

My Project

1.01

Generated by Doxygen 1.8.18

1 Hierarchical Index	1
1.1 Class Hierarchy	1
2 Class Index	3
2.1 Class List	3
3 Class Documentation	5
3.1 hospital.servlets.filter.AuthFilter Class Reference	5
3.1.1 Detailed Description	5
3.1.2 Constructor & Destructor Documentation	5
3.1.2.1 AuthFilter() [1/2]	6
3.1.2.2 AuthFilter() [2/2]	6
3.1.3 Member Function Documentation	6
3.1.3.1 destroy()	6
3.1.3.2 doFilter()	6
3.1.3.3 init()	7
3.2 hospital.utils.factories.CommandFactory Class Reference	7
3.2.1 Detailed Description	7
3.2.2 Member Function Documentation	8
3.2.2.1 createCommand()	8
3.3 hospital.utils.ConnectionPool Class Reference	8
3.3.1 Detailed Description	8
3.3.2 Member Function Documentation	9
3.3.2.1 getConnection()	9
3.3.3 Member Data Documentation	9
3.3.3.1 PASSWORD	9
3.3.3.2 URL	9
3.3.3.3 USER	9
3.4 hospital.utils.doctorcommands.CreateDiagnosisCommand Class Reference	10
3.4.1 Detailed Description	10
3.4.2 Constructor & Destructor Documentation	10
3.4.2.1 CreateDiagnosisCommand() [1/2]	10
3.4.2.2 CreateDiagnosisCommand() [2/2]	11
3.4.3 Member Function Documentation	11
3.4.3.1 execute()	11
3.5 hospital.utils.doctorcommands.CreatePatientsCommand Class Reference	11
3.5.1 Detailed Description	12
3.5.2 Constructor & Destructor Documentation	12
3.5.2.1 CreatePatientsCommand() [1/2]	12
3.5.2.2 CreatePatientsCommand() [2/2]	12
3.5.3 Member Function Documentation	13
3.5.3.1 execute()	13
3.6 hospital.utils.doctorcommands.CreatePrescriptionsCommand Class Reference	13

3.6.1 Detailed Description	14
3.6.2 Constructor & Destructor Documentation	14
3.6.2.1 CreatePrescriptionsCommand() [1/2]	14
3.6.2.2 CreatePrescriptionsCommand() [2/2]	14
3.6.3 Member Function Documentation	14
3.6.3.1 execute()	14
3.7 hospital.servlets.CreateUserServlet Class Reference	15
3.7.1 Detailed Description	16
3.7.2 Constructor & Destructor Documentation	16
3.7.2.1 CreateUserServlet() [1/2]	16
3.7.2.2 CreateUserServlet() [2/2]	16
3.7.3 Member Function Documentation	16
3.7.3.1 doGet()	16
3.7.3.2 doPost()	17
3.7.3.3 inputValidator()	17
3.8 hospital.utils.factories.DAOFactory Class Reference	18
3.8.1 Detailed Description	18
3.8.2 Member Function Documentation	18
3.8.2.1 createDiagnosisDao()	18
3.8.2.2 createPatientDataDao()	19
3.8.2.3 createPrescriptionDao()	19
3.8.2.4 createRoleDao()	19
3.8.2.5 createSingleTransaction()	19
3.8.2.6 createUserDao()	20
3.9 hospital.entity.diagnosis.Diagnosis Class Reference	20
3.9.1 Detailed Description	20
3.9.2 Constructor & Destructor Documentation	20
3.9.2.1 Diagnosis()	20
3.9.3 Member Function Documentation	21
3.9.3.1 getDifficulty()	21
3.9.3.2 getId()	21
3.9.3.3 getName()	21
3.9.3.4 setDifficulty()	21
3.9.3.5 setId()	22
3.9.3.6 setName()	22
3.9.4 Member Data Documentation	22
3.9.4.1 DIFFICULTY_COLUMN	22
3.9.4.2 ID_COLUMN	22
3.9.4.3 NAME_COLUMN	23
3.9.4.4 TABLE_NAME	23
3.10 hospital.dao.impl.DiagnosisDAOImpl Class Reference	23
3.10.1 Detailed Description	23

3.10.2 Member Function Documentation	23
3.10.2.1 findAll()	23
3.10.2.2 findById()	24
3.11 hospital.utils.queries.DiagnosisQueries Interface Reference	24
3.11.1 Detailed Description	24
3.11.2 Member Data Documentation	24
3.11.2.1 FIND_ALL	24
3.11.2.2 FIND_BY_ID	25
3.12 hospital.utils.doctorcommands.DoctorPageCommand Class Reference	25
3.12.1 Detailed Description	25
3.12.2 Constructor & Destructor Documentation	25
3.12.2.1 DoctorPageCommand() [1/2]	26
3.12.2.2 DoctorPageCommand() [2/2]	26
3.12.3 Member Function Documentation	26
3.12.3.1 execute()	26
3.13 hospital.servlets.doctorServlets.DoctorPageServlet Class Reference	27
3.13.1 Detailed Description	27
3.13.2 Member Function Documentation	27
3.13.2.1 doPost()	27
3.14 hospital.utils.doctorcommands.DoPrescriptionCommand Class Reference	28
3.14.1 Detailed Description	28
3.14.2 Constructor & Destructor Documentation	28
3.14.2.1 DoPrescriptionCommand() [1/2]	29
3.14.2.2 DoPrescriptionCommand() [2/2]	29
3.14.3 Member Function Documentation	29
3.14.3.1 execute()	29
3.15 hospital.utils.InputDataCheck Class Reference	30
3.15.1 Detailed Description	30
3.15.2 Member Function Documentation	30
3.15.2.1 inputCheck()	30
3.16 hospital.servlets.auth.LogoutServlet Class Reference	31
3.16.1 Detailed Description	31
3.16.2 Member Function Documentation	31
3.16.2.1 doGet()	31
3.17 hospital.utils.nursecommands.NurseDoPrescrCommand Class Reference	32
3.17.1 Detailed Description	32
3.17.2 Constructor & Destructor Documentation	32
3.17.2.1 NurseDoPrescrCommand() [1/2]	33
3.17.2.2 NurseDoPrescrCommand() [2/2]	33
3.17.3 Member Function Documentation	33
3.17.3.1 execute()	33
3.18 hospital.utils.nursecommands.NurseFindPatientsCommand Class Reference	34

3.18.1 Detailed Description	34
3.18.2 Constructor & Destructor Documentation	34
3.18.2.1 NurseFindPatientsCommand() [1/2]	34
3.18.2.2 NurseFindPatientsCommand() [2/2]	35
3.18.3 Member Function Documentation	35
3.18.3.1 execute()	35
3.19 hospital.utils.nursecommands.NursePageCommand Class Reference	35
3.19.1 Detailed Description	36
3.19.2 Constructor & Destructor Documentation	36
3.19.2.1 NursePageCommand() [1/2]	36
3.19.2.2 NursePageCommand() [2/2]	36
3.19.3 Member Function Documentation	37
3.19.3.1 execute()	37
3.20 hospital.servlets.nurseServlets.NursePageServlet Class Reference	37
3.20.1 Detailed Description	38
3.20.2 Member Function Documentation	38
3.20.2.1 doPost()	38
3.21 hospital.utils.PasswordEncryptorSHA256 Class Reference	38
3.21.1 Detailed Description	39
3.21.2 Member Function Documentation	39
3.21.2.1 encryptPasswordWithSHA256()	39
3.22 hospital.entity.PatientData Class Reference	39
3.22.1 Detailed Description	40
3.22.2 Constructor & Destructor Documentation	40
3.22.2.1 PatientData() [1/2]	40
3.22.2.2 PatientData() [2/2]	41
3.22.3 Member Function Documentation	41
3.22.3.1 getCurrentPrescriptions()	41
3.22.3.2 getDiagnosis()	42
3.22.3.3 getDoctor()	42
3.22.3.4 getId()	42
3.22.3.5 getNurse()	42
3.22.3.6 getPatient()	43
3.22.3.7 getPatientStatus()	43
3.22.3.8 getPrescriptionHistory()	43
3.22.3.9 setCurrentPrescriptions()	43
3.22.3.10 setDiagnosis()	44
3.22.3.11 setDoctor()	44
3.22.3.12 setId()	44
3.22.3.13 setNurse()	44
3.22.3.14 setPatient()	45
3.22.3.15 setPatientStatus()	45

3.22.3.16 setPrescriptionHistory()	45
3.22.4 Member Data Documentation	45
3.22.4.1 DIAGNOSIS_COLUMN	46
3.22.4.2 DOCTOR_COLUMN	46
3.22.4.3 ID_COLUMN	46
3.22.4.4 NURSE_COLUMN	46
3.22.4.5 PATIENT_COLUMN	46
3.22.4.6 STATUS_COLUMN	46
3.22.4.7 TABLE_NAME	46
3.23 hospital.dao.impl.PatientDataImpl Class Reference	47
3.23.1 Detailed Description	47
3.23.2 Member Function Documentation	47
3.23.2.1 findAll()	47
3.23.2.2 findAllWhereNoDoc()	48
3.23.2.3 findAllWhereNoNurse()	48
3.23.2.4 findByPatientId()	48
3.23.2.5 findDoctorPatients()	48
3.23.2.6 findNursePatients()	49
3.23.2.7 insert()	49
3.23.2.8 updateDiagnosis()	49
3.23.2.9 updateDoctor()	49
3.23.2.10 updateNurse()	49
3.23.2.11 updatePatientStatus()	50
3.24 hospital.utils.utilsForDBData.PatientDataManipulations Class Reference	50
3.24.1 Detailed Description	50
3.24.2 Constructor & Destructor Documentation	50
3.24.2.1 PatientDataManipulations() [1/2]	51
3.24.2.2 PatientDataManipulations() [2/2]	51
3.24.3 Member Function Documentation	51
3.24.3.1 checkPrescriptionHistory()	51
3.24.3.2 createPatientHistory()	51
3.24.3.3 detectDiagnosis()	51
3.24.3.4 detectPerson()	52
3.24.3.5 findNewPrescriptions()	52
3.24.3.6 findPatientHistory()	53
3.24.3.7 makePrescriptions()	53
3.24.3.8 prepareActivePrescriptions()	53
3.24.3.9 refreshPatients()	53
3.25 hospital.utils.queries.PatientDataQueries Interface Reference	54
3.25.1 Detailed Description	54
3.25.2 Member Data Documentation	54
3.25.2.1 ADD_DIAGNOSIS	54

3.25.2.2 ADD_DOCTOR	54
3.25.2.3 ADD_NURSE	55
3.25.2.4 FIND_ALL	55
3.25.2.5 FIND_DOCTOR_PATIENTS	55
3.25.2.6 FIND_NURSE_PATIENTS	55
3.25.2.7 FIND_PATIENT	55
3.25.2.8 FIND_WITHOUT_DOC	55
3.25.2.9 FIND_WITHOUT_NURSE	56
3.25.2.10 INSERT_PATIENT	56
3.25.2.11 UPDATE_PATIENT_STATUS	56
3.26 hospital.servlets.patientServlets.PatientPageServlet Class Reference	56
3.26.1 Detailed Description	57
3.26.2 Constructor & Destructor Documentation	57
3.26.2.1 PatientPageServlet() [1/2]	57
3.26.2.2 PatientPageServlet() [2/2]	57
3.26.3 Member Function Documentation	58
3.26.3.1 doPost()	58
3.27 hospital.entity.prescription.Prescription Class Reference	58
3.27.1 Detailed Description	59
3.27.2 Constructor & Destructor Documentation	59
3.27.2.1 Prescription() [1/2]	59
3.27.2.2 Prescription() [2/2]	60
3.27.3 Member Function Documentation	60
3.27.3.1 getAmountOfMedicine()	60
3.27.3.2 getId()	60
3.27.3.3 getName()	61
3.27.3.4 getPrescriptionClass()	61
3.27.3.5 setAmountOfMedicine()	61
3.27.3.6 setId()	61
3.27.3.7 setName()	62
3.27.3.8 setPrescriptionClass()	62
3.27.4 Member Data Documentation	62
3.27.4.1 CLASS_COLUMN	62
3.27.4.2 CONNECTED_TABLE_HISTORY	62
3.27.4.3 CONNECTED_TABLE_HISTORY_ID	63
3.27.4.4 CONNECTED_TABLE_HISTORY_PATIENT	63
3.27.4.5 CONNECTED_TABLE_HISTORY_PRESCR	63
3.27.4.6 CONNECTED_TABLE_NEW	63
3.27.4.7 CONNECTED_TABLE_NEW_ID	63
3.27.4.8 CONNECTED_TABLE_NEW_PATIENT	63
3.27.4.9 CONNECTED_TABLE_NEW_PRESCR	63
3.27.4.10 ID_COLUMN	64

3.27.4.11 MED_TABLE	64
3.27.4.12 MED_TABLE_AMOUNT	64
3.27.4.13 MED_TABLE_NAME	64
3.27.4.14 NAME_COLUMN	64
3.27.4.15 TABLE_NAME	64
3.28 hospital.dao.impl.PrescriptionDAOImpl Class Reference	65
3.28.1 Detailed Description	65
3.28.2 Member Function Documentation	65
3.28.2.1 delete()	65
3.28.2.2 findAll()	65
3.28.2.3 findByActivePrescriptions()	66
3.28.2.4 findByClass()	66
3.28.2.5 findById()	66
3.28.2.6 findByPatientHistory()	66
3.28.2.7 insertHistory()	67
3.28.2.8 insertNewPrescriptions()	67
3.29 hospital.utils.queries.PrescriptionQueries Interface Reference	67
3.29.1 Detailed Description	67
3.29.2 Member Data Documentation	68
3.29.2.1 DELETE_PRESCRIPTION	68
3.29.2.2 FIND_ALL	68
3.29.2.3 FIND_ALL_HISTORY	68
3.29.2.4 FIND_ALL_MED	68
3.29.2.5 FIND_ALL_NEW	68
3.29.2.6 FIND_BY_CLASS	69
3.29.2.7 FIND_BY_ID	69
3.29.2.8 FIND_MED_LIST	69
3.29.2.9 INSERT_NEW_PRESCRIPTION	69
3.29.2.10 INSERT_PRESCRIPTION_HISTORY	69
3.29.2.11 UPDATE_MED_LIST	69
3.30 hospital.utils.RandomNumber Class Reference	70
3.30.1 Detailed Description	70
3.30.2 Member Function Documentation	70
3.30.2.1 randNumber()	70
3.31 hospital.utils.RandomPatient Class Reference	70
3.31.1 Detailed Description	71
3.31.2 Constructor & Destructor Documentation	71
3.31.2.1 RandomPatient() [1/2]	71
3.31.2.2 RandomPatient() [2/2]	71
3.31.3 Member Function Documentation	71
3.31.3.1 randomPatientDoc()	71
3.31.3.2 randomPatientNurse()	72

3.32 hospital.utils.RegexContainer Interface Reference	72
3.32.1 Detailed Description	72
3.32.2 Member Data Documentation	72
3.32.2.1 REGEX_EMAIL	73
3.32.2.2 REGEX_NAME_ENG	73
3.32.2.3 REGEX_PASSWORD	73
3.33 hospital.utils.doctorcommands.ReleasePatientCommand Class Reference	73
3.33.1 Detailed Description	73
3.33.2 Constructor & Destructor Documentation	74
3.33.2.1 ReleasePatientCommand() [1/2]	74
3.33.2.2 ReleasePatientCommand() [2/2]	74
3.33.3 Member Function Documentation	74
3.33.3.1 execute()	74
3.34 hospital.entity.Role Class Reference	75
3.34.1 Detailed Description	75
3.34.2 Constructor & Destructor Documentation	75
3.34.2.1 Role()	75
3.34.3 Member Function Documentation	76
3.34.3.1 getId()	76
3.34.3.2 getName()	76
3.34.3.3 setId()	76
3.34.3.4 setName()	77
3.34.4 Member Data Documentation	77
3.34.4.1 ID_COLUMN	77
3.34.4.2 NAME_COLUMN	77
3.34.4.3 TABLE_NAME	77
3.35 hospital.dao.impl.RoleDAOImpl Class Reference	77
3.35.1 Detailed Description	78
3.35.2 Member Function Documentation	78
3.35.2.1 findById()	78
3.35.2.2 findByName()	78
3.36 hospital.utils.queries.RoleQueries Interface Reference	79
3.36.1 Detailed Description	79
3.36.2 Member Data Documentation	79
3.36.2.1 FIND_BY_ID	79
3.36.2.2 FIND_BY_NAME	79
3.37 hospital.utils.factories.ServletCommand Interface Reference	80
3.37.1 Detailed Description	80
3.37.2 Member Function Documentation	80
3.37.2.1 execute()	80
3.38 hospital.utils.SingleTransaction Class Reference	81
3.38.1 Detailed Description	81

3.38.2 Member Function Documentation	81
3.38.2.1 getInstance()	82
3.38.2.2 transfer()	82
3.39 hospital.utils.URLBase Enum Reference	82
3.39.1 Detailed Description	83
3.39.2 Constructor & Destructor Documentation	83
3.39.2.1 URLBase()	83
3.39.3 Member Function Documentation	83
3.39.3.1 getUrl()	83
3.39.3.2 getValue()	84
3.40 hospital.entity.User Class Reference	84
3.40.1 Detailed Description	85
3.40.2 Constructor & Destructor Documentation	85
3.40.2.1 User() [1/4]	85
3.40.2.2 User() [2/4]	86
3.40.2.3 User() [3/4]	86
3.40.2.4 User() [4/4]	87
3.40.3 Member Function Documentation	87
3.40.3.1 getEmail()	87
3.40.3.2 getId()	88
3.40.3.3 getName()	88
3.40.3.4 getPassword()	88
3.40.3.5 getRole()	88
3.40.3.6 getSurname()	89
3.40.3.7 setEmail()	89
3.40.3.8 setId()	90
3.40.3.9 setName()	90
3.40.3.10 setPassword()	90
3.40.3.11 setRole()	91
3.40.3.12 setSurname()	91
3.40.4 Member Data Documentation	91
3.40.4.1 EMAIL_COLUMN	91
3.40.4.2 ID_COLUMN	91
3.40.4.3 NAME_COLUMN	91
3.40.4.4 PASSWORD_COLUMN	92
3.40.4.5 ROLE_COLUMN	92
3.40.4.6 SURNAME_COLUMN	92
3.40.4.7 TABLE_NAME	92
3.41 hospital.utils.UserAlreadyExistsException Class Reference	92
3.41.1 Detailed Description	93
3.42 hospital.dao.impl.UserDAOImpl Class Reference	93
3.42.1 Detailed Description	93

3.42.2 Member Function Documentation	93
3.42.2.1 findAll()	94
3.42.2.2 findByEmailAndPass()	94
3.42.2.3 findById()	94
3.42.2.4 insert()	94
3.43 hospital.utils.utilsForDBData.UserDataManipulations Class Reference	95
3.43.1 Detailed Description	95
3.43.2 Constructor & Destructor Documentation	95
3.43.2.1 UserDataManipulations() [1/2]	95
3.43.2.2 UserDataManipulations() [2/2]	95
3.43.3 Member Function Documentation	95
3.43.3.1 detectRole()	95
3.44 hospital.utils.queries.UserQueries Interface Reference	96
3.44.1 Detailed Description	96
3.44.2 Member Data Documentation	96
3.44.2.1 FIND_ALL	96
3.44.2.2 FIND_BY_EMAIL_AND_PASS	97
3.44.2.3 FIND_BY_ID	97
3.44.2.4 INSERT_USER	97
Index	99

