

iLearnRW

0.0.1

Generated by Doxygen 1.8.4

Wed Jun 19 2013 13:32:20



# Contents

<b>1</b>	<b>Main Page</b>	<b>1</b>
<b>2</b>	<b>Log format</b>	<b>3</b>
<b>3</b>	<b>About</b>	<b>5</b>
<b>4</b>	<b>Todo List</b>	<b>7</b>
<b>5</b>	<b>Namespace Documentation</b>	<b>9</b>
5.1	Package ilearnrw . . . . .	9
5.2	Package ilearnrw.datalogger . . . . .	9
5.2.1	Detailed Description . . . . .	9
5.3	Package ilearnrw.prototype . . . . .	10
5.4	Package ilearnrw.prototype.application . . . . .	10
5.5	Package ilearnrw.user . . . . .	10
<b>6</b>	<b>Class Documentation</b>	<b>11</b>
6.1	ilearnrw.user.Category Class Reference . . . . .	11
6.1.1	Constructor & Destructor Documentation . . . . .	12
6.1.1.1	Category . . . . .	12
6.1.2	Member Function Documentation . . . . .	13
6.1.2.1	equals . . . . .	13
6.1.2.2	getTitle . . . . .	13
6.1.2.3	isReadingProblem . . . . .	13
6.1.2.4	setReadingProblem . . . . .	13
6.1.2.5	setTitle . . . . .	13
6.1.2.6	toString . . . . .	13
6.1.3	Member Data Documentation . . . . .	13
6.1.3.1	isReadingProblem . . . . .	13
6.1.3.2	serialVersionUID . . . . .	13
6.1.3.3	title . . . . .	13
6.2	ilearnrw.prototype.application.ConsoleMenu Class Reference . . . . .	13
6.2.1	Constructor & Destructor Documentation . . . . .	14

6.2.1.1	ConsoleMenu	14
6.2.2	Member Function Documentation	14
6.2.2.1	doModalMenu	14
6.2.2.2	in	14
6.2.2.3	out	14
6.2.2.4	showMenu	14
6.2.2.5	subMenu	14
6.2.3	Member Data Documentation	14
6.2.3.1	mIn	14
6.2.3.2	mMenuActions	14
6.2.3.3	mMenuTitle	14
6.2.3.4	mOut	14
6.2.3.5	mScanner	14
6.3	ilearnrw.prototype.application.ConsoleMenuAction Class Reference	15
6.3.1	Constructor & Destructor Documentation	16
6.3.1.1	ConsoleMenuAction	16
6.3.2	Member Function Documentation	16
6.3.2.1	getText	16
6.3.2.2	onSelected	16
6.3.3	Member Data Documentation	16
6.3.3.1	mText	16
6.4	ilearnrw.prototype.application.DatabaseManager Class Reference	17
6.4.1	Constructor & Destructor Documentation	18
6.4.1.1	DatabaseManager	18
6.4.2	Member Function Documentation	18
6.4.2.1	onSelected	19
6.5	ilearnrw.datalogger.DataLogger Class Reference	19
6.5.1	Detailed Description	20
6.5.2	Member Function Documentation	20
6.5.2.1	getUserStore	20
6.5.2.2	loadUserStore	20
6.5.3	Member Data Documentation	20
6.5.3.1	mUserStore	20
6.6	ilearnrw.prototype.application.ConsoleMenu.EConsoleMenuActionResult Enum Reference	20
6.6.1	Member Data Documentation	21
6.6.1.1	menuBreak	21
6.6.1.2	showThisMenuAgain	21
6.7	ilearnrw.user.GameAction Enum Reference	21
6.7.1	Member Data Documentation	21
6.7.1.1	DIFFICULT_WORD	21

