)
            LOCATION= NEW LOCATION();
        LOCATION.setId(LOCATIONID);
        THIS.LOCATIONID = LOCATIONID;
    }
    PUBLIC INTEGER getPersonId() {
        RETURN PERSONID;
    }
    PUBLIC VOID setPersonId(INT PERSONID) {
        IF(PERSON==NULL)
            PERSON=NEW PERSON();
        PERSON.setId(PERSONID);
        THIS.PERSONID = PERSONID;
    }
    PUBLIC INTEGER getUnitId() {
        RETURN UNITID;
    }
}
```

```
PUBLIC VOID setUnitId(INT UNITID) {
    IF(UNIT==NULL)
        UNIT=NEW UNIT();
    UNIT.setID(UNITID);
    THIS.UNITID = UNITID;
}
PUBLIC LOCATION getLocation() {
    RETURN LOCATION;
}
PUBLIC VOID setLocation(LOCATION LOCATION) {
    IF(LOCATION!=NULL){
        LOCATIONNAME=LOCATION.getName();
        LOCATIONID=LOCATION.getId();
    }
    THIS.LOCATION = LOCATION;
}
@Override
PUBLIC STRING toString() {
    RETURN "Item{" + "ID=" + ID + ", CODE=" + CODE + ", NAME=" + NAME + ",
DESCRIPTION=" + DESCRIPTION + ", LOCATION=" + LOCATION + ", PERSON=" +
PERSON + ", UNIT=" + UNIT + '}';
}
PUBLIC PERSON getPerson() {
    RETURN PERSON;
}
PUBLIC VOID setPerson(PERSON PERSON) {
    IF(PERSON!=NULL){
        PERSONEMAIL=PERSON.getEmail();
        PERSONID=PERSON.getId();
    }
    THIS.PERSON = PERSON;
}
PUBLIC UNIT getUnit() {
    RETURN UNIT;
}
PUBLIC VOID setUnit(UNIT UNIT) {
    IF(UNIT!=NULL){
        UNITSHORTNAME=UNIT.getShortName();
        UNITID=UNIT.getId();
    }
    THIS.UNIT = UNIT;
}
PUBLIC LOCATION getLocations() {
    RETURN LOCATION;
}
PUBLIC VOID setLocations(LOCATION LOCATIONS) {
    THIS.LOCATION = LOCATIONS;
}
PUBLIC INTEGER getId() {
    RETURN ID;
}
PUBLIC VOID setId(INTEGER ID) {
    THIS.ID = ID;
}
PUBLIC STRING getCode() {
    RETURN CODE;
}
PUBLIC VOID setCode(STRING CODE) {
    THIS.CODE = CODE;
}
PUBLIC STRING getName() {
    RETURN NAME;
}
PUBLIC VOID setName(STRING NAME) {
    THIS.NAME = NAME;
}
PUBLIC STRING getDescription() {
    RETURN DESCRIPTION;
}
PUBLIC VOID setDescription(STRING DESCRIPTION) {
    THIS.DESCRPTION = DESCRIPTION;
}
}
```



## SERVICE

|   |  |
|---|--|
| <pre> @Service PUBLIC CLASS ITEMService EXTENDS GenericService&lt;Item&gt;{     @Autowired     ItemDAO dao;     @Autowired     UnitService userService;     @Autowired     PersonService pService;     @Autowired     LocationService lService;     @Override     PUBLIC Iterable&lt;Item&gt; getAll() {         List&lt;Item&gt; ans = new ArrayList&lt;Item&gt;();         for(Item item :dao.findAll())             ans.add(item);         getData(ans);         return ans;     }     @Override     PUBLIC void save(Item toSave) {         setNotUsedToNull(toSave);         dao.save(toSave);     }     @Override     PUBLIC Item getObjById(int objId) {         return dao.findById(objId);     }     PUBLIC Item getObjByCode(String code) {         return dao.findByCode(code);     }     PUBLIC List&lt;Item&gt; getObjByLocationBuilding(float building) {         List&lt;Item&gt; ans = dao.findByLocationBuilding(building);         getData(ans);         return ans;     }     PUBLIC List&lt;Item&gt; getObjByLocationFloor(float floor) {         List&lt;Item&gt; ans = dao.findByLocationFloor(floor);         getData(ans);         return ans;     }     PRIVATE void getData(List&lt;Item&gt; ans) {         for(Item item :ans)         {             setNotUsedToNull(item);             item.setId(getObjByCode(item.getCode()).getId());             if(!IsNull(item.getUnitShortName()))  item.setUnit(userService.getObjByShortName(item.getUnit ShortName()));             if(!IsNull(item.getPersonEmail()))  item.setPerson(pService.getObjByEmail(item.getPersonE mail()));             if(!IsNull(item.getLocationName()))  item.setLocation(lService.getByname(item.getLocation Name()));         }     } </pre> | <pre> PUBLIC List&lt;Item&gt; getObjByLocationRoom(float room) {         List&lt;Item&gt; ans = dao.findByLocationRoom(room);         getData(ans);         return ans;     }     PUBLIC List&lt;Item&gt; getObjByUnitShortName(String shortName) {         List&lt;Item&gt; ans = dao.findByUnitShortName(shortName);         getData(ans);         return ans;     }     @Override     PUBLIC void update(Item toUpdate) {         setNotUsedToNull(toUpdate);  toUpdate.setId(getObjByCode(toUpdate.getCode()).getI d());         if(!IsNull(toUpdate.getUnitShortName()))  toUpdate.setUnit(userService.getObjByShortName(toUpdat e.getUnitShortName()));         if(!IsNull(toUpdate.getPersonEmail()))  toUpdate.setPerson(pService.getObjByEmail(toUpdate.g etPersonEmail()));         if(!IsNull(toUpdate.getLocationName()))  toUpdate.setLocation(lService.getByname(toUpdate.get LocationName()));         dao.save(toUpdate);     }     PUBLIC void deleteByCode(String code) {         dao.delete(getObjByCode(code));     }     BOOLEAN IsNull(String tmp){         return (tmp==null    tmp.equals(""));     }     PRIVATE void setNotUsedToNull(Item toSave) {         if(toSave.getLocation()! =null)if(toSave.getLocation().getId()==null)toSave.set Location(null);         if(toSave.getPerson()! =null)if(toSave.getPerson().getId()==null)toSave.setPe rson(null);         if(toSave.getUnit()! =null)if(toSave.getUnit().getId()==null)toSave.setUnit( null);         if(!IsNull(toSave.getPersonEmail()))  toSave.setPerson(pService.getObjByEmail(toSave.getPe rsonEmail()));         if(!IsNull(toSave.getLocationName()))  toSave.setLocation(lService.getByname(toSave.getLoca tionName()));         if(!IsNull(toSave.getUnitShortName()))  toSave.setUnit(userService.getObjByShortName(toSave.g etUnitShortName()));     } </pre> |
|---|--|