Chapter 1

Hierarchical Index

1.1 Class Hierarchy

This inheritance list is sorted roughly, but not completely, alphabetically:

hospital.utils.factories.CommandFactory	7
hospital.utils.ConnectionPool	8
hospital.utils.factories.DAOFactory	18
hospital.entity.diagnosis.Diagnosis	20
hospital.dao.impl.DiagnosisDAOImpl	23
hospital.utils.queries.DiagnosisQueries	24
Exception	
hospital.utils.UserAlreadyExistsException	92
Filter	
hospital.servlets.filter.AuthFilter	5
hospital.utils.InputDataCheck	30
hospital.utils.PasswordEncryptorSHA256	38
hospital.entity.PatientData	39
hospital.dao.impl.PatientDataImpl	47
hospital.utils.queries.PatientDataQueries	50
hospital.utils.queries.PatientDataManipulations	54
hospital.entity.prescription.Prescription	58
hospital.dao.impl.PrescriptionDAOImpl	65
hospital.utils.queries.PrescriptionQueries	67
hospital.utils.RandomNumber	70
hospital.utils.RandomPatient	70
hospital.utils.RegexContainer	72
hospital.entity.Role	75
hospital.dao.impl.RoleDAOImpl	77
hospital.utils.queries.RoleQueries	79
hospital.utils.factories.ServletCommand	80
hospital.utils.doctorcommands.CreateDiagnosisCommand	10
hospital.utils.doctorcommands.CreatePatientsCommand	11
hospital.utils.doctorcommands.CreatePrescriptionsCommand	13
hospital.utils.doctorcommands.DoctorPageCommand	25
hospital.utils.doctorcommands.DoPrescriptionCommand	28
hospital.utils.doctorcommands.ReleasePatientCommand	73
hospital.utils.nursecommands.NurseDoPrescrCommand	32
hospital.utils.nursecommands.NurseFindPatientsCommand	34
hospital.utils.nursecommands.NursePageCommand	35

hospital.utils.SingleTransaction	81
hospital.utils.URLBase	82
hospital.entity.User	84
hospital.dao.impl.UserDAOImpl	93
hospital.utils.utilsForDBData.UserDataManipulations	95
hospital.utils.queries.UserQueries	96
HttpServlet	
hospital.servlets.auth.LoginServlet	31
hospital.servlets.auth.LogoutServlet	31
hospital.servlets.CreateUserServlet	15
hospital.servlets.doctorServlets.DoctorPageServlet	27
hospital.servlets.nurseServlets.NursePageServlet	37
hospital.servlets.patientServlets.PatientPageServlet	56

Chapter 2

Class Index

2.1 Class List

Here are the classes, structs, unions and interfaces with brief descriptions:

hospital.servlets.filter.AuthFilter	5
hospital.utils.factories.CommandFactory	7
hospital.utils.ConnectionPool	8
hospital.utils.doctorcommands.CreateDiagnosisCommand	10
hospital.utils.doctorcommands.CreatePatientsCommand	11
hospital.utils.doctorcommands.CreatePrescriptionsCommand	13
hospital.servlets.CreateUserServlet	15
hospital.utils.factories.DAOFactory	18
hospital.entity.diagnosis.Diagnosis	20
hospital.dao.impl.DiagnosisDAOImpl	23
hospital.utils.queries.DiagnosisQueries	24
hospital.utils.doctorcommands.DoctorPageCommand	25
hospital.servlets.doctorServlets.DoctorPageServlet	27
hospital.utils.doctorcommands.DoPrescriptionCommand	28
hospital.utils.InputDataCheck	30
hospital.servlets.auth.LogoutServlet	31
hospital.utils.nursecommands.NurseDoPrescrCommand	32
hospital.utils.nursecommands.NurseFindPatientsCommand	34
hospital.utils.nursecommands.NursePageCommand	35
hospital.servlets.nurseServlets.NursePageServlet	37
hospital.utils.PasswordEncryptorSHA256	38
hospital.entity.PatientData	39
hospital.dao.impl.PatientDataImpl	47
hospital.utils.utilsForDBData.PatientDataManipulations	50
hospital.utils.queries.PatientDataQueries	54
hospital.servlets.patientServlets.PatientPageServlet	56
hospital.entity.prescription.Prescription	58
hospital.dao.impl.PrescriptionDAOImpl	65
hospital.utils.queries.PrescriptionQueries	67
hospital.utils.RandomNumber	70
hospital.utils.RandomPatient	70
hospital.utils.RegexContainer	72
hospital.utils.doctorcommands.ReleasePatientCommand	73
hospital.entity.Role	75
hospital.dao.impl.RoleDAOImpl	77

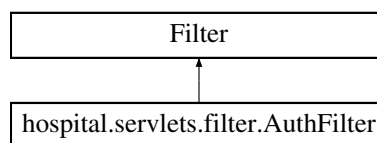
hospital.utils.queries.RoleQueries	79
hospital.utils.factories.ServletCommand	80
hospital.utils.SingleTransaction	81
hospital.utils.URLBase	82
hospital.entity.User	84
hospital.utils.UserAlreadyExistsException	92
hospital.dao.impl.UserDAOImpl	93
hospital.utils.utilsForDBData.UserDataManipulations	95
hospital.utils.queries.UserQueries	96

Chapter 3

Class Documentation

3.1 hospital.servlets.filter.AuthFilter Class Reference

Inheritance diagram for hospital.servlets.filter.AuthFilter:



Public Member Functions

- [AuthFilter](#) ()
- [AuthFilter](#) (DAOFactory daoFactory)
- void [init](#) (FilterConfig filterConfig) throws ServletException
- void [doFilter](#) (ServletRequest request, ServletResponse response, FilterChain chain) throws IOException, ServletException
- void [destroy](#) ()

3.1.1 Detailed Description

Class, which represents authentication filter.

Author

Yelyzaveta Onyshchenko

Version

1.01

3.1.2 Constructor & Destructor Documentation

3.1.2.1 AuthFilter() [1/2]

```
hospital.servlets.filter.AuthFilter.AuthFilter ( )
```

Constructor for creating new object

See also

[AuthFilter\(DAOFactory\)](#)

3.1.2.2 AuthFilter() [2/2]

```
hospital.servlets.filter.AuthFilter.AuthFilter (
    DAOFactory daoFactory )
```

Constructor for creating new object

See also

[AuthFilter\(\)](#)

3.1.3 Member Function Documentation

3.1.3.1 destroy()

```
void hospital.servlets.filter.AuthFilter.destroy ( )
```

On case of calling this method, doFilter method will not perform again on current instance

3.1.3.2 doFilter()

```
void hospital.servlets.filter.AuthFilter.doFilter (
    ServletRequest request,
    ServletResponse response,
    FilterChain chain ) throws IOException, ServletException
```

Method which perfoms filtering

Parameters

<i>request</i>	HttpRequest
<i>response</i>	HttpResponse
<i>chain</i>	FilterChain instance

Exceptions

<i>IOException</i>	on error of performing filtering
<i>ServletException</i>	on servlet error

3.1.3.3 init()

```
void hospital.servlets.filter.AuthFilter.init (
    FilterConfig filterConfig ) throws ServletException
```

Init filter method

Parameters

<i>filterConfig</i>	filter config
---------------------	---------------

Exceptions

<i>ServletException</i>	on servlet error
-------------------------	------------------

The documentation for this class was generated from the following file:

- C:/Users/Liza/Desktop/Hospital/src/main/java/hospital/servlets/filter/AuthFilter.java

3.2 hospital.utils.factories.CommandFactory Class Reference

Static Public Member Functions

- static [ServletCommand](#) [createCommand](#) (String url)

3.2.1 Detailed Description

Class, which represents factory of commands.

Author

Yelyzaveta Onyshchenko

Version

1.01

3.2.2 Member Function Documentation

3.2.2.1 createCommand()

```
static ServletCommand hospital.utils.factories.CommandFactory.createCommand (
    String url ) [static]
```

Method which creates command, based on incoming url

Parameters

<i>url</i>	incoming url
------------	--------------

Returns

created command to perform

The documentation for this class was generated from the following file:

- C:/Users/Liza/Desktop/Hospital/src/main/java/hospital/utils/factories/CommandFactory.java

3.3 hospital.utils.ConnectionPool Class Reference

Static Public Member Functions

- static Connection [getConnection](#) () throws SQLException

Static Public Attributes

- static final String [USER](#) = "root"
- static final String [PASSWORD](#) = "root"
- static final String [URL](#) = "jdbc:mysql://localhost:3306/hospitaldb?useSSL=false&serverTimezone=UTC"

3.3.1 Detailed Description

Class, which represents connection pool

Author

Yelyzaveta Onyshchenko

Version

1.01

3.3.2 Member Function Documentation

3.3.2.1 getConnection()

```
static Connection hospital.utils.ConnectionPool.getConnection ( ) throws SQLException [static]
```

Method for getting connection

Returns

connection to database

Exceptions

<i>SQLException</i>	SQLException
---------------------	--------------

3.3.3 Member Data Documentation

3.3.3.1 PASSWORD

```
final String hospital.utils.ConnectionPool.PASSWORD = "root" [static]
```

MySQL user's password

3.3.3.2 URL

```
final String hospital.utils.ConnectionPool.URL = "jdbc:mysql://localhost:3306/hospitaldb?use↵  
SSL=false&serverTimezone=UTC" [static]
```

Url for accessing the database

3.3.3.3 USER

```
final String hospital.utils.ConnectionPool.USER = "root" [static]
```

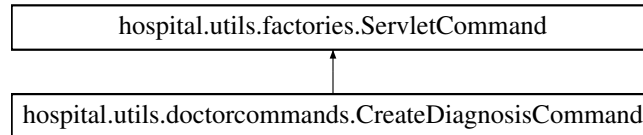
MySQL user's name

The documentation for this class was generated from the following file:

- C:/Users/Liza/Desktop/Hospital/src/main/java/hospital/utils/ConnectionPool.java

3.4 hospital.utils.doctorcommands.CreateDiagnosisCommand Class Reference

Inheritance diagram for hospital.utils.doctorcommands.CreateDiagnosisCommand:



Public Member Functions

- [CreateDiagnosisCommand](#) ()
- [CreateDiagnosisCommand](#) ([DAOFactory](#) factory)
- void [execute](#) (HttpServletRequest request, HttpServletResponse response) throws ServletException, IOException

3.4.1 Detailed Description

Class, which represents creating of the diagnosis command.

Author

Yelyzaveta Onyshchenko

Version

1.01

3.4.2 Constructor & Destructor Documentation

3.4.2.1 CreateDiagnosisCommand() [1/2]

```
hospital.utils.doctorcommands.CreateDiagnosisCommand.CreateDiagnosisCommand ( )
```

Constructor for creating new object

See also

[CreateDiagnosisCommand\(DAOFactory\)](#)

3.4.2.2 CreateDiagnosisCommand() [2/2]

```
hospital.utils.doctorcommands.CreateDiagnosisCommand.CreateDiagnosisCommand (
    DAOFactory factory )
```

Constructor for creating new object,used for testing

See also

[CreateDiagnosisCommand\(\)](#)

3.4.3 Member Function Documentation

3.4.3.1 execute()

```
void hospital.utils.doctorcommands.CreateDiagnosisCommand.execute (
    HttpServletRequest request,
    HttpServletResponse response ) throws ServletException, IOException
```

Execute command method

Parameters

<i>request</i>	HttpRequest
<i>response</i>	HttpResponse

Exceptions

<i>ServletException</i>	on servlet exception
<i>IOException</i>	on error occured while processing the request

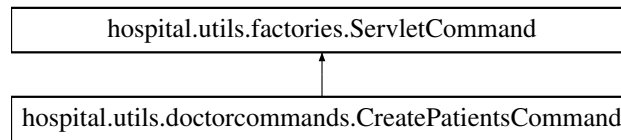
Implements [hospital.utils.factories.ServletCommand](#).

The documentation for this class was generated from the following file:

- C:/Users/Liza/Desktop/Hospital/src/main/java/hospital/utils/doctorcommands/CreateDiagnosisCommand.↔
java

3.5 hospital.utils.doctorcommands.CreatePatientsCommand Class Reference

Inheritance diagram for hospital.utils.doctorcommands.CreatePatientsCommand:



Public Member Functions

- [CreatePatientsCommand](#) ()
- [CreatePatientsCommand](#) ([RandomPatient](#) randomPatient)
- void [execute](#) (HttpServletRequest request, HttpServletResponse response) throws ServletException, IOException

3.5.1 Detailed Description

Class, which represents creating of the patient command.

Author

Yelyzaveta Onyshchenko

Version

1.01

3.5.2 Constructor & Destructor Documentation

3.5.2.1 [CreatePatientsCommand\(\)](#) [1/2]

```
hospital.utils.doctorcommands.CreatePatientsCommand.CreatePatientsCommand ( )
```

Constructor for creating new object

See also

[CreatePatientsCommand\(RandomPatient \)](#)

3.5.2.2 [CreatePatientsCommand\(\)](#) [2/2]

```
hospital.utils.doctorcommands.CreatePatientsCommand.CreatePatientsCommand (
    RandomPatient randomPatient )
```

Constructor for creating new object

See also

[CreatePatientsCommand\(\)](#)

3.5.3 Member Function Documentation

3.5.3.1 execute()

```
void hospital.utils.doctorcommands.CreatePatientsCommand.execute (
    HttpServletRequest request,
    HttpServletResponse response ) throws ServletException, IOException
```

Execute command method

Parameters

<i>request</i>	HttpRequest
<i>response</i>	HttpResponse

Exceptions

<i>ServletException</i>	on servlet exception
<i>IOException</i>	on error occured while processing the request

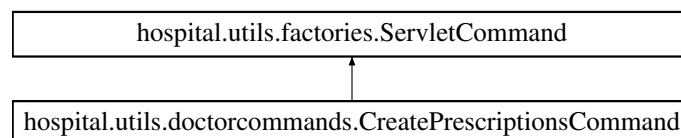
Implements [hospital.utils.factories.ServletCommand](#).

The documentation for this class was generated from the following file:

- C:/Users/Liza/Desktop/Hospital/src/main/java/hospital/utils/doctorcommands/CreatePatientsCommand.java

3.6 hospital.utils.doctorcommands.CreatePrescriptionsCommand Class Reference

Inheritance diagram for hospital.utils.doctorcommands.CreatePrescriptionsCommand:



Public Member Functions

- [CreatePrescriptionsCommand](#) ()
- [CreatePrescriptionsCommand](#) ([DAOFactory](#) daoFactory, [PatientDataManipulations](#) patientDataManipulations)
- void [execute](#) (HttpServletRequest request, HttpServletResponse response) throws ServletException, IOException

3.6.1 Detailed Description

Class, which represents creating of prescriptions command.

Author

Yelyzaveta Onyshchenko

Version

1.01

3.6.2 Constructor & Destructor Documentation

3.6.2.1 CreatePrescriptionsCommand() [1/2]

```
hospital.utils.doctorcommands.CreatePrescriptionsCommand.CreatePrescriptionsCommand ( )
```

Constructor for creating new object

See also

CreatePrescriptionsCommand(DAOFactory)

3.6.2.2 CreatePrescriptionsCommand() [2/2]

```
hospital.utils.doctorcommands.CreatePrescriptionsCommand.CreatePrescriptionsCommand (
    DAOFactory daoFactory,
    PatientDataManipulations patientDataManipulations )
```

Constructor for creating new object,used for testing

See also

[CreatePrescriptionsCommand\(\)](#)

3.6.3 Member Function Documentation

3.6.3.1 execute()

```
void hospital.utils.doctorcommands.CreatePrescriptionsCommand.execute (
    HttpServletRequest request,
    HttpServletResponse response ) throws ServletException, IOException
```

Execute command method

Parameters

<i>request</i>	HttpRequest
<i>response</i>	HttpResponse

Exceptions

<i>ServletException</i>	on servlet exception
<i>IOException</i>	on error occured while processing the request

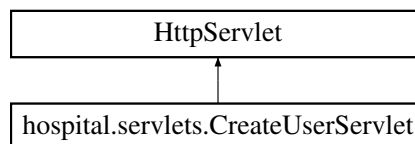
Implements [hospital.utils.factories.ServletCommand](#).

The documentation for this class was generated from the following file:

- C:/Users/Liza/Desktop/Hospital/src/main/java/hospital/utlis/doctorcommands/CreatePrescriptionsCommand.↔
java

3.7 hospital.servlets.CreateUserServlet Class Reference

Inheritance diagram for hospital.servlets.CreateUserServlet:



Public Member Functions

- [CreateUserServlet](#) ()
- [CreateUserServlet](#) (DAOFactory daoFactory, [PatientDataManipulations](#) patientDataManipulations2)

Static Public Member Functions

- static boolean [inputValidator](#) (String name, String surname, String email, String password)

Protected Member Functions

- void [doGet](#) (HttpServletRequest req, HttpServletResponse resp) throws ServletException, IOException
- void [doPost](#) (HttpServletRequest req, HttpServletResponse resp) throws ServletException, IOException

3.7.1 Detailed Description

Class, which represents servlet for creating new user.

Author

Yelyzaveta Onyshchenko

Version

1.01

3.7.2 Constructor & Destructor Documentation

3.7.2.1 `CreateUserServlet()` [1/2]

```
hospital.servlets.CreateUserServlet.CreateUserServlet ( )
```

Constructor for creating new object

See also

[CreateUserServlet\(DAOFactory,PatientDataManipulations\)](#)

3.7.2.2 `CreateUserServlet()` [2/2]

```
hospital.servlets.CreateUserServlet.CreateUserServlet (
    DAOFactory daoFactory,
    PatientDataManipulations patientDataManipulations2 )
```

Constructor for creating new object

See also

[CreateUserServlet\(\)](#)

3.7.3 Member Function Documentation

3.7.3.1 `doGet()`

```
void hospital.servlets.CreateUserServlet.doGet (
    HttpServletRequest req,
    HttpServletResponse resp ) throws ServletException, IOException [protected]
```

Method performing HHTP GET request

Parameters

<i>req</i>	HttpRequest
<i>resp</i>	HttpResponse

Exceptions

<i>ServletException</i>	On servlet error
<i>IOException</i>	On error fulfilling the request

3.7.3.2 doPost()

```
void hospital.servlets.CreateUserServlet.doPost (
    HttpServletRequest req,
    HttpServletResponse resp ) throws ServletException, IOException [protected]
```

Method performing HHTP POST request

Parameters

<i>req</i>	HttpRequest
<i>resp</i>	HttpResponse

Exceptions

<i>ServletException</i>	On servlet error
<i>IOException</i>	On error fulfilling the request

3.7.3.3 inputValidator()

```
static boolean hospital.servlets.CreateUserServlet.inputValidator (
    String name,
    String surname,
    String email,
    String password ) [static]
```

Method for validating input values using regular expressions

Parameters

<i>name</i>	name of the user to check
<i>surname</i>	surname of the user to check
<i>email</i>	email of the user to check
<i>password</i>	password of the user to check

Returns

true if data is valid,else - false

The documentation for this class was generated from the following file:

- C:/Users/Liza/Desktop/Hospital/src/main/java/hospital/servlets/CreateUserServlet.java

3.8 hospital.utils.factories.DAOFactory Class Reference

Public Member Functions

- [UserDAOImpl](#) `createUserDao ()`
- [PatientDataImpl](#) `createPatientDataDao ()`
- [PrescriptionDAOImpl](#) `createPrescriptionDao ()`
- [DiagnosisDAOImpl](#) `createDiagnosisDao ()`
- [RoleDAOImpl](#) `createRoleDao ()`
- [SingleTransaction](#) `createSingleTransaction ()`

3.8.1 Detailed Description

Class, which represents DAO factory.