6.8	<a href="#">ilearnrw.prototype.application.ConsoleMenu.IConsoleMenuAction Interface Reference</a>	22
6.8.1	Member Function Documentation	22
6.8.1.1	getText	22
6.8.1.2	onSelected	23
6.9	<a href="#">ilearnrw.datalogger.IDataLogger Interface Reference</a>	23
6.9.1	Member Function Documentation	23
6.9.1.1	getLoginProvider	23
6.9.1.2	getProfileAccessUpdater	23
6.10	<a href="#">ilearnrw.user.IlearnException Class Reference</a>	24
6.10.1	Constructor & Destructor Documentation	25
6.10.1.1	IlearnException	25
6.10.1.2	IlearnException	25
6.10.2	Member Function Documentation	25
6.10.2.1	getMessage	25
6.11	<a href="#">ilearnrw.datalogger.ILoginProvider Interface Reference</a>	25
6.11.1	Member Function Documentation	26
6.11.1.1	getUser	26
6.12	<a href="#">ilearnrw.datalogger.IProfileAccessUpdater Interface Reference</a>	26
6.12.1	Detailed Description	29
6.12.2	Member Function Documentation	29
6.12.2.1	addProblemDefinition	29
6.12.2.2	discardPendingChanges	29
6.12.2.3	hasPendingChanges	29
6.12.2.4	registerUserSession	29
6.12.2.5	removeProblemDefinition	29
6.12.2.6	setCurrentProfile	29
6.12.2.7	setPreference	30
6.12.2.8	setUserDetail	30
6.12.2.9	setUserLanguage	30
6.12.2.10	writePendingChanges	30
6.13	<a href="#">ilearnrw.prototype.application.DatabaseManager.ISelectUser Interface Reference</a>	31
6.13.1	Member Function Documentation	31
6.13.1.1	selectUser	31
6.14	<a href="#">ilearnrw.user.LanguageCode Enum Reference</a>	31
6.14.1	Member Function Documentation	32
6.14.1.1	getEnglishCode	32
6.14.1.2	getGreekCode	32
6.14.2	Member Data Documentation	32
6.14.2.1	EN	32
6.15	<a href="#">ilearnrw.datalogger.LoginDetails Class Reference</a>	32

6.15.1	Member Data Documentation . . . . .	32
6.15.1.1	Password . . . . .	32
6.15.1.2	Username . . . . .	32
6.16	ilearnrw.datalogger.IProfileAccessUpdater.PendingChangesAvailable Class Reference . . . . .	32
6.16.1	Detailed Description . . . . .	34
6.16.2	Constructor & Destructor Documentation . . . . .	34
6.16.2.1	PendingChangesAvailable . . . . .	34
6.16.2.2	PendingChangesAvailable . . . . .	34
6.16.3	Member Function Documentation . . . . .	34
6.16.3.1	getMessage . . . . .	34
6.16.4	Member Data Documentation . . . . .	34
6.16.4.1	serialVersionUID . . . . .	34
6.17	ilearnrw.datalogger.IProfileAccessUpdater.PermissionException Class Reference . . . . .	34
6.17.1	Detailed Description . . . . .	35
6.17.2	Constructor & Destructor Documentation . . . . .	35
6.17.2.1	PermissionException . . . . .	35
6.17.2.2	PermissionException . . . . .	35
6.17.3	Member Function Documentation . . . . .	35
6.17.3.1	getMessage . . . . .	35
6.17.4	Member Data Documentation . . . . .	35
6.17.4.1	serialVersionUID . . . . .	36
6.18	ilearnrw.user.ProblemDefinition Class Reference . . . . .	36
6.18.1	Constructor & Destructor Documentation . . . . .	38
6.18.1.1	ProblemDefinition . . . . .	38
6.18.2	Member Function Documentation . . . . .	38
6.18.2.1	getAvailableLanguages . . . . .	38
6.18.2.2	getScoreUpperBound . . . . .	38
6.18.2.3	getType . . . . .	38
6.18.2.4	getURI . . . . .	38
6.18.2.5	setAvailableLanguages . . . . .	38
6.18.2.6	setScoreUpperBound . . . . .	38
6.18.2.7	setSubcategory . . . . .	38
6.18.2.8	setURI . . . . .	38
6.18.2.9	toString . . . . .	38
6.18.3	Member Data Documentation . . . . .	38
6.18.3.1	availableLanguages . . . . .	38
6.18.3.2	scoreUpperBound . . . . .	38
6.18.3.3	serialVersionUID . . . . .	38
6.18.3.4	type . . . . .	38
6.18.3.5	URI . . . . .	38