Author

Yelyzaveta Onyshchenko

Version

1.01

3.8.2 Member Function Documentation

3.8.2.1 createDiagnosisDao()

[DiagnosisDAOImpl](#) `hospital.utils.factories.DAOFactory.createDiagnosisDao ()`

Returns new diagnosis DAO.

Returns

new instance of `diagnosisDAOImpl`

3.8.2.2 createPatientDataDao()

```
PatientDataImpl hospital.utils.factories.DAOFactory.createPatientDataDao ( )
```

Returns new patient data DAO.

Returns

new instance of patientDataImpl

3.8.2.3 createPrescriptionDao()

```
PrescriptionDAOImpl hospital.utils.factories.DAOFactory.createPrescriptionDao ( )
```

Returns new prescription DAO.

Returns

new instance of prescriptionDAOImpl

3.8.2.4 createRoleDao()

```
RoleDAOImpl hospital.utils.factories.DAOFactory.createRoleDao ( )
```

Returns new role DAO.

Returns

new instance of roleDAOImpl

3.8.2.5 createSingleTransaction()

```
SingleTransaction hospital.utils.factories.DAOFactory.createSingleTransaction ( )
```

Returns new single transaction for medical prescriptions

Returns

new instance of [SingleTransaction](#)

3.8.2.6 createUserDao()

```
UserDAOImpl hospital.utils.factories.DAOFactory.createUserDao ( )
```

Returns new user DAO.

Returns

new instance of userDAOImpl

The documentation for this class was generated from the following file:

- C:/Users/Liza/Desktop/Hospital/src/main/java/hospital/utils/factories/DAOFactory.java

3.9 hospital.entity.diagnosis.Diagnosis Class Reference

Public Member Functions

- [Diagnosis](#) (int id, String name, String difficulty)
- int [getId](#) ()
- void [setId](#) (int id)
- String [getName](#) ()
- void [setName](#) (String name)
- String [getDifficulty](#) ()
- void [setDifficulty](#) (String difficulty)

Static Public Attributes

- static final String [TABLE_NAME](#) = "diagnosis_table"
- static final String [ID_COLUMN](#) = "id_diagnosis"
- static final String [NAME_COLUMN](#) = "diagnosis_name"
- static final String [DIFFICULTY_COLUMN](#) = "diagnosis_difficulty"

3.9.1 Detailed Description

Class, which represents diagnosis which can be established to the patient.

Author

Yelyzaveta Onyshchenko

Version

1.01

3.9.2 Constructor & Destructor Documentation

3.9.2.1 Diagnosis()

```
hospital.entity.diagnosis.Diagnosis.Diagnosis (
    int id,
    String name,
    String difficulty )
```

Constructor for creating new object.

Parameters

<i>id</i>	corresponding id of the diagnosis
<i>name</i>	coresponding name of the diagnosis
<i>difficulty</i>	coresponding difficulty of the diagnosis

3.9.3 Member Function Documentation

3.9.3.1 getDifficulty()

```
String hospital.entity.diagnosis.Diagnosis.getDifficulty ( )
```

Method, which returns difficulty of the diagnosis

Returns

difficulty of the diagnosis

3.9.3.2 getId()

```
int hospital.entity.diagnosis.Diagnosis.getId ( )
```

Method, which returns id of the diagnosis

Returns

id of the diagnosis

3.9.3.3 getName()

```
String hospital.entity.diagnosis.Diagnosis.getName ( )
```

Method, which returns name of the diagnosis

Returns

name of the diagnosis

3.9.3.4 setDifficulty()

```
void hospital.entity.diagnosis.Diagnosis.setDifficulty (
    String difficulty )
```

Method, which sets difficulty of the diagnosis

Parameters

<i>difficulty</i>	corresponding difficulty of the diagnosis
-------------------	---

3.9.3.5 setId()

```
void hospital.entity.diagnosis.Diagnosis.setId (  
    int id )
```

Method, which sets id of the diagnosis

Parameters

<i>id</i>	corresponding id of the diagnosis
-----------	-----------------------------------

3.9.3.6 setName()

```
void hospital.entity.diagnosis.Diagnosis.setName (  
    String name )
```

Method, which sets name of the diagnosis

Parameters

<i>name</i>	corresponding name of the diagnosis
-------------	-------------------------------------

3.9.4 Member Data Documentation**3.9.4.1 DIFFICULTY_COLUMN**

```
final String hospital.entity.diagnosis.Diagnosis.DIFFICULTY_COLUMN = "diagnosis_difficulty"  
[static]
```

The corresponding name of the "diagnosis difficulty" column in table name.

3.9.4.2 ID_COLUMN

```
final String hospital.entity.diagnosis.Diagnosis.ID_COLUMN = "id_diagnosis" [static]
```

The corresponding name of the "id" column in table name.

3.9.4.3 NAME_COLUMN

```
final String hospital.entity.diagnosis.Diagnosis.NAME_COLUMN = "diagnosis_name" [static]
```

The corresponding name of the "name" column in table name.

3.9.4.4 TABLE_NAME

```
final String hospital.entity.diagnosis.Diagnosis.TABLE_NAME = "diagnosis_table" [static]
```

The corresponding name of table in database.

The documentation for this class was generated from the following file:

- C:/Users/Liza/Desktop/Hospital/src/main/java/hospital/entity/diagnosis/Diagnosis.java

3.10 hospital.dao.impl.DiagnosisDAOImpl Class Reference

Public Member Functions

- List< [Diagnosis](#) > [findAll](#) ()
- [Diagnosis](#) [findById](#) (int id)

3.10.1 Detailed Description

Class, which represents Diagnosis DAO

Author

Yelyzaveta Onyshchenko

Version

1.01

3.10.2 Member Function Documentation

3.10.2.1 findAll()

```
List<Diagnosis> hospital.dao.impl.DiagnosisDAOImpl.findAll ( )
```

Find all diagnoses

Returns

list of diagnoses

3.10.2.2 findById()

```
Diagnosis hospital.dao.impl.DiagnosisDAOImpl.findById (
    int id )
```

Find diagnosis by id

Returns

corresponding diagnosis

The documentation for this class was generated from the following file:

- C:/Users/Liza/Desktop/Hospital/src/main/java/hospital/dao/impl/DiagnosisDAOImpl.java

3.11 hospital.utils.queries.DiagnosisQueries Interface Reference

Static Public Attributes

- static final String [FIND_ALL](#) = "select * from " + Diagnosis.TABLE_NAME
- static final String [FIND_BY_ID](#) = [FIND_ALL](#) + " where " + Diagnosis.ID_COLUMN + " = ?"

3.11.1 Detailed Description

Basic queries for diagnosis table.

Author

Yelyzaveta Onyshchenko

Version

1.01

3.11.2 Member Data Documentation

3.11.2.1 FIND_ALL

```
final String hospital.utils.queries.DiagnosisQueries.FIND_ALL = "select * from " + Diagnosis.↵
TABLE_NAME [static]
```

```
SELECT * FROM diagnosis_table;
```

3.11.2.2 FIND_BY_ID

```
final String hospital.utils.queries.DiagnosisQueries.FIND_BY_ID = FIND_ALL + " where " + Diagnosis.ID_COLUMN + " = ?" [static]
```

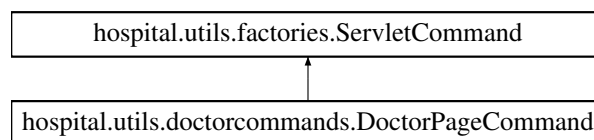
```
SELECT * FROM diagnosis_table WHERE id_diagnosis = ?;
```

The documentation for this interface was generated from the following file:

- C:/Users/Liza/Desktop/Hospital/src/main/java/hospital/utils/queries/DiagnosisQueries.java

3.12 hospital.utils.doctorcommands.DoctorPageCommand Class Reference

Inheritance diagram for hospital.utils.doctorcommands.DoctorPageCommand:



Public Member Functions

- [DoctorPageCommand](#) ()
- [DoctorPageCommand](#) ([DAOFactory](#) factory)
- void [execute](#) (HttpServletRequest request, HttpServletResponse response) throws ServletException, IOException

3.12.1 Detailed Description

Class, which represents doctor page command.

Author

Yelyzaveta Onyshchenko

Version

1.01

3.12.2 Constructor & Destructor Documentation

3.12.2.1 DoctorPageCommand() [1/2]

```
hospital.utils.doctorcommands.DoctorPageCommand.DoctorPageCommand ( )
```

Constructor for creating new object

See also

[DoctorPageCommand\(DAOFactory\)](#)

3.12.2.2 DoctorPageCommand() [2/2]

```
hospital.utils.doctorcommands.DoctorPageCommand.DoctorPageCommand (
    DAOFactory factory )
```

Constructor for creating new object

See also

[DoctorPageCommand\(\)](#)

3.12.3 Member Function Documentation

3.12.3.1 execute()

```
void hospital.utils.doctorcommands.DoctorPageCommand.execute (
    HttpServletRequest request,
    HttpServletResponse response ) throws ServletException, IOException
```

Execute command method

Parameters

<i>request</i>	HttpRequest
<i>response</i>	HttpResponse

Exceptions

<i>ServletException</i>	on servlet exception
<i>IOException</i>	on error occured while processing the request

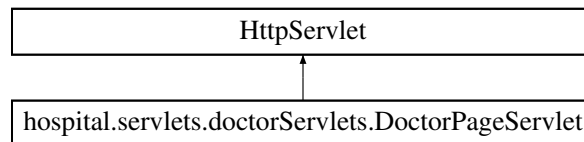
Implements [hospital.utils.factories.ServletCommand](#).

The documentation for this class was generated from the following file:

- C:/Users/Liza/Desktop/Hospital/src/main/java/hospital/utills/doctorcommands/DoctorPageCommand.java

3.13 hospital.servlets.doctorServlets.DoctorPageServlet Class Reference

Inheritance diagram for hospital.servlets.doctorServlets.DoctorPageServlet:



Protected Member Functions

- void [doPost](#) (HttpServletRequest request, HttpServletResponse response) throws ServletException, IOException

3.13.1 Detailed Description

Class, which represents doctor page servlet.

Author

Yelyzaveta Onyshchenko

Version

1.01

3.13.2 Member Function Documentation

3.13.2.1 doPost()

```
void hospital.servlets.doctorServlets.DoctorPageServlet.doPost (  
    HttpServletRequest request,  
    HttpServletResponse response ) throws ServletException, IOException [protected]
```

Method performing HHTP POST request

Parameters

<i>request</i>	HttpRequest
<i>response</i>	HttpResponse

Exceptions

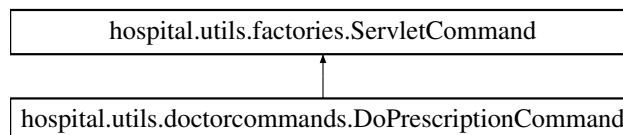
<i>ServletException</i>	On servlet error
<i>IOException</i>	On error fulfilling the request

The documentation for this class was generated from the following file:

- C:/Users/Liza/Desktop/Hospital/src/main/java/hospital/servlets/doctorServlets/DoctorPageServlet.java

3.14 hospital.utils.doctorcommands.DoPrescriptionCommand Class Reference

Inheritance diagram for hospital.utils.doctorcommands.DoPrescriptionCommand:

**Public Member Functions**

- [DoPrescriptionCommand](#) ()
- [DoPrescriptionCommand](#) ([DAOFactory](#) factory)
- void [execute](#) (HttpServletRequest request, HttpServletResponse response) throws ServletException, IOException

3.14.1 Detailed Description

Class, which represents performing of the diagnosis command.

Author

Yelyzaveta Onyshchenko

Version

1.01

3.14.2 Constructor & Destructor Documentation

3.14.2.1 DoPrescriptionCommand() [1/2]

```
hospital.utils.doctorcommands.DoPrescriptionCommand.DoPrescriptionCommand ( )
```

Constructor for creating new object

See also

[DoPrescriptionCommand\(DAOFactory\)](#)

3.14.2.2 DoPrescriptionCommand() [2/2]

```
hospital.utils.doctorcommands.DoPrescriptionCommand.DoPrescriptionCommand (
    DAOFactory factory )
```

Constructor for creating new object

See also

[DoPrescriptionCommand\(\)](#)

3.14.3 Member Function Documentation

3.14.3.1 execute()

```
void hospital.utils.doctorcommands.DoPrescriptionCommand.execute (
    HttpServletRequest request,
    HttpServletResponse response ) throws ServletException, IOException
```

Execute command method

Parameters

<i>request</i>	HttpRequest
<i>response</i>	HttpResponse

Exceptions

<i>ServletException</i>	on servlet exception
<i>IOException</i>	on error occured while processing the request

Implements [hospital.utils.factories.ServletCommand](#).

The documentation for this class was generated from the following file:

- C:/Users/Liza/Desktop/Hospital/src/main/java/hospital/utills/doctorcommands/DoPrescriptionCommand.java

3.15 hospital.utills.InputDataCheck Class Reference

Static Public Member Functions

- static boolean [inputCheck](#) (String input, String regex)

3.15.1 Detailed Description

Class for checking input data

Author

Yelyzaveta Onyshchenko

Version

1.01

3.15.2 Member Function Documentation

3.15.2.1 inputCheck()

```
static boolean hospital.utills.InputDataCheck.inputCheck (  
    String input,  
    String regex ) [static]
```

Method which compares input string with regular expressin

Parameters

<i>input</i>	input string
<i>regex</i>	regular expression

Returns

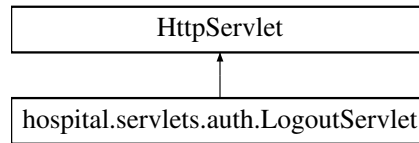
result of string comparison

The documentation for this class was generated from the following file:

- C:/Users/Liza/Desktop/Hospital/src/main/java/hospital/utills/InputDataCheck.java

3.16 hospital.servlets.auth.LogoutServlet Class Reference

Inheritance diagram for hospital.servlets.auth.LogoutServlet:



Protected Member Functions

- void [doGet](#) (HttpServletRequest req, HttpServletResponse resp) throws ServletException, IOException

3.16.1 Detailed Description

Class, which represents log out servlet.

Author

Yelyzaveta Onyshchenko

Version

1.01

3.16.2 Member Function Documentation

3.16.2.1 doGet()

```
void hospital.servlets.auth.LogoutServlet.doGet (  
    HttpServletRequest req,  
    HttpServletResponse resp ) throws ServletException, IOException [protected]
```

Method performing HTTP GET request

Parameters

<i>req</i>	HttpRequest
<i>resp</i>	HttpResponse

Exceptions

<i>ServletException</i>	On servlet error
-------------------------	------------------

Exceptions

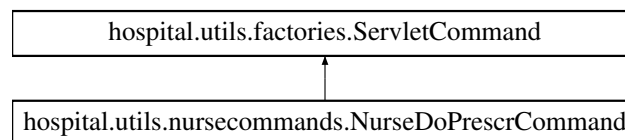
<i>IOException</i>	On error fulfilling the request
--------------------	---------------------------------

The documentation for this class was generated from the following file:

- C:/Users/Liza/Desktop/Hospital/src/main/java/hospital/servlets/auth/LogoutServlet.java

3.17 hospital.utils.nursecommands.NurseDoPrescrCommand Class Reference

Inheritance diagram for hospital.utils.nursecommands.NurseDoPrescrCommand:



Public Member Functions

- [NurseDoPrescrCommand](#) ()
- [NurseDoPrescrCommand](#) ([DAOFactory](#) factory)
- void [execute](#) (HttpServletRequest request, HttpServletResponse response) throws ServletException, [IOException](#)

3.17.1 Detailed Description

Class, which represents performing of the diagnosis command.

Author

Yelyzaveta Onyshchenko

Version

1.01

3.17.2 Constructor & Destructor Documentation

3.17.2.1 NurseDoPrescrCommand() [1/2]

```
hospital.utils.nursecommands.NurseDoPrescrCommand.NurseDoPrescrCommand ( )
```

Constructor for creating new object

See also

[NurseDoPrescrCommand\(DAOFactory\)](#)

3.17.2.2 NurseDoPrescrCommand() [2/2]

```
hospital.utils.nursecommands.NurseDoPrescrCommand.NurseDoPrescrCommand (
    DAOFactory factory )
```

Constructor for creating new object

See also

[NurseDoPrescrCommand\(\)](#)

3.17.3 Member Function Documentation

3.17.3.1 execute()

```
void hospital.utils.nursecommands.NurseDoPrescrCommand.execute (
    HttpServletRequest request,
    HttpServletResponse response ) throws ServletException, IOException
```

Execute command method

Parameters

<i>request</i>	HttpRequest
<i>response</i>	HttpResponse

Exceptions

<i>ServletException</i>	on servlet exception
<i>IOException</i>	on error occured while processing the request

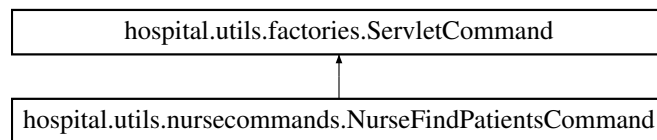
Implements [hospital.utils.factories.ServletCommand](#).

The documentation for this class was generated from the following file:

- C:/Users/Liza/Desktop/Hospital/src/main/java/hospital/utlis/nursecommands/NurseDoPrescrCommand.java

3.18 hospital.utlis.nursecommands.NurseFindPatientsCommand Class Reference

Inheritance diagram for hospital.utlis.nursecommands.NurseFindPatientsCommand:



Public Member Functions

- [NurseFindPatientsCommand](#) ()
- [NurseFindPatientsCommand](#) ([RandomPatient](#) randomPatient)
- void [execute](#) (HttpServletRequest request, HttpServletResponse response) throws ServletException, IOException

3.18.1 Detailed Description

Class, which represents creating of patients command.

Author

Yelyzaveta Onyshchenko

Version

1.01

3.18.2 Constructor & Destructor Documentation

3.18.2.1 NurseFindPatientsCommand() [1/2]

```
hospital.utlis.nursecommands.NurseFindPatientsCommand.NurseFindPatientsCommand ( )
```

Constructor for creating new object,used for testing

See also

[NurseFindPatientsCommand\(RandomPatient\)](#)

3.18.2.2 NurseFindPatientsCommand() [2/2]

```
hospital.utils.nursecommands.NurseFindPatientsCommand.NurseFindPatientsCommand (
    RandomPatient randomPatient )
```

Constructor for creating new object,used for testing

See also

[NurseFindPatientsCommand\(\)](#)

3.18.3 Member Function Documentation**3.18.3.1 execute()**

```
void hospital.utils.nursecommands.NurseFindPatientsCommand.execute (
    HttpServletRequest request,
    HttpServletResponse response ) throws ServletException, IOException
```

Execute command method

Parameters

<i>request</i>	HttpRequest
<i>response</i>	HttpResponse

Exceptions

<i>ServletException</i>	on servlet exception
<i>IOException</i>	on error occurred while processing the request

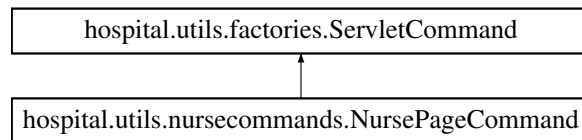
Implements [hospital.utils.factories.ServletCommand](#).

The documentation for this class was generated from the following file:

- C:/Users/Liza/Desktop/Hospital/src/main/java/hospital/utils/nursecommands/NurseFindPatientsCommand.[↔](#)java

3.19 hospital.utils.nursecommands.NursePageCommand Class Reference

Inheritance diagram for hospital.utils.nursecommands.NursePageCommand:



Public Member Functions

- [NursePageCommand\(\)](#)
- [NursePageCommand\(DAOFactory factory\)](#)
- void [execute](#) (HttpServletRequest request, HttpServletResponse response) throws ServletException, IOException

3.19.1 Detailed Description

Class, which represents nurse page command.

Author

Yelyzaveta Onyshchenko

Version

1.01

3.19.2 Constructor & Destructor Documentation

3.19.2.1 NursePageCommand() [1/2]

```
hospital.utils.nursecommands.NursePageCommand.NursePageCommand ( )
```

Constructor for creating new object

See also

[NursePageCommand\(DAOFactory\)](#)

3.19.2.2 NursePageCommand() [2/2]

```
hospital.utils.nursecommands.NursePageCommand.NursePageCommand (
    DAOFactory factory )
```

Constructor for creating new object

See also

[NursePageCommand\(\)](#)

3.19.3 Member Function Documentation

3.19.3.1 execute()

```
void hospital.utils.nursecommands.NursePageCommand.execute (
    HttpServletRequest request,
    HttpServletResponse response ) throws ServletException, IOException
```

Execute command method

Parameters

<i>request</i>	HttpRequest
<i>response</i>	HttpResponse

Exceptions

<i>ServletException</i>	on servlet exception
<i>IOException</i>	on error occured while processing the request

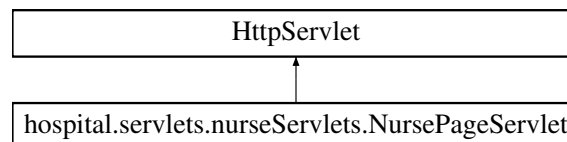
Implements [hospital.utils.factories.ServletCommand](#).

The documentation for this class was generated from the following file:

- C:/Users/Liza/Desktop/Hospital/src/main/java/hospital/utils/nursecommands/NursePageCommand.java

3.20 hospital.servlets.nurseServlets.NursePageServlet Class Reference

Inheritance diagram for hospital.servlets.nurseServlets.NursePageServlet:



Protected Member Functions

- void [doPost](#) (HttpServletRequest request, HttpServletResponse response) throws ServletException, IOException

3.20.1 Detailed Description

Class, which represents nurse page servlet.

Author

Yelyzaveta Onyshchenko

Version

1.01

3.20.2 Member Function Documentation

3.20.2.1 doPost()

```
void hospital.servlets.nurseServlets.NursePageServlet.doPost (  
    HttpServletRequest request,  
    HttpServletResponse response ) throws ServletException, IOException [protected]
```

Method performing HHTP POST request

Parameters

<i>request</i>	HttpRequest
<i>response</i>	HttpServletResponse

Exceptions

<i>ServletException</i>	On servlet error
<i>IOException</i>	On error fulfilling the request

The documentation for this class was generated from the following file:

- C:/Users/Liza/Desktop/Hospital/src/main/java/hospital/servlets/nurseServlets/NursePageServlet.java

3.21 hospital.utils.PasswordEncryptorSHA256 Class Reference

Static Public Member Functions

- static String [encryptPasswordWithSHA256](#) (String input) throws NoSuchAlgorithmException

3.21.1 Detailed Description

Class, which represents password encryptor using algorithm SHA-256

Author

Yelyzaveta Onyshchenko

Version

1.01

3.21.2 Member Function Documentation

3.21.2.1 encryptPasswordWithSHA256()

```
static String hospital.utils.PasswordEncryptorSHA256.encryptPasswordWithSHA256 (  
    String input ) throws NoSuchAlgorithmException [static]
```

Method that encrypts string using SHA-256

Parameters

<i>input</i>	input string
--------------	--------------

Returns

encrypted string

Exceptions

<i>NoSuchAlgorithmException</i>	no such algorithm exception
---------------------------------	-----------------------------

The documentation for this class was generated from the following file:

- C:/Users/Liza/Desktop/Hospital/src/main/java/hospital/utils/PasswordEncryptorSHA256.java

3.22 hospital.entity.PatientData Class Reference

Public Member Functions

- [PatientData](#) (int patientId, int doctorId, int nursed, int diagnosisId, int patientStatus)

- [PatientData](#) (int patientId, int doctorId, int nurseId, int diagnosisId, int patientStatus, [PatientDataManipulations](#) manipulations)
- Set< [Prescription](#) > [getPrescriptionHistory](#) ()
- void [setPrescriptionHistory](#) (Set< [Prescription](#) > prescriptionHistory)
- Set< [Prescription](#) > [getCurrentPrescriptions](#) ()
- void [setCurrentPrescriptions](#) (Set< [Prescription](#) > currentPrescriptions)
- int [getId](#) ()
- void [setId](#) (int id)
- [User](#) [getPatient](#) ()
- void [setPatient](#) ([User](#) patient)
- [User](#) [getDoctor](#) ()
- void [setDoctor](#) ([User](#) doctor)
- [User](#) [getNurse](#) ()
- void [setNurse](#) ([User](#) nurse)
- [Diagnosis](#) [getDiagnosis](#) ()
- void [setDiagnosis](#) ([Diagnosis](#) diagnosis)
- int [getPatientStatus](#) ()
- void [setPatientStatus](#) (int patientStatus)

Static Public Attributes

- static final String [TABLE_NAME](#) = "patient_info"
- static final String [ID_COLUMN](#) = "id"
- static final String [PATIENT_COLUMN](#) = "patient_id"
- static final String [DOCTOR_COLUMN](#) = "doctor_id"
- static final String [NURSE_COLUMN](#) = "nurse_id"
- static final String [DIAGNOSIS_COLUMN](#) = "diagnosis_id"
- static final String [STATUS_COLUMN](#) = "patient_status"

3.22.1 Detailed Description

Class, which represents patient data.

Author

Yelyzaveta Onyshchenko

Version

1.01

3.22.2 Constructor & Destructor Documentation

3.22.2.1 PatientData() [1/2]

```
hospital.entity.PatientData.PatientData (
    int patientId,
    int doctorId,
    int nurseId,
    int diagnosisId,
    int patientStatus )
```

Constructor for creating new object.

Parameters

<i>patientId</i>	corresponding patient id
<i>doctorId</i>	corresponding doctor id
<i>nurseId</i>	corresponding nurse id
<i>diagnosisId</i>	corresponding diagnosis id
<i>patientStatus</i>	corresponding patient status

See also

[PatientData\(int,int,int,int,int,PatientDataManipulations\)](#)

3.22.2.2 PatientData() [2/2]

```
hospital.entity.PatientData.PatientData (
    int patientId,
    int doctorId,
    int nurseId,
    int diagnosisId,
    int patientStatus,
    PatientDataManipulations manipulations )
```

Constructor for creating new object, used for testing.

Parameters

<i>patientId</i>	corresponding patient id
<i>doctorId</i>	corresponding doctor id
<i>nurseId</i>	corresponding nurse id
<i>diagnosisId</i>	corresponding diagnosis id
<i>patientStatus</i>	corresponding patient status
<i>manipulations</i>	corresponding instance of PatientDataManipulations class

See also

[PatientData\(int,int,int,int,int\)](#)

3.22.3 Member Function Documentation

3.22.3.1 getCurrentPrescriptions()

```
Set<Prescription> hospital.entity.PatientData.getCurrentPrescriptions ( )
```

Gets current prescriptions

Returns

current prescriptions

3.22.3.2 getDiagnosis()

```
Diagnosis hospital.entity.PatientData.getDiagnosis ( )
```

Gets patient's diagnosis.

Returns

patient's diagnosis

3.22.3.3 getDoctor()

```
User hospital.entity.PatientData.getDoctor ( )
```

Gets patient's doctor.

Returns

patient's doctor

3.22.3.4 getId()

```
int hospital.entity.PatientData.getId ( )
```

Gets id of instance

Returns

id of instance

3.22.3.5 getNurse()

```
User hospital.entity.PatientData.getNurse ( )
```

Gets patient's nurse.

Returns

patient's nurse

3.22.3.6 getPatient()

```
User hospital.entity.PatientData.getPatient ( )
```

Gets patient

Returns

patient

3.22.3.7 getPatientStatus()

```
int hospital.entity.PatientData.getPatientStatus ( )
```

Gets patient's status.

Returns

patient's status

3.22.3.8 getPrescriptionHistory()

```
Set<Prescription> hospital.entity.PatientData.getPrescriptionHistory ( )
```

Gets prescription history

Returns

prescription history

3.22.3.9 setCurrentPrescriptions()

```
void hospital.entity.PatientData.setCurrentPrescriptions (
    Set< Prescription > currentPrescriptions )
```

Sets current prescriptions

Parameters

<i>currentPrescriptions</i>	current prescriptions to set
-----------------------------	------------------------------

3.22.3.10 setDiagnosis()

```
void hospital.entity.PatientData.setDiagnosis (
    Diagnosis diagnosis )
```

Sets patient's diagnosis

Parameters

<i>diagnosis</i>	patient's diagnosis to set
------------------	----------------------------

3.22.3.11 setDoctor()

```
void hospital.entity.PatientData.setDoctor (
    User doctor )
```

Sets patient's doctor

Parameters

<i>doctor</i>	patient's doctor to set
---------------	-------------------------

3.22.3.12 setId()

```
void hospital.entity.PatientData.setId (
    int id )
```

Sets id of instance

Parameters

<i>id</i>	id of instance to set
-----------	-----------------------

3.22.3.13 setNurse()

```
void hospital.entity.PatientData.setNurse (
    User nurse )
```

Sets patient's nurse

Parameters

<i>nurse</i>	patient's nurse to set
--------------	------------------------

3.22.3.14 setPatient()

```
void hospital.entity.PatientData.setPatient (
    User patient )
```

Sets patient

Parameters

<i>patient</i>	patient to set to current instance
----------------	------------------------------------

3.22.3.15 setPatientStatus()

```
void hospital.entity.PatientData.setPatientStatus (
    int patientStatus )
```

Sets patient's status

Parameters

<i>patientStatus</i>	patient's status to set
----------------------	-------------------------

3.22.3.16 setPrescriptionHistory()

```
void hospital.entity.PatientData.setPrescriptionHistory (
    Set< Prescription > prescriptionHistory )
```

Sets prescription history

Parameters

<i>prescriptionHistory</i>	prescription history to set
----------------------------	-----------------------------

3.22.4 Member Data Documentation

3.22.4.1 DIAGNOSIS_COLUMN

```
final String hospital.entity.PatientData.DIAGNOSIS_COLUMN = "diagnosis_id" [static]
```

The corresponding name of diagnosis id column in table.

3.22.4.2 DOCTOR_COLUMN

```
final String hospital.entity.PatientData.DOCTOR_COLUMN = "doctor_id" [static]
```

The corresponding name of doctor id column in table.

3.22.4.3 ID_COLUMN

```
final String hospital.entity.PatientData.ID_COLUMN = "id" [static]
```

The corresponding name of id column in table.

3.22.4.4 NURSE_COLUMN

```
final String hospital.entity.PatientData.NURSE_COLUMN = "nurse_id" [static]
```

The corresponding name of nurse id column in table.

3.22.4.5 PATIENT_COLUMN

```
final String hospital.entity.PatientData.PATIENT_COLUMN = "patient_id" [static]
```

The corresponding name of patient id column in table.

3.22.4.6 STATUS_COLUMN

```
final String hospital.entity.PatientData.STATUS_COLUMN = "patient_status" [static]
```

The corresponding name of patient status column in table.

3.22.4.7 TABLE_NAME

```
final String hospital.entity.PatientData.TABLE_NAME = "patient_info" [static]
```

The corresponding name of table in database.

The documentation for this class was generated from the following file:

- C:/Users/Liza/Desktop/Hospital/src/main/java/hospital/entity/PatientData.java

3.23 hospital.dao.impl.PatientDataImpl Class Reference

Public Member Functions

- [PatientData](#) [findByPatientId](#) (int id)
- List< [PatientData](#) > [findDoctorPatients](#) (int id)
- List< [PatientData](#) > [findNursePatients](#) (int id)
- void [insert](#) (int id)
- List< [PatientData](#) > [findAll](#) ()
- List< [PatientData](#) > [findAllWhereNoDoc](#) ()
- List< [PatientData](#) > [findAllWhereNoNurse](#) ()
- void [updateDoctor](#) (int doctorId, int patientId)
- void [updateNurse](#) (int nurseId, int patientId)
- void [updateDiagnosis](#) (int diagnosisId, int patientId)
- void [updatePatientStatus](#) (int id)

3.23.1 Detailed Description

Class, which represents PatientData DAO

Author

Yelyzaveta Onyshchenko

Version

1.01

3.23.2 Member Function Documentation

3.23.2.1 [findAll\(\)](#)

```
List<PatientData> hospital.dao.impl.PatientDataImpl.findAll ( )
```

Find all patient data in database

Returns

list of patient data

3.23.2.2 findAllWhereNoDoc()

```
List<PatientData> hospital.dao.impl.PatientDataImpl.findAllWhereNoDoc ( )
```

Find all patients which have no doctor at the moment

Returns

list of patients

3.23.2.3 findAllWhereNoNurse()

```
List<PatientData> hospital.dao.impl.PatientDataImpl.findAllWhereNoNurse ( )
```

Find all patients which have no nurse at the moment

Returns

list of patients

3.23.2.4 findByPatientId()

```
PatientData hospital.dao.impl.PatientDataImpl.findByPatientId (
    int id )
```

Find patient data by patient id

Returns

corresponding patient info

3.23.2.5 findDoctorPatients()

```
List<PatientData> hospital.dao.impl.PatientDataImpl.findDoctorPatients (
    int id )
```

Find existing doctor's patients by doctor id

Returns

list of patients

3.23.2.6 findNursePatients()

```
List<PatientData> hospital.dao.impl.PatientDataImpl.findNursePatients (
    int id )
```

Find existing nurse's patients by nurse id

Returns

list of patients

3.23.2.7 insert()

```
void hospital.dao.impl.PatientDataImpl.insert (
    int id )
```

Insert new patient to patient_info table

3.23.2.8 updateDiagnosis()

```
void hospital.dao.impl.PatientDataImpl.updateDiagnosis (
    int diagnosisId,
    int patientId )
```

Appoint diagnosis to the patient

3.23.2.9 updateDoctor()

```
void hospital.dao.impl.PatientDataImpl.updateDoctor (
    int doctorId,
    int patientId )
```

Appoint doctor to the patient

3.23.2.10 updateNurse()

```
void hospital.dao.impl.PatientDataImpl.updateNurse (
    int nurseId,
    int patientId )
```

Appoint nurse to the patient

3.23.2.11 updatePatientStatus()

```
void hospital.dao.impl.PatientDataImpl.updatePatientStatus (
    int id )
```

Change status of the patient.

The documentation for this class was generated from the following file:

- C:/Users/Liza/Desktop/Hospital/src/main/java/hospital/dao/impl/PatientDataImpl.java

3.24 hospital.utils.utilsForDBData.PatientDataManipulations Class Reference

Public Member Functions

- [PatientDataManipulations](#) ()
- [PatientDataManipulations](#) ([DAOFactory](#) factory)
- [User](#) [detectPerson](#) (int id)
- [Diagnosis](#) [detectDiagnosis](#) (int id)
- void [refreshPatients](#) ()
- void [checkPrescriptionHistory](#) ()
- void [createPatientHistory](#) (int patientId)
- void [makePrescriptions](#) ([PatientData](#) patientData)
- List< [Prescription](#) > [prepareActivePrescriptions](#) (Set< [Prescription](#) > pastPrescriptions, List< [Prescription](#) > listToPrepare)
- Set< [Prescription](#) > [findPatientHistory](#) (int patientId)
- Set< [Prescription](#) > [findNewPrescriptions](#) (int patientId)

3.24.1 Detailed Description

Class, which used for operations with patient data.

Author

Yelyzaveta Onyshchenko

Version

1.01

3.24.2 Constructor & Destructor Documentation

3.24.2.1 PatientDataManipulations() [1/2]

```
hospital.utils.utilsForDBData.PatientDataManipulations.PatientDataManipulations ( )
```

Constructor for creating new object.

See also

[PatientDataManipulations\(DAOFactory\)](#)

3.24.2.2 PatientDataManipulations() [2/2]

```
hospital.utils.utilsForDBData.PatientDataManipulations.PatientDataManipulations (
    DAOFactory factory )
```

Constructor for creating new object.

See also

[PatientDataManipulations\(\)](#)

3.24.3 Member Function Documentation

3.24.3.1 checkPrescriptionHistory()

```
void hospital.utils.utilsForDBData.PatientDataManipulations.checkPrescriptionHistory ( )
```

Method which checks that all patients have their prescription history

3.24.3.2 createPatientHistory()

```
void hospital.utils.utilsForDBData.PatientDataManipulations.createPatientHistory (
    int patientId )
```

Method which creates prescription history in case of absence

3.24.3.3 detectDiagnosis()

```
Diagnosis hospital.utils.utilsForDBData.PatientDataManipulations.detectDiagnosis (
    int id )
```

Method which defines diagnosis by id

Parameters

<i>id</i>	id of user
-----------	------------

Returns

corresponding instance of Diagnosis class

3.24.3.4 detectPerson()

```
User hospital.utils.utilsForDBData.PatientDataManipulations.detectPerson (  
    int id )
```

Method which defines user by id

Parameters

<i>id</i>	id of user
-----------	------------

Returns

corresponding instance of User class

3.24.3.5 findNewPrescriptions()

```
Set<Prescription> hospital.utils.utilsForDBData.PatientDataManipulations.findNewPrescriptions  
(  
    int patientId )
```

Finds new prescriptions in database

Parameters

<i>patient↔ Id</i>	id of patient
------------------------	---------------

Returns

new prescriptions for patient

3.24.3.6 findPatientHistory()

```
Set<Prescription> hospital.utils.utilsForDBData.PatientDataManipulations.findPatientHistory (
    int patientId )
```

Finds prescription history in database

Parameters

<i>patientId</i>	id of patient
------------------	---------------

Returns

prescription history of patient

3.24.3.7 makePrescriptions()

```
void hospital.utils.utilsForDBData.PatientDataManipulations.makePrescriptions (
    PatientData patientData )
```

Method which defines which prescriptions are necessary for patient based on his/her diagnosis

3.24.3.8 prepareActivePrescriptions()

```
List<Prescription> hospital.utils.utilsForDBData.PatientDataManipulations.prepareActivePrescriptions (
    Set< Prescription > pastPrescriptions,
    List< Prescription > listToPrepare )
```

Method which removes prescriptions that patient has already tried

3.24.3.9 refreshPatients()

```
void hospital.utils.utilsForDBData.PatientDataManipulations.refreshPatients ( )
```

Method which checks that all patient data is present in the database

The documentation for this class was generated from the following file:

- C:/Users/Liza/Desktop/Hospital/src/main/java/hospital/utils/utilsForDBData/PatientDataManipulations.java

3.25 hospital.utils.queries.PatientDataQueries Interface Reference

Static Public Attributes

- static final String [FIND_ALL](#) = "select * from " + PatientData.TABLE_NAME
- static final String [FIND_DOCTOR_PATIENTS](#)
- static final String [FIND_NURSE_PATIENTS](#)
- static final String [FIND_PATIENT](#)
- static final String [FIND_WITHOUT_DOC](#)
- static final String [FIND_WITHOUT_NURSE](#)
- static final String [INSERT_PATIENT](#)
- static final String [UPDATE_PATIENT_STATUS](#)
- static final String [ADD_DOCTOR](#) = "update " + PatientData.TABLE_NAME + " set " + PatientData.DOCTOR_↵
R_COLUMN + " = ? where " + PatientData.PATIENT_COLUMN + " = ?"
- static final String [ADD_NURSE](#) = "update " + PatientData.TABLE_NAME + " set " + PatientData.NURSE_↵
COLUMN + " = ? where " + PatientData.PATIENT_COLUMN + " = ?"
- static final String [ADD_DIAGNOSIS](#) = "update " + PatientData.TABLE_NAME + " set " + PatientData.DIA_↵
GNOSIS_COLUMN + " = ? where " + PatientData.PATIENT_COLUMN + " = ?"