6.19	<a href="#">ilearnrw.user.ProblemDefinitionIndex Class Reference</a>	39
6.19.1	Member Function Documentation	39
6.19.1.1	<a href="#">getProblemsByCategory</a>	39
6.19.1.2	<a href="#">getProblemsByLanguage</a>	39
6.19.2	Member Data Documentation	39
6.19.2.1	<a href="#">index</a>	39
6.20	<a href="#">ilearnrw.datalogger.ProfileAccessUpdater Class Reference</a>	40
6.20.1	Detailed Description	42
6.20.2	Member Function Documentation	42
6.20.2.1	<a href="#">addProblemDefinition</a>	42
6.20.2.2	<a href="#">discardPendingChanges</a>	42
6.20.2.3	<a href="#">hasPendingChanges</a>	42
6.20.2.4	<a href="#">registerUserSession</a>	43
6.20.2.5	<a href="#">removeProblemDefinition</a>	43
6.20.2.6	<a href="#">setCurrentProfile</a>	43
6.20.2.7	<a href="#">setDirty</a>	43
6.20.2.8	<a href="#">setPreference</a>	43
6.20.2.9	<a href="#">setUserDetail</a>	43
6.20.2.10	<a href="#">setUserLanguage</a>	43
6.20.2.11	<a href="#">writePendingChanges</a>	43
6.20.3	Member Data Documentation	44
6.20.3.1	<a href="#">mIsDirty</a>	44
6.20.3.2	<a href="#">mMarker</a>	44
6.20.3.3	<a href="#">mUser</a>	44
6.21	<a href="#">ilearnrw.prototype.application.Program Class Reference</a>	45
6.21.1	Member Function Documentation	46
6.21.1.1	<a href="#">getLoginDetails</a>	46
6.21.1.2	<a href="#">getStringArg</a>	46
6.21.1.3	<a href="#">main</a>	46
6.21.2	Member Data Documentation	46
6.21.2.1	<a href="#">gDataLogger</a>	46
6.21.2.2	<a href="#">sWelcomeMessage</a>	46
6.22	<a href="#">ilearnrw.user.Subcategory Class Reference</a>	47
6.22.1	Constructor & Destructor Documentation	49
6.22.1.1	<a href="#">Subcategory</a>	49
6.22.2	Member Function Documentation	49
6.22.2.1	<a href="#">equals</a>	49
6.22.2.2	<a href="#">getCategory</a>	49
6.22.2.3	<a href="#">getSubcategoryTitle</a>	49
6.22.2.4	<a href="#">setCategory</a>	49

6.22.2.5	setSubcategoryTitle	49
6.22.2.6	toString	49
6.22.3	Member Data Documentation	49
6.22.3.1	category	49
6.22.3.2	serialVersionUID	49
6.22.3.3	subcategoryTitle	49
6.23	ilearnrw.user.Tester Class Reference	49
6.23.1	Member Function Documentation	50
6.23.1.1	main	50
6.24	ilearnrw.user.User Class Reference	50
6.24.1	Constructor & Destructor Documentation	52
6.24.1.1	User	52
6.24.1.2	User	52
6.24.1.3	User	52
6.24.2	Member Function Documentation	52
6.24.2.1	getDetails	52
6.24.2.2	getOldSessions	52
6.24.2.3	getProfile	52
6.24.2.4	getSession	52
6.24.2.5	getUserId	52
6.24.2.6	setDetails	52
6.24.2.7	setOldSessions	52
6.24.2.8	setProfile	52
6.24.2.9	setSession	52
6.24.3	Member Data Documentation	52
6.24.3.1	details	52
6.24.3.2	oldSessions	52
6.24.3.3	profile	52
6.24.3.4	serialVersionUID	52
6.24.3.5	session	52
6.24.3.6	userId	52
6.25	ilearnrw.user.UserAction Class Reference	53
6.25.1	Member Function Documentation	53
6.25.1.1	getGameAction	53
6.25.1.2	getUserId	53
6.25.1.3	setGameAction	53
6.25.1.4	setUserId	53
6.25.2	Member Data Documentation	53
6.25.2.1	gameAction	53
6.25.2.2	UserId	54