3.25.1 Detailed Description

Basic queries for patient data table.

Author

Yelyzaveta Onyshchenko

Version

1.01

3.25.2 Member Data Documentation

3.25.2.1 ADD_DIAGNOSIS

```
final String hospital.utils.queries.PatientDataQueries.ADD_DIAGNOSIS = "update " + Patient↵
Data.TABLE_NAME + " set " + PatientData.DIAGNOSIS_COLUMN + " = ?  where " + PatientData.PATIE↵
NT_COLUMN + " = ?"  [static]
```

```
UPDATE patient_info SET diagnosis_id = ? WHERE patient_id = ?;
```

3.25.2.2 ADD_DOCTOR

```
final String hospital.utils.queries.PatientDataQueries.ADD_DOCTOR = "update " + PatientData.↵
TABLE_NAME + " set " + PatientData.DOCTOR_COLUMN + " = ?  where " + PatientData.PATIENT_COLUMN
+ " = ?"  [static]
```

```
UPDATE patient_info SET doctor_id = ? WHERE patient_id = ?;
```

3.25.2.3 ADD_NURSE

```
final String hospital.utils.queries.PatientDataQueries.ADD_NURSE = "update " + PatientData.TABLE_NAME + " set " + PatientData.NURSE_COLUMN + " = ? where " + PatientData.PATIENT_COLUMN + " = ?" [static]
```

```
UPDATE patient_info SET nurse_id = ? WHERE patient_id = ?;
```

3.25.2.4 FIND_ALL

```
final String hospital.utils.queries.PatientDataQueries.FIND_ALL = "select * from " + PatientData.TABLE_NAME [static]
```

```
SELECT * FROM patient_info;
```

3.25.2.5 FIND_DOCTOR_PATIENTS

```
final String hospital.utils.queries.PatientDataQueries.FIND_DOCTOR_PATIENTS [static]
```

Initial value:

```
= "select * from " + PatientData.TABLE_NAME + " where " + PatientData.DOCTOR_COLUMN + " = ? and " + PatientData.STATUS_COLUMN + " = 1"
```

```
SELECT * FROM patient_info WHERE doctor_id = ? AND patient_status = 1;
```

3.25.2.6 FIND_NURSE_PATIENTS

```
final String hospital.utils.queries.PatientDataQueries.FIND_NURSE_PATIENTS [static]
```

Initial value:

```
= "select * from " + PatientData.TABLE_NAME + " where " + PatientData.NURSE_COLUMN + " = ? and " + PatientData.STATUS_COLUMN + " = 1"
```

```
SELECT * FROM patient_info WHERE nurse_id = ? AND patient_status = 1;
```

3.25.2.7 FIND_PATIENT

```
final String hospital.utils.queries.PatientDataQueries.FIND_PATIENT [static]
```

Initial value:

```
= "select * from " + PatientData.TABLE_NAME + " where " + PatientData.PATIENT_COLUMN + " = ?"
```

```
SELECT * FROM patient_info WHERE patient_id = ?
```

3.25.2.8 FIND_WITHOUT_DOC

```
final String hospital.utils.queries.PatientDataQueries.FIND_WITHOUT_DOC [static]
```

Initial value:

```
= "select * from " + PatientData.TABLE_NAME + " where " + PatientData.DOCTOR_COLUMN + " IS NULL and " + PatientData.STATUS_COLUMN + " = 1"
```

```
SELECT * FROM patient_info WHERE doctor_id IS NULL AND patient_status = 1;
```

3.25.2.9 FIND_WITHOUT_NURSE

```
final String hospital.utils.queries.PatientDataQueries.FIND_WITHOUT_NURSE [static]
```

Initial value:

```
= "select * from " + PatientData.TABLE_NAME + " where "  
  + PatientData.NURSE_COLUMN + " IS NULL and " + PatientData.STATUS_COLUMN + " = 1"
```

```
SELECT * FROM patient_info WHERE nurse_id IS NULL AND patient_status = 1;
```

3.25.2.10 INSERT_PATIENT

```
final String hospital.utils.queries.PatientDataQueries.INSERT_PATIENT [static]
```

Initial value:

```
= "insert into " + PatientData.TABLE_NAME  
  + " (" + PatientData.PATIENT_COLUMN + ") values (?)"
```

```
INSERT INTO patient_info (patient_id) VALUES (?);
```

3.25.2.11 UPDATE_PATIENT_STATUS

```
final String hospital.utils.queries.PatientDataQueries.UPDATE_PATIENT_STATUS [static]
```

Initial value:

```
= "update " + PatientData.TABLE_NAME + " set " + PatientData.STATUS_COLUMN + " = 0, " +  
  PatientData.DOCTOR_COLUMN + " = NULL, " + PatientData.NURSE_COLUMN + " = NULL where " +  
  PatientData.PATIENT_COLUMN + " = ?"
```

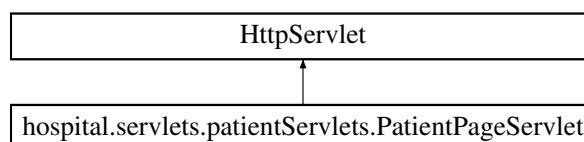
```
UPDATE patient_info SET patient_status = 0, doctor_id = NULL, nurse_id = NULL WHERE patient_id = ?;
```

The documentation for this interface was generated from the following file:

- C:/Users/Liza/Desktop/Hospital/src/main/java/hospital/utils/queries/PatientDataQueries.java

3.26 hospital.servlets.patientServlets.PatientPageServlet Class Reference

Inheritance diagram for hospital.servlets.patientServlets.PatientPageServlet:



Public Member Functions

- [PatientPageServlet \(\)](#)
- [PatientPageServlet \(DAOFactory factory\)](#)

Protected Member Functions

- void [doPost](#) (HttpServletRequest request, HttpServletResponse response) throws ServletException, IOException

3.26.1 Detailed Description

Class, which represents patient page servlet.

Author

Yelyzaveta Onyshchenko

Version

1.01

3.26.2 Constructor & Destructor Documentation

3.26.2.1 PatientPageServlet() [1/2]

```
hospital.servlets.patientServlets.PatientPageServlet.PatientPageServlet ( )
```

Constructor for creating new object

See also

[PatientPageServlet\(DAOFactory\)](#)

3.26.2.2 PatientPageServlet() [2/2]

```
hospital.servlets.patientServlets.PatientPageServlet.PatientPageServlet (
    DAOFactory factory )
```

Constructor for creating new object

Parameters

<i>factory</i>	instance of DAO factory
----------------	-------------------------

See also

[PatientPageServlet\(DAOFactory\)](#)

3.26.3 Member Function Documentation

3.26.3.1 doPost()

```
void hospital.servlets.patientServlets.PatientPageServlet.doPost (
    HttpServletRequest request,
    HttpServletResponse response ) throws ServletException, IOException [protected]
```

Method performing HHTP POST request

Parameters

<i>request</i>	HttpRequest
<i>response</i>	HttpResponse

Exceptions

<i>ServletException</i>	On servlet error
<i>IOException</i>	On error fulfilling the request

The documentation for this class was generated from the following file:

- C:/Users/Liza/Desktop/Hospital/src/main/java/hospital/servlets/patientServlets/PatientPageServlet.java

3.27 hospital.entity.prescription.Prescription Class Reference

Public Member Functions

- [Prescription](#) (int id, String name, String prescriptionClass)
- [Prescription](#) (String name, int amount)
- int [getAmountOfMedicine](#) ()
- void [setAmountOfMedicine](#) (int amountOfMedicine)
- int [getId](#) ()
- void [setId](#) (int id)
- String [getName](#) ()
- void [setName](#) (String name)
- String [getPrescriptionClass](#) ()
- void [setPrescriptionClass](#) (String prescriptionClass)

Static Public Attributes

- static final String `TABLE_NAME` = "prescription_table"
- static final String `ID_COLUMN` = "id_prescription"
- static final String `NAME_COLUMN` = "prescription_name"
- static final String `CLASS_COLUMN` = "prescription_class"
- static final String `CONNECTED_TABLE_HISTORY` = "previous_prescriptions"
- static final String `CONNECTED_TABLE_HISTORY_ID` = "id"
- static final String `CONNECTED_TABLE_HISTORY_PATIENT` = "patient_id"
- static final String `CONNECTED_TABLE_HISTORY_PRESCR` = "prescription_id"
- static final String `CONNECTED_TABLE_NEW` = "new_prescriptions"
- static final String `CONNECTED_TABLE_NEW_ID` = "id"
- static final String `CONNECTED_TABLE_NEW_PATIENT` = "patient_id"
- static final String `CONNECTED_TABLE_NEW_PRESCR` = "prescription_id"
- static final String `MED_TABLE` = "medicine_list"
- static final String `MED_TABLE_NAME` = "medicine_name"
- static final String `MED_TABLE_AMOUNT` = "medicine_amount"

3.27.1 Detailed Description

Class, which represents prescription which can be set to the patient.

Author

Yelyzaveta Onyshchenko

Version

1.01

3.27.2 Constructor & Destructor Documentation

3.27.2.1 Prescription() [1/2]

```
hospital.entity.prescription.Prescription.Prescription (
    int id,
    String name,
    String prescriptionClass )
```

Constructor for creating new object.

Parameters

<i>id</i>	corresponding id of the prescription
<i>name</i>	coresponding name of the prescription
<i>prescriptionClass</i>	corresponding class of the prescription

3.27.2.2 Prescription() [2/2]

```
hospital.entity.prescription.Prescription.Prescription (
    String name,
    int amount )
```

Constructor for creating new object of medical prescription.

Parameters

<i>name</i>	coresponding name of the prescription
<i>amount</i>	coresponding amount of the prescription

3.27.3 Member Function Documentation

3.27.3.1 getAmountOfMedicine()

```
int hospital.entity.prescription.Prescription.getAmountOfMedicine ( )
```

Method, which returns amount of medicine

Returns

amount of medicine

3.27.3.2 getId()

```
int hospital.entity.prescription.Prescription.getId ( )
```

Method, which returns id of prescription

Returns

id of prescription

3.27.3.3 getName()

```
String hospital.entity.prescription.Prescription.getName ( )
```

Method, which returns name of prescription

Returns

name of prescription

3.27.3.4 getPrescriptionClass()

```
String hospital.entity.prescription.Prescription.getPrescriptionClass ( )
```

Method, which returns prescription class

Returns

class of prescription

3.27.3.5 setAmountOfMedicine()

```
void hospital.entity.prescription.Prescription.setAmountOfMedicine (
    int amountOfMedicine )
```

Method, which sets amount of medicine

Parameters

<i>amountOfMedicine</i>	amount of medicine
-------------------------	--------------------

3.27.3.6 setId()

```
void hospital.entity.prescription.Prescription.setId (
    int id )
```

Sets the id of prescription

Parameters

<i>id</i>	id of prescription
-----------	--------------------

3.27.3.7 setName()

```
void hospital.entity.prescription.Prescription.setName (  
    String name )
```

Sets name of prescription

Parameters

<i>name</i>	name of prescription
-------------	----------------------

3.27.3.8 setPrescriptionClass()

```
void hospital.entity.prescription.Prescription.setPrescriptionClass (  
    String prescriptionClass )
```

Sets class of prescription

Parameters

<i>prescriptionClass</i>	prescription class to set
--------------------------	---------------------------

3.27.4 Member Data Documentation

3.27.4.1 CLASS_COLUMN

```
final String hospital.entity.prescription.Prescription.CLASS_COLUMN = "prescription_class"  
[static]
```

The corresponding name of prescription class column in table.

3.27.4.2 CONNECTED_TABLE_HISTORY

```
final String hospital.entity.prescription.Prescription.CONNECTED_TABLE_HISTORY = "previous_↔  
prescriptions" [static]
```

The corresponding name of previous prescriptions table in database.

3.27.4.3 CONNECTED_TABLE_HISTORY_ID

```
final String hospital.entity.prescription.Prescription.CONNECTED_TABLE_HISTORY_ID = "id" [static]
```

The corresponding name of id column in table.

3.27.4.4 CONNECTED_TABLE_HISTORY_PATIENT

```
final String hospital.entity.prescription.Prescription.CONNECTED_TABLE_HISTORY_PATIENT = "patient↔  
_id" [static]
```

The corresponding name of patient id column in table.

3.27.4.5 CONNECTED_TABLE_HISTORY_PRESCR

```
final String hospital.entity.prescription.Prescription.CONNECTED_TABLE_HISTORY_PRESCR = "prescription↔  
_id" [static]
```

The corresponding name of prescription id column in table.

3.27.4.6 CONNECTED_TABLE_NEW

```
final String hospital.entity.prescription.Prescription.CONNECTED_TABLE_NEW = "new_prescriptions"  
[static]
```

The corresponding name of new prescriptions table in database.

3.27.4.7 CONNECTED_TABLE_NEW_ID

```
final String hospital.entity.prescription.Prescription.CONNECTED_TABLE_NEW_ID = "id" [static]
```

The corresponding name of id column in table.

3.27.4.8 CONNECTED_TABLE_NEW_PATIENT

```
final String hospital.entity.prescription.Prescription.CONNECTED_TABLE_NEW_PATIENT = "patient↔  
_id" [static]
```

The corresponding name of patient id column in table.

3.27.4.9 CONNECTED_TABLE_NEW_PRESCR

```
final String hospital.entity.prescription.Prescription.CONNECTED_TABLE_NEW_PRESCR = "prescription↔  
_id" [static]
```

The corresponding name of prescription id column in table.

3.27.4.10 ID_COLUMN

```
final String hospital.entity.prescription.Prescription.ID_COLUMN = "id_prescription" [static]
```

The corresponding name of id column in table.

3.27.4.11 MED_TABLE

```
final String hospital.entity.prescription.Prescription.MED_TABLE = "medicine_list" [static]
```

The corresponding name of medicine list table in database.

3.27.4.12 MED_TABLE_AMOUNT

```
final String hospital.entity.prescription.Prescription.MED_TABLE_AMOUNT = "medicine_amount" [static]
```

The corresponding name of medicine amount column in table.

3.27.4.13 MED_TABLE_NAME

```
final String hospital.entity.prescription.Prescription.MED_TABLE_NAME = "medicine_name" [static]
```

The corresponding name of medicine name column in table.

3.27.4.14 NAME_COLUMN

```
final String hospital.entity.prescription.Prescription.NAME_COLUMN = "prescription_name" [static]
```

The corresponding name of prescription name in table.

3.27.4.15 TABLE_NAME

```
final String hospital.entity.prescription.Prescription.TABLE_NAME = "prescription_table" [static]
```

The corresponding name of table in database.

The documentation for this class was generated from the following file:

- C:/Users/Liza/Desktop/Hospital/src/main/java/hospital/entity/prescription/Prescription.java

3.28 hospital.dao.impl.PrescriptionDAOImpl Class Reference

Public Member Functions

- List< [Prescription](#) > [findAll](#) ()
- [Prescription](#) [findById](#) (int id)
- List< Integer > [findByPatientHistory](#) (int id)
- List< Integer > [findByActivePrescriptions](#) (int id)
- List< [Prescription](#) > [findByClass](#) (String prescriptionClass)
- void [insertHistory](#) (int prescriptionId, int patientId)
- void [insertNewPrescriptions](#) (int prescriptionId, int patientId)
- void [delete](#) (int id, int patientId)

3.28.1 Detailed Description

Class, which represents Prescription DAO

Author

Yelyzaveta Onyshchenko

Version

1.01

3.28.2 Member Function Documentation

3.28.2.1 delete()

```
void hospital.dao.impl.PrescriptionDAOImpl.delete (
    int id,
    int patientId )
```

Delete prescription for particular patient

3.28.2.2 findAll()

```
List<Prescription> hospital.dao.impl.PrescriptionDAOImpl.findAll ( )
```

Find all prescriptions

Returns

list of prescriptions

3.28.2.3 findByActivePrescriptions()

```
List<Integer> hospital.dao.impl.PrescriptionDAOImpl.findByActivePrescriptions (
    int id )
```

Find patients new prescriptions

Returns

new prescriptions - list of prescriptions id

3.28.2.4 findByClass()

```
List<Prescription> hospital.dao.impl.PrescriptionDAOImpl.findByClass (
    String prescriptionClass )
```

Find prescription by prescription class

Returns

list of corresponding prescriptions

3.28.2.5 findById()

```
Prescription hospital.dao.impl.PrescriptionDAOImpl.findById (
    int id )
```

Find prescription by id

Returns

corresponding prescription

3.28.2.6 findByPatientHistory()

```
List<Integer> hospital.dao.impl.PrescriptionDAOImpl.findByPatientHistory (
    int id )
```

Find patients prescription history

Returns

prescription history - list of prescriptions id

3.28.2.7 insertHistory()

```
void hospital.dao.impl.PrescriptionDAOImpl.insertHistory (
    int prescriptionId,
    int patientId )
```

Insert patient's history in table

3.28.2.8 insertNewPrescriptions()

```
void hospital.dao.impl.PrescriptionDAOImpl.insertNewPrescriptions (
    int prescriptionId,
    int patientId )
```

Insert new prescriptions for patient in table

The documentation for this class was generated from the following file:

- C:/Users/Liza/Desktop/Hospital/src/main/java/hospital/dao/impl/PrescriptionDAOImpl.java

3.29 hospital.utils.queries.PrescriptionQueries Interface Reference

Static Public Attributes

- static final String [FIND_ALL](#) = "select * from " + Prescription.TABLE_NAME
- static final String [FIND_ALL_HISTORY](#)
- static final String [FIND_ALL_NEW](#)
- static final String [FIND_ALL_MED](#) = "select * from " + Prescription.MED_TABLE
- static final String [FIND_BY_ID](#) = [FIND_ALL](#) + " where " + Prescription.ID_COLUMN + " = ?"
- static final String [FIND_BY_CLASS](#) = [FIND_ALL](#) + " where " + Prescription.CLASS_COLUMN + " = ?"
- static final String [INSERT_PRESCRIPTION_HISTORY](#)
- static final String [INSERT_NEW_PRESCRIPTION](#)
- static final String [DELETE_PRESCRIPTION](#)
- static final String [FIND_MED_LIST](#) = [FIND_ALL_MED](#) + " where " + Prescription.MED_TABLE_NAME + " = ?"
- static final String [UPDATE_MED_LIST](#)

3.29.1 Detailed Description

Basic queries for prescription table.

Author

Yelyzaveta Onyshchenko

Version

1.01

3.29.2 Member Data Documentation

3.29.2.1 DELETE_PRESCRIPTION

```
final String hospital.utils.queries.PrescriptionQueries.DELETE_PRESCRIPTION [static]
```

Initial value:

```
= "delete from " + Prescription.CONNECTED_TABLE_NEW + " where " + Prescription.CONNECTED_TABLE_NEW_PRESCR  
+ " = ? and " + Prescription.CONNECTED_TABLE_NEW_PATIENT + " = ?"
```

```
DELETE FROM new_prescriptions WHERE prescription_id = ? AND patient_id = ?;
```

3.29.2.2 FIND_ALL

```
final String hospital.utils.queries.PrescriptionQueries.FIND_ALL = "select * from " + Prescription.↵  
TABLE_NAME [static]
```

```
SELECT * FROM prescription_table;
```

3.29.2.3 FIND_ALL_HISTORY

```
final String hospital.utils.queries.PrescriptionQueries.FIND_ALL_HISTORY [static]
```

Initial value:

```
= "select * from " + Prescription.CONNECTED_TABLE_HISTORY + " where " +  
Prescription.CONNECTED_TABLE_HISTORY_PATIENT + " = ?"
```

```
SELECT * FROM previous_prescriptions WHERE patient_id = ?;
```

3.29.2.4 FIND_ALL_MED

```
final String hospital.utils.queries.PrescriptionQueries.FIND_ALL_MED = "select * from " +  
Prescription.MED_TABLE [static]
```

```
SELECT * FROM medicine_list;
```

3.29.2.5 FIND_ALL_NEW

```
final String hospital.utils.queries.PrescriptionQueries.FIND_ALL_NEW [static]
```

Initial value:

```
= "select * from " + Prescription.CONNECTED_TABLE_NEW + " where " +  
Prescription.CONNECTED_TABLE_NEW_PATIENT + " = ?"
```

```
SELECT * FROM new_prescriptions WHERE patient_id = ?;
```


3.29.2.6 FIND_BY_CLASS

```
final String hospital.utils.queries.PrescriptionQueries.FIND_BY_CLASS = FIND_ALL + " where " +
Prescription.CLASS_COLUMN + " = ?" [static]
```

```
SELECT * FROM prescription_table WHERE prescription_class = ?;
```

3.29.2.7 FIND_BY_ID

```
final String hospital.utils.queries.PrescriptionQueries.FIND_BY_ID = FIND_ALL + " where " +
Prescription.ID_COLUMN + " = ?" [static]
```

```
SELECT * FROM prescription_table WHERE id_prescription = ?;
```

3.29.2.8 FIND_MED_LIST

```
final String hospital.utils.queries.PrescriptionQueries.FIND_MED_LIST = FIND_ALL_MED + " where
" + Prescription.MED_TABLE_NAME + " = ?" [static]
```

```
SELECT * FROM medicine_list WHERE medicine_name = ?;
```

3.29.2.9 INSERT_NEW_PRESCRIPTION

```
final String hospital.utils.queries.PrescriptionQueries.INSERT_NEW_PRESCRIPTION [static]
```

Initial value:

```
= "insert into " + Prescription.CONNECTED_TABLE_NEW
    + " (" + Prescription.CONNECTED_TABLE_NEW_PATIENT + ", " +
    Prescription.CONNECTED_TABLE_NEW_PRESCR
    + ") values (?, ?)"
```

```
INSERT INTO new_prescriptions(patient_id,prescription_id) VALUES (?,?);
```

3.29.2.10 INSERT_PRESCRIPTION_HISTORY

```
final String hospital.utils.queries.PrescriptionQueries.INSERT_PRESCRIPTION_HISTORY [static]
```

Initial value:

```
= "insert into " + Prescription.CONNECTED_TABLE_HISTORY
    + " (" + Prescription.CONNECTED_TABLE_HISTORY_PATIENT + ", " +
    Prescription.CONNECTED_TABLE_HISTORY_PRESCR
    + ") values (?, ?)"
```

```
INSERT INTO previous_prescriptions(patient_id,prescription_id) VALUES (?,?);
```

3.29.2.11 UPDATE_MED_LIST

```
final String hospital.utils.queries.PrescriptionQueries.UPDATE_MED_LIST [static]
```

Initial value:

```
= "update " + Prescription.MED_TABLE + " set " + Prescription.MED_TABLE_AMOUNT +
    " = ? where " + Prescription.MED_TABLE_NAME + " = ?"
```

```
UPDATE medicine_list SET medicine_amount = ? WHERE medicine_name = ?;
```

The documentation for this interface was generated from the following file:

- C:/Users/Liza/Desktop/Hospital/src/main/java/hospital/utils/queries/PrescriptionQueries.java

3.30 hospital.utils.RandomNumber Class Reference

Static Public Member Functions

- static int [randNumber](#) (int min, int max)

3.30.1 Detailed Description

Class, which used to random number

Author

Yelyzaveta Onyshchenko

Version

1.01

3.30.2 Member Function Documentation

3.30.2.1 randNumber()

```
static int hospital.utils.RandomNumber.randNumber (  
    int min,  
    int max ) [static]
```

Method which generates random number

Parameters

<i>min</i>	min number
<i>max</i>	max number

Returns

generated number

The documentation for this class was generated from the following file:

- C:/Users/Liza/Desktop/Hospital/src/main/java/hospital/Utils/RandomNumber.java

3.31 hospital.utils.RandomPatient Class Reference

Public Member Functions

- [RandomPatient](#) ()

- [RandomPatient](#) ([DAOFactory](#) factory)
- void [randomPatientNurse](#) (int id)
- void [randomPatientDoc](#) (int id)

3.31.1 Detailed Description

Class, which used to random patient for doctor or nurse

Author

Yelyzaveta Onyshchenko

Version

1.01

3.31.2 Constructor & Destructor Documentation

3.31.2.1 RandomPatient() [1/2]

```
hospital.utils.RandomPatient.RandomPatient ( )
```

Constructor for creating new object.

See also

[RandomPatient\(DAOFactory\)](#)

3.31.2.2 RandomPatient() [2/2]

```
hospital.utils.RandomPatient.RandomPatient (
    DAOFactory factory )
```

Constructor for creating new object.

See also

[RandomPatient\(\)](#)

3.31.3 Member Function Documentation

3.31.3.1 randomPatientDoc()

```
void hospital.utils.RandomPatient.randomPatientDoc (
    int id )
```

Create patient for doctor

Parameters

<i>id</i>	id of doctor
-----------	--------------

3.31.3.2 randomPatientNurse()

```
void hospital.utils.RandomPatient.randomPatientNurse (  
    int id )
```

Create patient for nurse

Parameters

<i>id</i>	id of nurse
-----------	-------------

The documentation for this class was generated from the following file:

- C:/Users/Liza/Desktop/Hospital/src/main/java/hospital/utils/RandomPatient.java

3.32 hospital.utils.RegexContainer Interface Reference**Public Attributes**

- String [REGEX_NAME_ENG](#) = "^[A-Z][a-z]{3,29}\$"
- String [REGEX_EMAIL](#) = "([a-z0-9_-]+\\.)*[a-z0-9_-]+\\@[a-z0-9_-]+(\\.[a-z0-9_-]+)*\\.[a-z]{2,6}\$"
- String [REGEX_PASSWORD](#) = "^[A-Za-z0-9_-]{3,30}\$"

3.32.1 Detailed Description

Basic interface for regular expressions

Author

Yelyzaveta Onyshchenko

Version

1.01

3.32.2 Member Data Documentation

3.33.2 Constructor & Destructor Documentation

3.33.2.1 ReleasePatientCommand() [1/2]

```
hospital.utils.doctorcommands.ReleasePatientCommand.ReleasePatientCommand ( )
```

Constructor for creating new object

See also

[ReleasePatientCommand\(DAOFactory\)](#)

3.33.2.2 ReleasePatientCommand() [2/2]

```
hospital.utils.doctorcommands.ReleasePatientCommand.ReleasePatientCommand (
    DAOFactory factory )
```

Constructor for creating new object

See also

[ReleasePatientCommand\(\)](#)

3.33.3 Member Function Documentation

3.33.3.1 execute()

```
void hospital.utils.doctorcommands.ReleasePatientCommand.execute (
    HttpServletRequest request,
    HttpServletResponse response ) throws ServletException, IOException
```

Execute command method

Parameters

<i>request</i>	HttpRequest
<i>response</i>	HttpResponse

Exceptions

<i>ServletException</i>	on servlet exception
-------------------------	----------------------

Exceptions

<i>IOException</i>	on error occurred while processing the request
--------------------	--

Implements [hospital.utils.factories.ServletCommand](#).

The documentation for this class was generated from the following file:

- C:/Users/Liza/Desktop/Hospital/src/main/java/hospital/utills/doctorcommands/ReleasePatientCommand.java

3.34 hospital.entity.Role Class Reference

Public Member Functions

- [Role](#) (int id, String name)
- String [getName](#) ()
- void [setName](#) (String name)
- int [getId](#) ()
- void [setId](#) (int id)

Static Public Attributes

- static final String [TABLE_NAME](#) = "user_roles"
- static final String [ID_COLUMN](#) = "role_id"
- static final String [NAME_COLUMN](#) = "name"

3.34.1 Detailed Description

Class, which represents possible role in system.

Author

Yelyzaveta Onyshchenko

Version

1.01

3.34.2 Constructor & Destructor Documentation

3.34.2.1 Role()

```
hospital.entity.Role.Role (
    int id,
    String name )
```

Constructor for creating new object.

Parameters

<i>id</i>	corresponding id
<i>name</i>	corresponding name of the role

3.34.3 Member Function Documentation

3.34.3.1 getId()

```
int hospital.entity.Role.getId ( )
```

Gets id of the role

Returns

id of the role

3.34.3.2 getName()

```
String hospital.entity.Role.getName ( )
```

Gets name of the role

Returns

name of the role

3.34.3.3 setId()

```
void hospital.entity.Role.setId (
    int id )
```

Sets id of the role

Parameters

<i>id</i>	id of the role to set
-----------	-----------------------

3.34.3.4 setName()

```
void hospital.entity.Role.setName (
    String name )
```

Sets name of the role

Parameters

<i>name</i>	name of the role to set
-------------	-------------------------

3.34.4 Member Data Documentation

3.34.4.1 ID_COLUMN

```
final String hospital.entity.Role.ID_COLUMN = "role_id" [static]
```

The corresponding name of id column in table.

3.34.4.2 NAME_COLUMN

```
final String hospital.entity.Role.NAME_COLUMN = "name" [static]
```

The corresponding name of name column in table.

3.34.4.3 TABLE_NAME

```
final String hospital.entity.Role.TABLE_NAME = "user_roles" [static]
```

The corresponding name of table in database.

The documentation for this class was generated from the following file:

- C:/Users/Liza/Desktop/Hospital/src/main/java/hospital/entity/Role.java

3.35 hospital.dao.impl.RoleDAOImpl Class Reference

Public Member Functions

- [Role findById](#) (int id)
- [Role findByName](#) (String name)

3.35.1 Detailed Description

Class, which represents Role DAO

Author

Yelyzaveta Onyshchenko

Version

1.01

3.35.2 Member Function Documentation

3.35.2.1 findById()

```
Role hospital.dao.impl.RoleDAOImpl.findById (
    int id )
```

Find role by id

Returns

corresponding role

3.35.2.2 findByName()

```
Role hospital.dao.impl.RoleDAOImpl.findByName (
    String name )
```

Find role by name

Returns

corresponding role

The documentation for this class was generated from the following file:

- C:/Users/Liza/Desktop/Hospital/src/main/java/hospital/dao/impl/RoleDAOImpl.java

3.36 hospital.utils.queries.RoleQueries Interface Reference

Static Public Attributes

- static final String [FIND_BY_ID](#)
- static final String [FIND_BY_NAME](#)

3.36.1 Detailed Description

Basic queries for ROLE table.

Author

Yelyzaveta Onyshchenko

Version

1.01

3.36.2 Member Data Documentation

3.36.2.1 FIND_BY_ID

```
final String hospital.utils.queries.RoleQueries.FIND_BY_ID [static]
```

Initial value:

```
= "select * from " + Role.TABLE_NAME + " where "  
  + Role.ID_COLUMN + " = ?"
```

```
SELECT * FROM user_roles WHERE role_id = ?;
```

3.36.2.2 FIND_BY_NAME

```
final String hospital.utils.queries.RoleQueries.FIND_BY_NAME [static]
```

Initial value:

```
= "select * from " + Role.TABLE_NAME + " where "  
  + Role.NAME_COLUMN + " = ?"
```

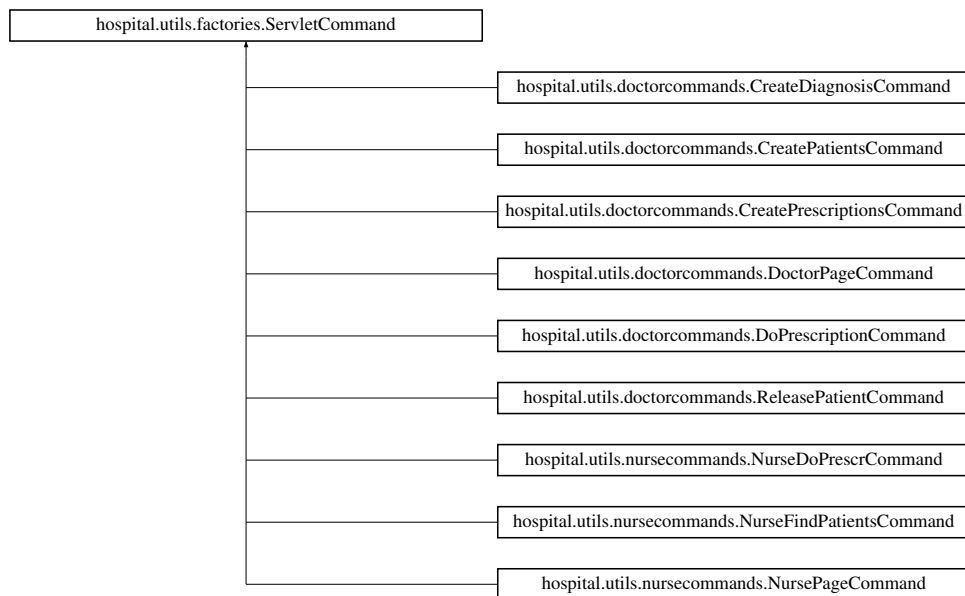
```
SELECT * FROM user_roles WHERE name = ?;
```

The documentation for this interface was generated from the following file:

- C:/Users/Liza/Desktop/Hospital/src/main/java/hospital/utils/queries/RoleQueries.java

3.37 hospital.utils.factories.ServletCommand Interface Reference

Inheritance diagram for hospital.utils.factories.ServletCommand:



Public Member Functions

- void [execute](#) (HttpServletRequest request, HttpServletResponse response) throws ServletException, IOException

3.37.1 Detailed Description

Basic interface for servlet command.

Author

Yelyzaveta Onyshchenko

Version

1.01

3.37.2 Member Function Documentation

3.37.2.1 execute()

```
void hospital.utils.factories.ServletCommand.execute (
    HttpServletRequest request,
    HttpServletResponse response ) throws ServletException, IOException
```

Instructions for command to perform.

Parameters

<i>request</i>	HttpRequest
<i>response</i>	HttpResponse

Exceptions

<i>ServletException</i>	on servlet exception
<i>IOException</i>	on error while performing the request

Implemented in [hospital.utils.doctorcommands.CreatePrescriptionsCommand](#), [hospital.utils.doctorcommands.CreateDiagnosisCommand](#), [hospital.utils.nursecommands.NursePageCommand](#), [hospital.utils.doctorcommands.DoctorPageCommand](#), [hospital.utils.doctorcommands.ReleasePatientCommand](#), [hospital.utils.doctorcommands.DoPrescriptionCommand](#), [hospital.utils.doctorcommands.CreatePatientsCommand](#), [hospital.utils.nursecommands.NurseDoPrescrCommand](#), and [hospital.utils.nursecommands.NurseFindPatientsCommand](#).

The documentation for this interface was generated from the following file:

- C:/Users/Liza/Desktop/Hospital/src/main/java/hospital/utils/factories/ServletCommand.java

3.38 hospital.utils.SingleTransaction Class Reference

Public Member Functions

- void [transfer](#) (int id, int patientId) throws SQLException

Static Public Member Functions

- static synchronized [SingleTransaction getInstance](#) ()

3.38.1 Detailed Description

Class, which used for transaction for medical prescriptions

Author

Yelyzaveta Onyshchenko

Version

1.01

3.38.2 Member Function Documentation

3.38.2.1 getInstance()

```
static synchronized SingleTransaction hospital.utils.SingleTransaction.getInstance ( ) [static]
```

Gets instance of [SingleTransaction](#)

Returns

[SingleTransaction](#) instance

3.38.2.2 transfer()

```
void hospital.utils.SingleTransaction.transfer (
    int id,
    int patientId ) throws SQLException
```

Method that performs transaction

Parameters

<i>id</i>	id of prescription
<i>patientId</i>	patient id

Exceptions

<i>SQLException</i>	SQLException which can occur during transaction
---------------------	---

The documentation for this class was generated from the following file:

- C:/Users/Liza/Desktop/Hospital/src/main/java/hospital/utils/SingleTransaction.java

3.39 hospital.utils.URLBase Enum Reference

Public Member Functions

- [URLBase](#) (String url)
- String [getUrl](#) ()

Static Public Member Functions

- static [URLBase](#) [getValue](#) (String value)

Public Attributes

- **NURSE_PAGE** = ("http://localhost:8000/nursePage")
- **DO_PRESCR_NURSE** = ("http://localhost:8000/nursePage/doPrescriptionNurse")
- **CREATE_NURSE_PATIENTS** = ("http://localhost:8000/nursePage/createNursePatients")
- **DOCTOR_PAGE** = ("http://localhost:8000/doctorPage")
- **CREATE_DIAGNOSIS** = ("http://localhost:8000/doctorPage/createPatientDiagnosis")
- **CREATE_PRESCR** = ("http://localhost:8000/doctorPage/createPatientPrescriptions")
- **DO_PRESCR_DOCTOR** = ("http://localhost:8000/doctorPage/doPrescriptionDoctor")
- **RELEASE_PATIENT** = ("http://localhost:8000/doctorPage/releasePatient")
- **CREATEPATIENTSDOC** = ("http://localhost:8000/doctorPage/createPatients")

3.39.1 Detailed Description

Enum which stores URLs of project pages

Author

Yelyzaveta Onyshchenko

Version

1.01

3.39.2 Constructor & Destructor Documentation

3.39.2.1 URLBase()

```
hospital.utils.URLBase.URLBase (  
    String url )
```

Constructor

Parameters

<i>url</i>	url of page
------------	-------------

3.39.3 Member Function Documentation

3.39.3.1 getUrl()

```
String hospital.utils.URLBase.getUrl ( )
```

Returns url

Returns

page url

3.39.3.2 getValue()

```
static URLBase hospital.utils.URLBase.getValue (
    String value ) [static]
```

Retrieves value

Parameters

<i>value</i>	url, which used to find value
--------------	-------------------------------

Returns

value

The documentation for this enum was generated from the following file:

- C:/Users/Liza/Desktop/Hospital/src/main/java/hospital/utils/URLBase.java

3.40 hospital.entity.User Class Reference**Public Member Functions**

- [User](#) (int id, String name, String surname, String email, String password, int role)
- [User](#) (String name, String surname, String email, String password, int role)
- [User](#) (int id, String name, String surname, String email, String password, int role, [UserDataManipulations](#) userDataManipulations1)
- [User](#) (String name, String surname, String email, String password, int role, [UserDataManipulations](#) user↔DataManipulations1)
- int [getId](#) ()
- [Role](#) [getRole](#) ()
- void [setRole](#) ([Role](#) role)
- void [setId](#) (int id)
- String [getName](#) ()
- void [setName](#) (String name)
- String [getSurname](#) ()
- void [setSurname](#) (String surname)
- String [getEmail](#) ()
- void [setEmail](#) (String email)
- String [getPassword](#) ()
- void [setPassword](#) (String password)

Static Public Attributes

- static final String `TABLE_NAME` = "app_users"
- static final String `ID_COLUMN` = "id"
- static final String `NAME_COLUMN` = "name"
- static final String `SURNAME_COLUMN` = "surname"
- static final String `EMAIL_COLUMN` = "email"
- static final String `PASSWORD_COLUMN` = "password"
- static final String `ROLE_COLUMN` = "role"

3.40.1 Detailed Description

Class, which represents app user.

Author

Yelyzaveta Onyshchenko

Version

1.01

3.40.2 Constructor & Destructor Documentation

3.40.2.1 User() [1/4]

```
hospital.entity.User.User (  
    int id,  
    String name,  
    String surname,  
    String email,  
    String password,  
    int role )
```

Constructor for creating new object.

Parameters

<i>id</i>	corresponding user id
<i>name</i>	corresponding name
<i>surname</i>	corresponding surname
<i>email</i>	corresponding email
<i>password</i>	corresponding password
<i>role</i>	corresponding role

See also

[User\(int,String,String,String,String,int,UserDataManipulations\)](#)

3.40.2.2 User() [2/4]

```
hospital.entity.User.User (
    String name,
    String surname,
    String email,
    String password,
    int role )
```

Constructor for creating new object.

Parameters

<i>name</i>	corresponding name
<i>surname</i>	corresponding surname
<i>email</i>	corresponding email
<i>password</i>	corresponding password
<i>role</i>	corresponding role

See also

[User\(String,String,String,String,int,UserDataManipulations\)](#)

3.40.2.3 User() [3/4]

```
hospital.entity.User.User (
    int id,
    String name,
    String surname,
    String email,
    String password,
    int role,
    UserDataManipulations userDataManipulations1 )
```

Constructor for creating new object,used for testing

Parameters

<i>id</i>	corresponding user id
<i>name</i>	corresponding name
<i>surname</i>	corresponding surname
<i>email</i>	corresponding email
<i>password</i>	corresponding password
<i>role</i>	corresponding role
<i>userDataManipulations1</i>	instance for operations with data

See also

[User\(int,String,String,String,String,int\)](#)

3.40.2.4 User() [4/4]

```
hospital.entity.User.User (
    String name,
    String surname,
    String email,
    String password,
    int role,
    UserDataManipulations userDataManipulations1 )
```

Constructor for creating new object,used for testing

Parameters

<i>name</i>	corresponding name
<i>surname</i>	corresponding surname
<i>email</i>	corresponding email
<i>password</i>	corresponding password
<i>role</i>	corresponding role
<i>userDataManipulations1</i>	instance for operations with data

See also

[User\(int,String,String,String,String,int\)](#)

3.40.3 Member Function Documentation

3.40.3.1 getEmail()

```
String hospital.entity.User.getEmail ( )
```

Gets user email

Returns

user id

3.40.3.2 getId()

```
int hospital.entity.User.getId ( )
```

Gets user id

Returns

user id

3.40.3.3 getName()

```
String hospital.entity.User.getName ( )
```

Gets user name

Returns

user name

3.40.3.4 getPassword()

```
String hospital.entity.User.getPassword ( )
```

Gets user password

Returns

user id

3.40.3.5 getRole()

```
Role hospital.entity.User.getRole ( )
```

Gets user role

Returns

user role

3.40.3.6 getSurname()

```
String hospital.entity.User.getSurname ( )
```

Gets user surname

Returns

user surname

3.40.3.7 setEmail()

```
void hospital.entity.User.setEmail (
    String email )
```

Sets user email

Parameters

<i>email</i>	user email to set
--------------	-------------------

3.40.3.8 setId()

```
void hospital.entity.User.setId (  
    int id )
```

Sets user id

Parameters

<i>id</i>	user id to set
-----------	----------------

3.40.3.9 setName()

```
void hospital.entity.User.setName (  
    String name )
```

Sets user name

Parameters

<i>name</i>	user name to set
-------------	------------------

3.40.3.10 setPassword()

```
void hospital.entity.User.setPassword (  
    String password )
```

Sets user password

Parameters

<i>password</i>	user password to set
-----------------	----------------------

3.40.3.11 setRole()

```
void hospital.entity.User.setRole (
    Role role )
```

Sets user role

Parameters

<i>role</i>	user role to set
-------------	------------------

3.40.3.12 setSurname()

```
void hospital.entity.User.setSurname (
    String surname )
```

Sets user surname

Parameters

<i>surname</i>	user surname to set
----------------	---------------------

3.40.4 Member Data Documentation

3.40.4.1 EMAIL_COLUMN

```
final String hospital.entity.User.EMAIL_COLUMN = "email" [static]
```

The corresponding name of email column in table.

3.40.4.2 ID_COLUMN

```
final String hospital.entity.User.ID_COLUMN = "id" [static]
```

The corresponding name of id column in table.

3.40.4.3 NAME_COLUMN

```
final String hospital.entity.User.NAME_COLUMN = "name" [static]
```

The corresponding name of name column in table.

3.40.4.4 PASSWORD_COLUMN

```
final String hospital.entity.User.PASSWORD_COLUMN = "password" [static]
```

The corresponding name of password column in table.

3.40.4.5 ROLE_COLUMN

```
final String hospital.entity.User.ROLE_COLUMN = "role" [static]
```

The corresponding name of role column in table.

3.40.4.6 SURNAME_COLUMN

```
final String hospital.entity.User.SURNAME_COLUMN = "surname" [static]
```

The corresponding name of surname column in table.

3.40.4.7 TABLE_NAME

```
final String hospital.entity.User.TABLE_NAME = "app_users" [static]
```

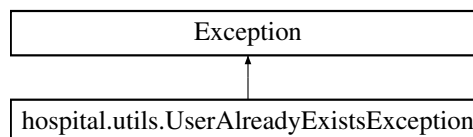
The corresponding name of table in database.

The documentation for this class was generated from the following file:

- C:/Users/Liza/Desktop/Hospital/src/main/java/hospital/entity/User.java

3.41 hospital.utils.UserAlreadyExistsException Class Reference

Inheritance diagram for hospital.utils.UserAlreadyExistsException:



Public Member Functions

- **UserAlreadyExistsException** (String message)
- **UserAlreadyExistsException** (String message, Throwable cause)
- **UserAlreadyExistsException** (Throwable cause)

3.41.1 Detailed Description

Class, which represents exception, which means that user already exists in database. Used in registration process in CreateUserServlet.

Author

Yelyzaveta Onyshchenko

Version

1.01

The documentation for this class was generated from the following file:

- C:/Users/Liza/Desktop/Hospital/src/main/java/hospital/utills/UserAlreadyExistsException.java

3.42 hospital.dao.impl.UserDAOImpl Class Reference

Public Member Functions

- List< [User](#) > [findAll](#) ()
- [User](#) [findByEmailAndPass](#) (String email, String password)
- void [insert](#) ([User](#) user)
- [User](#) [findById](#) (int id)

3.42.1 Detailed Description

Class, which represents User DAO

Author

Yelyzaveta Onyshchenko

Version

1.01

3.42.2 Member Function Documentation

3.42.2.1 findAll()

```
List<User> hospital.dao.impl.UserDAOImpl.findAll ( )
```

Find all users

Returns

list of users

3.42.2.2 findByEmailAndPass()

```
User hospital.dao.impl.UserDAOImpl.findByEmailAndPass (
    String email,
    String password )
```

Find user by email and password

Returns

corresponding user

3.42.2.3 findById()

```
User hospital.dao.impl.UserDAOImpl.findById (
    int id )
```

Find user by id

Returns

corresponding user

3.42.2.4 insert()

```
void hospital.dao.impl.UserDAOImpl.insert (
    User user )
```

Insert new user in database

The documentation for this class was generated from the following file:

- C:/Users/Liza/Desktop/Hospital/src/main/java/hospital/dao/impl/UserDAOImpl.java

3.43 hospital.utils.utilsForDBData.UserDataManipulations Class Reference

Public Member Functions

- [UserDataManipulations](#) ()
- [UserDataManipulations](#) ([DAOFactory](#) factory)
- [Role detectRole](#) (int id)

3.43.1 Detailed Description

Class, which used for operations with user data.

Author

Yelyzaveta Onyshchenko

Version

1.01

3.43.2 Constructor & Destructor Documentation

3.43.2.1 UserDataManipulations() [1/2]

```
hospital.utils.utilsForDBData.UserDataManipulations.UserDataManipulations ( )
```

Constructor for creating new object.

See also

[UserDataManipulations\(DAOFactory\)](#)

3.43.2.2 UserDataManipulations() [2/2]

```
hospital.utils.utilsForDBData.UserDataManipulations.UserDataManipulations (
    DAOFactory factory )
```

Constructor for creating new object.

See also

[UserDataManipulations\(\)](#)

3.43.3 Member Function Documentation

3.43.3.1 detectRole()

```
Role hospital.utils.utilsForDBData.UserDataManipulations.detectRole (
    int id )
```

Method which defines user role by id

Parameters

<i>id</i>	id of user
-----------	------------

Returns

corresponding instance of Role class

The documentation for this class was generated from the following file:

- C:/Users/Liza/Desktop/Hospital/src/main/java/hospital/utills/utillsForDBData/UserDataManipulations.java

3.44 hospital.utills.queries.UserQueries Interface Reference

Static Public Attributes

- static final String [FIND_BY_EMAIL_AND_PASS](#)
- static final String [FIND_ALL](#) = "select * from " + User.TABLE_NAME
- static final String [FIND_BY_ID](#) = [FIND_ALL](#) + " where " + User.ID_COLUMN + " = ?"
- static final String [INSERT_USER](#)

3.44.1 Detailed Description

Basic queries for user table.

Author

Yelyzaveta Onyshchenko

Version

1.01

3.44.2 Member Data Documentation

3.44.2.1 FIND_ALL

```
final String hospital.utills.queries.UserQueries.FIND_ALL = "select * from " + User.TABLE_NAME
[static]
```

```
SELECT * FROM app_users;
```

3.44.2.2 FIND_BY_EMAIL_AND_PASS

```
final String hospital.utils.queries.UserQueries.FIND_BY_EMAIL_AND_PASS [static]
```

Initial value:

```
= "select * from " + User.TABLE_NAME + " where "  
    + User.EMAIL_COLUMN + " = ? and " + User.PASSWORD_COLUMN + " = ?"
```

```
SELECT * FROM app_users WHERE email = ? AND password = ?;
```

3.44.2.3 FIND_BY_ID

```
final String hospital.utils.queries.UserQueries.FIND_BY_ID = FIND\_ALL + " where " + User.ID_↵  
COLUMN + " = ?" [static]
```

```
SELECT * FROM app_users WHERE id = ?;
```

3.44.2.4 INSERT_USER

```
final String hospital.utils.queries.UserQueries.INSERT_USER [static]
```

Initial value:

```
= "insert into " + User.TABLE_NAME  
    + " (" + User.NAME_COLUMN + ", " + User.SURNAME_COLUMN + ", " +  
    User.EMAIL_COLUMN + ", " + User.PASSWORD_COLUMN + ", " + User.ROLE_COLUMN + ") values (?, ?, ?,  
    ?, ?)"
```

```
INSERT INTO app_users(name,surname,email,password,role) VALUES (?, ?, ?, ?, ?);
```

The documentation for this interface was generated from the following file:

- C:/Users/Liza/Desktop/Hospital/src/main/java/hospital/utils/queries/UserQueries.java

Index

- ADD_DIAGNOSIS
 - hospital.utils.queries.PatientDataQueries, 54
- ADD_DOCTOR
 - hospital.utils.queries.PatientDataQueries, 54
- ADD_NURSE
 - hospital.utils.queries.PatientDataQueries, 54
- AuthFilter
 - hospital.servlets.filter.AuthFilter, 5, 6
- checkPrescriptionHistory
 - hospital.utils.utilsForDBData.PatientDataManipulations, 51
- CLASS_COLUMN
 - hospital.entity.prescription.Prescription, 62
- CONNECTED_TABLE_HISTORY
 - hospital.entity.prescription.Prescription, 62
- CONNECTED_TABLE_HISTORY_ID
 - hospital.entity.prescription.Prescription, 62
- CONNECTED_TABLE_HISTORY_PATIENT
 - hospital.entity.prescription.Prescription, 63
- CONNECTED_TABLE_HISTORY_PRESCR
 - hospital.entity.prescription.Prescription, 63
- CONNECTED_TABLE_NEW
 - hospital.entity.prescription.Prescription, 63
- CONNECTED_TABLE_NEW_ID
 - hospital.entity.prescription.Prescription, 63
- CONNECTED_TABLE_NEW_PATIENT
 - hospital.entity.prescription.Prescription, 63
- CONNECTED_TABLE_NEW_PRESCR
 - hospital.entity.prescription.Prescription, 63
- createCommand
 - hospital.utils.factories.CommandFactory, 8
- CreateDiagnosisCommand
 - hospital.utils.doctorcommands.CreateDiagnosisCommand, 10
- createDiagnosisDao
 - hospital.utils.factories.DAOFactory, 18
- createPatientDataDao
 - hospital.utils.factories.DAOFactory, 18
- createPatientHistory
 - hospital.utils.utilsForDBData.PatientDataManipulations, 51
- CreatePatientsCommand
 - hospital.utils.doctorcommands.CreatePatientsCommand, 12
- createPrescriptionDao
 - hospital.utils.factories.DAOFactory, 19
- CreatePrescriptionsCommand
 - hospital.utils.doctorcommands.CreatePrescriptionsCommand, 14
- createRoleDao
 - hospital.utils.factories.DAOFactory, 19
- createSingleTransaction
 - hospital.utils.factories.DAOFactory, 19
- createUserDao
 - hospital.utils.factories.DAOFactory, 19
- CreateUserServlet
 - hospital.servlets.CreateUserServlet, 16
- delete
 - hospital.dao.impl.PrescriptionDAOImpl, 65
- DELETE_PRESCRIPTION
 - hospital.utils.queries.PrescriptionQueries, 68
- destroy
 - hospital.servlets.filter.AuthFilter, 6
- detectDiagnosis
 - hospital.utils.utilsForDBData.PatientDataManipulations, 51
- detectPerson
 - hospital.utils.utilsForDBData.PatientDataManipulations, 52
- detectRole
 - hospital.utils.utilsForDBData.UserDataManipulations, 95
- Diagnosis
 - hospital.entity.diagnosis.Diagnosis, 20
- DIAGNOSIS_COLUMN
 - hospital.entity.PatientData, 45
- DIFFICULTY_COLUMN
 - hospital.entity.diagnosis.Diagnosis, 22
- DOCTOR_COLUMN
 - hospital.entity.PatientData, 46
- DoctorPageCommand
 - hospital.utils.doctorcommands.DoctorPageCommand, 25, 26
- doFilter
 - hospital.servlets.filter.AuthFilter, 6
- doGet
 - hospital.servlets.auth.LogoutServlet, 31
 - hospital.servlets.CreateUserServlet, 16
- doPost
 - hospital.servlets.CreateUserServlet, 17
 - hospital.servlets.doctorServlets.DoctorPageServlet, 27
 - hospital.servlets.nurseServlets.NursePageServlet, 38
 - hospital.servlets.patientServlets.PatientPageServlet, 58
- DoPrescriptionCommand

- hospital.utils.doctorcommands.DoPrescriptionCommand, 28, 29
- EMAIL_COLUMN
 - hospital.entity.User, 91
- encryptPasswordWithSHA256
 - hospital.utils.PasswordEncryptorSHA256, 39
- execute
 - hospital.utils.doctorcommands.CreateDiagnosisCommand, 11
 - hospital.utils.doctorcommands.CreatePatientsCommand, 13
 - hospital.utils.doctorcommands.CreatePrescriptionsCommand, 14
 - hospital.utils.doctorcommands.DoctorPageCommand, 26
 - hospital.utils.doctorcommands.DoPrescriptionCommand, 29
 - hospital.utils.doctorcommands.ReleasePatientCommand, 74
 - hospital.utils.factories.ServletCommand, 80
 - hospital.utils.nursecommands.NurseDoPrescrCommand, 33
 - hospital.utils.nursecommands.NurseFindPatientsCommand, 35
 - hospital.utils.nursecommands.NursePageCommand, 37
- FIND_ALL
 - hospital.utils.queries.DiagnosisQueries, 24
 - hospital.utils.queries.PatientDataQueries, 55
 - hospital.utils.queries.PrescriptionQueries, 68
 - hospital.utils.queries.UserQueries, 96
- FIND_ALL_HISTORY
 - hospital.utils.queries.PrescriptionQueries, 68
- FIND_ALL_MED
 - hospital.utils.queries.PrescriptionQueries, 68
- FIND_ALL_NEW
 - hospital.utils.queries.PrescriptionQueries, 68
- FIND_BY_CLASS
 - hospital.utils.queries.PrescriptionQueries, 68
- FIND_BY_EMAIL_AND_PASS
 - hospital.utils.queries.UserQueries, 96
- FIND_BY_ID
 - hospital.utils.queries.DiagnosisQueries, 24
 - hospital.utils.queries.PrescriptionQueries, 69
 - hospital.utils.queries.RoleQueries, 79
 - hospital.utils.queries.UserQueries, 97
- FIND_BY_NAME
 - hospital.utils.queries.RoleQueries, 79
- FIND_DOCTOR_PATIENTS
 - hospital.utils.queries.PatientDataQueries, 55
- FIND_MED_LIST
 - hospital.utils.queries.PrescriptionQueries, 69
- FIND_NURSE_PATIENTS
 - hospital.utils.queries.PatientDataQueries, 55
- FIND_PATIENT
 - hospital.utils.queries.PatientDataQueries, 55
- FIND_WITHOUT_DOC
 - hospital.utils.queries.PatientDataQueries, 55
 - FIND_WITHOUT_NURSE
 - hospital.utils.queries.PatientDataQueries, 55
 - findAll
 - hospital.dao.impl.DiagnosisDAOImpl, 23
 - hospital.dao.impl.PatientDataImpl, 47
 - hospital.dao.impl.PrescriptionDAOImpl, 65
 - hospital.dao.impl.UserDAOImpl, 93
 - findAllWhereNoDoc
 - hospital.dao.impl.PatientDataImpl, 47
 - findAllWhereNoNurse
 - hospital.dao.impl.PatientDataImpl, 48
 - findByActivePrescriptions
 - hospital.dao.impl.PrescriptionDAOImpl, 65
 - findByClass
 - hospital.dao.impl.PrescriptionDAOImpl, 66
 - findByEmailAndPass
 - hospital.dao.impl.UserDAOImpl, 94
 - findById
 - hospital.dao.impl.DiagnosisDAOImpl, 23
 - hospital.dao.impl.PrescriptionDAOImpl, 66
 - hospital.dao.impl.RoleDAOImpl, 78
 - hospital.dao.impl.UserDAOImpl, 94
 - findBy Name
 - hospital.dao.impl.RoleDAOImpl, 78
 - findByPatientHistory
 - hospital.dao.impl.PrescriptionDAOImpl, 66
 - findByPatientId
 - hospital.dao.impl.PatientDataImpl, 48
 - findDoctorPatients
 - hospital.dao.impl.PatientDataImpl, 48
 - findNewPrescriptions
 - hospital.utils.utilsForDBData.PatientDataManipulations, 52
 - findNursePatients
 - hospital.dao.impl.PatientDataImpl, 48
 - findPatientHistory
 - hospital.utils.utilsForDBData.PatientDataManipulations, 52
 - getAmountOfMedicine
 - hospital.entity.prescription.Prescription, 60
 - getConnection
 - hospital.utils.ConnectionPool, 9
 - getCurrentPrescriptions
 - hospital.entity.PatientData, 41
 - getDiagnosis
 - hospital.entity.PatientData, 42
 - getDifficulty
 - hospital.entity.diagnosis.Diagnosis, 21
 - getDoctor
 - hospital.entity.PatientData, 42
 - getEmail
 - hospital.entity.User, 87
 - getId
 - hospital.entity.diagnosis.Diagnosis, 21
 - hospital.entity.PatientData, 42
 - hospital.entity.prescription.Prescription, 60
 - hospital.entity.Role, 76

- hospital.entity.User, 87
- getInstance
 - hospital.utils.SingleTransaction, 81
- getName
 - hospital.entity.diagnosis.Diagnosis, 21
 - hospital.entity.prescription.Prescription, 60
 - hospital.entity.Role, 76
 - hospital.entity.User, 88
- getNurse
 - hospital.entity.PatientData, 42
- getPassword
 - hospital.entity.User, 88
- getPatient
 - hospital.entity.PatientData, 42
- getPatientStatus
 - hospital.entity.PatientData, 43
- getPrescriptionClass
 - hospital.entity.prescription.Prescription, 61
- getPrescriptionHistory
 - hospital.entity.PatientData, 43
- getRole
 - hospital.entity.User, 88
- getSurname
 - hospital.entity.User, 88
- getUrl
 - hospital.utils.URLBase, 83
- getValue
 - hospital.utils.URLBase, 84
- hospital.dao.impl.DiagnosisDAOImpl, 23
 - findAll, 23
 - findById, 23
- hospital.dao.impl.PatientDataImpl, 47
 - findAll, 47
 - findAllWhereNoDoc, 47
 - findAllWhereNoNurse, 48
 - findByPatientId, 48
 - findDoctorPatients, 48
 - findNursePatients, 48
 - insert, 49
 - updateDiagnosis, 49
 - updateDoctor, 49
 - updateNurse, 49
 - updatePatientStatus, 49
- hospital.dao.impl.PrescriptionDAOImpl, 65
 - delete, 65
 - findAll, 65
 - findByActivePrescriptions, 65
 - findByClass, 66
 - findById, 66
 - findByPatientHistory, 66
 - insertHistory, 66
 - insertNewPrescriptions, 67
- hospital.dao.impl.RoleDAOImpl, 77
 - findById, 78
 - findByName, 78
- hospital.dao.impl.UserDAOImpl, 93
 - findAll, 93
 - findByEmailAndPass, 94
 - findById, 94
 - insert, 94
- hospital.entity.diagnosis.Diagnosis, 20
 - Diagnosis, 20
 - DIFFICULTY_COLUMN, 22
 - getDifficulty, 21
 - getId, 21
 - getName, 21
 - ID_COLUMN, 22
 - NAME_COLUMN, 22
 - setDifficulty, 21
 - setId, 22
 - setName, 22
 - TABLE_NAME, 23
- hospital.entity.PatientData, 39
 - DIAGNOSIS_COLUMN, 45
 - DOCTOR_COLUMN, 46
 - getCurrentPrescriptions, 41
 - getDiagnosis, 42
 - getDoctor, 42
 - getId, 42
 - getNurse, 42
 - getPatient, 42
 - getPatientStatus, 43
 - getPrescriptionHistory, 43
 - ID_COLUMN, 46
 - NURSE_COLUMN, 46
 - PATIENT_COLUMN, 46
 - PatientData, 40, 41
 - setCurrentPrescriptions, 43
 - setDiagnosis, 43
 - setDoctor, 44
 - setId, 44
 - setNurse, 44
 - setPatient, 45
 - setPatientStatus, 45
 - setPrescriptionHistory, 45
 - STATUS_COLUMN, 46
 - TABLE_NAME, 46
- hospital.entity.prescription.Prescription, 58
 - CLASS_COLUMN, 62
 - CONNECTED_TABLE_HISTORY, 62
 - CONNECTED_TABLE_HISTORY_ID, 62
 - CONNECTED_TABLE_HISTORY_PATIENT, 63
 - CONNECTED_TABLE_HISTORY_PRESCR, 63
 - CONNECTED_TABLE_NEW, 63
 - CONNECTED_TABLE_NEW_ID, 63
 - CONNECTED_TABLE_NEW_PATIENT, 63
 - CONNECTED_TABLE_NEW_PRESCR, 63
 - getAmountOfMedicine, 60
 - getId, 60
 - getName, 60
 - getPrescriptionClass, 61
 - ID_COLUMN, 63
 - MED_TABLE, 64
 - MED_TABLE_AMOUNT, 64
 - MED_TABLE_NAME, 64
 - NAME_COLUMN, 64

- Prescription, 59, 60
- setAmountOfMedicine, 61
- setId, 61
- setName, 62
- setPrescriptionClass, 62
- TABLE_NAME, 64
- hospital.entity.Role, 75
 - getId, 76
 - getName, 76
 - ID_COLUMN, 77
 - NAME_COLUMN, 77
 - Role, 75
 - setId, 76
 - setName, 76
 - TABLE_NAME, 77
- hospital.entity.User, 84
 - EMAIL_COLUMN, 91
 - getEmail, 87
 - getId, 87
 - getName, 88
 - getPassword, 88
 - getRole, 88
 - getSurname, 88
 - ID_COLUMN, 91
 - NAME_COLUMN, 91
 - PASSWORD_COLUMN, 91
 - ROLE_COLUMN, 92
 - setEmail, 89
 - setId, 90
 - setName, 90
 - setPassword, 90
 - setRole, 90
 - setSurname, 91
 - SURNAME_COLUMN, 92
 - TABLE_NAME, 92
 - User, 85–87
- hospital.servlets.auth.LogoutServlet, 31
 - doGet, 31
- hospital.servlets.CreateUserServlet, 15
 - CreateUserServlet, 16
 - doGet, 16
 - doPost, 17
 - inputValidator, 17
- hospital.servlets.doctorServlets.DoctorPageServlet, 27
 - doPost, 27
- hospital.servlets.filter.AuthFilter, 5
 - AuthFilter, 5, 6
 - destroy, 6
 - doFilter, 6
 - init, 7
- hospital.servlets.nurseServlets.NursePageServlet, 37
 - doPost, 38
- hospital.servlets.patientServlets.PatientPageServlet, 56
 - doPost, 58
 - PatientPageServlet, 57
- hospital.utils.ConnectionPool, 8
 - getConnection, 9
 - PASSWORD, 9
 - URL, 9
 - USER, 9
- hospital.utils.doctorcommands.CreateDiagnosisCommand, 10
 - CreateDiagnosisCommand, 10
 - execute, 11
- hospital.utils.doctorcommands.CreatePatientsCommand, 11
 - CreatePatientsCommand, 12
 - execute, 13
- hospital.utils.doctorcommands.CreatePrescriptionsCommand, 13
 - CreatePrescriptionsCommand, 14
 - execute, 14
- hospital.utils.doctorcommands.DoctorPageCommand, 25
 - DoctorPageCommand, 25, 26
 - execute, 26
- hospital.utils.doctorcommands.DoPrescriptionCommand, 28
 - DoPrescriptionCommand, 28, 29
 - execute, 29
- hospital.utils.doctorcommands.ReleasePatientCommand, 73
 - execute, 74
 - ReleasePatientCommand, 74
- hospital.utils.factories.CommandFactory, 7
 - createCommand, 8
- hospital.utils.factories.DAOFactory, 18
 - createDiagnosisDao, 18
 - createPatientDataDao, 18
 - createPrescriptionDao, 19
 - createRoleDao, 19
 - createSingleTransaction, 19
 - createUserDao, 19
- hospital.utils.factories.ServletCommand, 80
 - execute, 80
- hospital.utils.InputDataCheck, 30
 - inputCheck, 30
- hospital.utils.nursecommands.NurseDoPrescrCommand, 32
 - execute, 33
 - NurseDoPrescrCommand, 32, 33
- hospital.utils.nursecommands.NurseFindPatientsCommand, 34
 - execute, 35
 - NurseFindPatientsCommand, 34
- hospital.utils.nursecommands.NursePageCommand, 35
 - execute, 37
 - NursePageCommand, 36
- hospital.utils.PasswordEncryptorSHA256, 38
 - encryptPasswordWithSHA256, 39
- hospital.utils.queries.DiagnosisQueries, 24
 - FIND_ALL, 24
 - FIND_BY_ID, 24
- hospital.utils.queries.PatientDataQueries, 54
 - ADD_DIAGNOSIS, 54
 - ADD_DOCTOR, 54

- ADD_NURSE, [54](#)
- FIND_ALL, [55](#)
- FIND_DOCTOR_PATIENTS, [55](#)
- FIND_NURSE_PATIENTS, [55](#)
- FIND_PATIENT, [55](#)
- FIND_WITHOUT_DOC, [55](#)
- FIND_WITHOUT_NURSE, [55](#)
- INSERT_PATIENT, [56](#)
- UPDATE_PATIENT_STATUS, [56](#)
- hospital.utils.queries.PrescriptionQueries, [67](#)
 - DELETE_PRESCRIPTION, [68](#)
 - FIND_ALL, [68](#)
 - FIND_ALL_HISTORY, [68](#)
 - FIND_ALL_MED, [68](#)
 - FIND_ALL_NEW, [68](#)
 - FIND_BY_CLASS, [68](#)
 - FIND_BY_ID, [69](#)
 - FIND_MED_LIST, [69](#)
 - INSERT_NEW_PRESCRIPTION, [69](#)
 - INSERT_PRESCRIPTION_HISTORY, [69](#)
 - UPDATE_MED_LIST, [69](#)
- hospital.utils.queries.RoleQueries, [79](#)
 - FIND_BY_ID, [79](#)
 - FIND_BY_NAME, [79](#)
- hospital.utils.queries.UserQueries, [96](#)
 - FIND_ALL, [96](#)
 - FIND_BY_EMAIL_AND_PASS, [96](#)
 - FIND_BY_ID, [97](#)
 - INSERT_USER, [97](#)
- hospital.utils.RandomNumber, [70](#)
 - randNumber, [70](#)
- hospital.utils.RandomPatient, [70](#)
 - RandomPatient, [71](#)
 - randomPatientDoc, [71](#)
 - randomPatientNurse, [72](#)
- hospital.utils.RegexContainer, [72](#)
 - REGEX_EMAIL, [72](#)
 - REGEX_NAME_ENG, [73](#)
 - REGEX_PASSWORD, [73](#)
- hospital.utils.SingleTransaction, [81](#)
 - getInstance, [81](#)
 - transfer, [82](#)
- hospital.utils.URLBase, [82](#)
 - getUrl, [83](#)
 - getValue, [84](#)
 - URLBase, [83](#)
- hospital.utils.UserAlreadyExistsException, [92](#)
- hospital.utils.utilsForDBData.PatientDataManipulations, [50](#)
 - checkPrescriptionHistory, [51](#)
 - createPatientHistory, [51](#)
 - detectDiagnosis, [51](#)
 - detectPerson, [52](#)
 - findNewPrescriptions, [52](#)
 - findPatientHistory, [52](#)
 - makePrescriptions, [53](#)
 - PatientDataManipulations, [50, 51](#)
 - prepareActivePrescriptions, [53](#)
 - refreshPatients, [53](#)
- hospital.utils.utilsForDBData.UserDataManipulations, [95](#)
 - detectRole, [95](#)
 - UserDataManipulations, [95](#)
- ID_COLUMN
 - hospital.entity.diagnosis.Diagnosis, [22](#)
 - hospital.entity.PatientData, [46](#)
 - hospital.entity.prescription.Prescription, [63](#)
 - hospital.entity.Role, [77](#)
 - hospital.entity.User, [91](#)
- init
 - hospital.servlets.filter.AuthFilter, [7](#)
- inputCheck
 - hospital.utils.InputDataCheck, [30](#)
- inputValidator
 - hospital.servlets.CreateUserServlet, [17](#)
- insert
 - hospital.dao.impl.PatientDataImpl, [49](#)
 - hospital.dao.impl.UserDAOImpl, [94](#)
- INSERT_NEW_PRESCRIPTION
 - hospital.utils.queries.PrescriptionQueries, [69](#)
- INSERT_PATIENT
 - hospital.utils.queries.PatientDataQueries, [56](#)
- INSERT_PRESCRIPTION_HISTORY
 - hospital.utils.queries.PrescriptionQueries, [69](#)
- INSERT_USER
 - hospital.utils.queries.UserQueries, [97](#)
- insertHistory
 - hospital.dao.impl.PrescriptionDAOImpl, [66](#)
- insertNewPrescriptions
 - hospital.dao.impl.PrescriptionDAOImpl, [67](#)
- makePrescriptions
 - hospital.utils.utilsForDBData.PatientDataManipulations, [53](#)
- MED_TABLE
 - hospital.entity.prescription.Prescription, [64](#)
- MED_TABLE_AMOUNT
 - hospital.entity.prescription.Prescription, [64](#)
- MED_TABLE_NAME
 - hospital.entity.prescription.Prescription, [64](#)
- NAME_COLUMN
 - hospital.entity.diagnosis.Diagnosis, [22](#)
 - hospital.entity.prescription.Prescription, [64](#)
 - hospital.entity.Role, [77](#)
 - hospital.entity.User, [91](#)
- NURSE_COLUMN
 - hospital.entity.PatientData, [46](#)
- NurseDoPrescrCommand
 - hospital.utils.nursecommands.NurseDoPrescrCommand, [32, 33](#)
- NurseFindPatientsCommand
 - hospital.utils.nursecommands.NurseFindPatientsCommand, [34](#)
- NursePageCommand

- hospital.utils.nursecommands.NursePageCommand, 36
- PASSWORD
 - hospital.utils.ConnectionPool, 9
- PASSWORD_COLUMN
 - hospital.entity.User, 91
- PATIENT_COLUMN
 - hospital.entity.PatientData, 46
- PatientData
 - hospital.entity.PatientData, 40, 41
- PatientDataManipulations
 - hospital.utils.utilsForDBData.PatientDataManipulations, 50, 51
- PatientPageServlet
 - hospital.servlets.patientServlets.PatientPageServlet, 57
- prepareActivePrescriptions
 - hospital.utils.utilsForDBData.PatientDataManipulations, 53
- Prescription
 - hospital.entity.prescription.Prescription, 59, 60
- randNumber
 - hospital.utils.RandomNumber, 70
- RandomPatient
 - hospital.utils.RandomPatient, 71
- randomPatientDoc
 - hospital.utils.RandomPatient, 71
- randomPatientNurse
 - hospital.utils.RandomPatient, 72
- refreshPatients
 - hospital.utils.utilsForDBData.PatientDataManipulations, 53
- REGEX_EMAIL
 - hospital.utils.RegexContainer, 72
- REGEX_NAME_ENG
 - hospital.utils.RegexContainer, 73
- REGEX_PASSWORD
 - hospital.utils.RegexContainer, 73
- ReleasePatientCommand
 - hospital.utils.doctorcommands.ReleasePatientCommand, 74
- Role
 - hospital.entity.Role, 75
- ROLE_COLUMN
 - hospital.entity.User, 92
- setAmountOfMedicine
 - hospital.entity.prescription.Prescription, 61
- setCurrentPrescriptions
 - hospital.entity.PatientData, 43
- setDiagnosis
 - hospital.entity.PatientData, 43
- setDifficulty
 - hospital.entity.diagnosis.Diagnosis, 21
- setDoctor
 - hospital.entity.PatientData, 44
- setEmail
 - hospital.entity.User, 89
- setId
 - hospital.entity.diagnosis.Diagnosis, 22
 - hospital.entity.PatientData, 44
 - hospital.entity.prescription.Prescription, 61
 - hospital.entity.Role, 76
 - hospital.entity.User, 90
- setName
 - hospital.entity.diagnosis.Diagnosis, 22
 - hospital.entity.prescription.Prescription, 62
 - hospital.entity.Role, 76
 - hospital.entity.User, 90
- setNurse
 - hospital.entity.PatientData, 44
- setPassword
 - hospital.entity.User, 90
- setPatient
 - hospital.entity.PatientData, 45
- setPatientStatus
 - hospital.entity.PatientData, 45
- setPrescriptionClass
 - hospital.entity.prescription.Prescription, 62
- setPrescriptionHistory
 - hospital.entity.PatientData, 45
- setRole
 - hospital.entity.User, 90
- setSurname
 - hospital.entity.User, 91
- STATUS_COLUMN
 - hospital.entity.PatientData, 46
- SURNAME_COLUMN
 - hospital.entity.User, 92
- TABLE_NAME
 - hospital.entity.diagnosis.Diagnosis, 23
 - hospital.entity.PatientData, 46
 - hospital.entity.prescription.Prescription, 64
 - hospital.entity.Role, 77
 - hospital.entity.User, 92
- transfer
 - hospital.utils.SingleTransaction, 82
- UPDATE_MED_LIST
 - hospital.utils.queries.PrescriptionQueries, 69
- UPDATE_PATIENT_STATUS
 - hospital.utils.queries.PatientDataQueries, 56
- updateDiagnosis
 - hospital.dao.impl.PatientDataImpl, 49
- updateDoctor
 - hospital.dao.impl.PatientDataImpl, 49
- updateNurse
 - hospital.dao.impl.PatientDataImpl, 49
- updatePatientStatus
 - hospital.dao.impl.PatientDataImpl, 49
- URL
 - hospital.utils.ConnectionPool, 9
- URLBase
 - hospital.utils.URLBase, 83
- USER

- [hospital.utils.ConnectionPool](#), [9](#)
- User
 - [hospital.entity.User](#), [85–87](#)
- UserDataManipulations
 - [hospital.utils.utilsForDBData.UserDataManipulations](#),
[95](#)