6.26	ilearnrw.user.UserDetails Class Reference	54
6.26.1	Constructor & Destructor Documentation	56
6.26.1.1	UserDetails	56
6.26.1.2	UserDetails	56
6.26.1.3	UserDetails	56
6.26.2	Member Function Documentation	56
6.26.2.1	checkPassword	56
6.26.2.2	getLanguage	56
6.26.2.3	getUsername	56
6.26.2.4	hash	56
6.26.2.5	hasPassword	56
6.26.2.6	passwordIsSet	56
6.26.2.7	setLanguage	56
6.26.2.8	setLanguage	56
6.26.2.9	setPassword	57
6.26.2.10	setPasswordHash	57
6.26.2.11	setUsername	57
6.26.3	Member Data Documentation	57
6.26.3.1	desiredKeyLen	57
6.26.3.2	iterations	57
6.26.3.3	language	57
6.26.3.4	mPasswordHash	57
6.26.3.5	mPasswordSalt	57
6.26.3.6	saltLen	57
6.26.3.7	serialVersionUID	57
6.26.3.8	username	57
6.27	ilearnrw.prototype.application.UserLogin Class Reference	57
6.27.1	Detailed Description	59
6.27.2	Constructor & Destructor Documentation	59
6.27.2.1	UserLogin	59
6.27.3	Member Function Documentation	59
6.27.3.1	onSelected	60
6.28	ilearnrw.user.UserPreferences Class Reference	60
6.28.1	Constructor & Destructor Documentation	61
6.28.1.1	UserPreferences	61
6.28.1.2	UserPreferences	61
6.28.2	Member Function Documentation	61
6.28.2.1	getFontSize	61
6.28.2.2	setFontSize	61
6.28.3	Member Data Documentation	62

6.28.3.1	fontSize	62
6.28.3.2	serialVersionUID	62
6.29	ilearnrw.user.UserProblem Class Reference	62
6.29.1	Constructor & Destructor Documentation	63
6.29.1.1	UserProblem	63
6.29.1.2	UserProblem	63
6.29.2	Member Function Documentation	63
6.29.2.1	getProblem	63
6.29.2.2	getScore	63
6.29.2.3	setProblem	63
6.29.2.4	setScore	63
6.29.3	Member Data Documentation	63
6.29.3.1	problem	63
6.29.3.2	score	63
6.30	ilearnrw.prototype.application.UserProblems Class Reference	64
6.30.1	Constructor & Destructor Documentation	65
6.30.1.1	UserProblems	65
6.30.2	Member Function Documentation	65
6.30.2.1	onSelected	66
6.30.3	Member Data Documentation	66
6.30.3.1	mUser	66
6.31	ilearnrw.user.UserProblemsList Class Reference	66
6.31.1	Constructor & Destructor Documentation	67
6.31.1.1	UserProblemsList	67
6.31.1.2	UserProblemsList	67
6.31.1.3	UserProblemsList	68
6.31.2	Member Function Documentation	68
6.31.2.1	getList	68
6.31.2.2	loadProblems	68
6.31.2.3	loadProblems	68
6.31.2.4	setList	68
6.31.3	Member Data Documentation	68
6.31.3.1	list	68
6.31.3.2	serialVersionUID	68
6.32	ilearnrw.user.UserProfile Class Reference	68
6.32.1	Constructor & Destructor Documentation	70
6.32.1.1	UserProfile	70
6.32.1.2	UserProfile	70
6.32.1.3	UserProfile	70
6.32.2	Member Function Documentation	70

6.32.2.1	getLanguage	70
6.32.2.2	getPreferences	70
6.32.2.3	getProblemsList	70
6.32.2.4	setLanguage	70
6.32.2.5	setPreferences	70
6.32.2.6	setProblemsList	70
6.32.3	Member Data Documentation	70
6.32.3.1	language	70
6.32.3.2	preferences	70
6.32.3.3	problems	70
6.32.3.4	serialVersionUID	70
6.33	ilearnrw.user.UserSession Class Reference	71
6.33.1	Member Function Documentation	71
6.33.1.1	insertAction	71
6.33.1.2	loadSession	71
6.33.1.3	startSession	71
6.33.1.4	stopSession	72
6.33.2	Member Data Documentation	72
6.33.2.1	end	72
6.33.2.2	history	72
6.33.2.3	isActive	72
6.33.2.4	start	72
6.34	ilearnrw.datalogger.UserStore Class Reference	72
6.34.1	Detailed Description	74
6.34.2	Constructor & Destructor Documentation	74
6.34.2.1	UserStore	74
6.34.3	Member Function Documentation	74
6.34.3.1	create	74
6.34.3.2	deepCopy	74
6.34.3.3	delete	74
6.34.3.4	getUser	75
6.34.3.5	getUserIds	75
6.34.3.6	load	75
6.34.3.7	read	75
6.34.3.8	read_noCopy	75
6.34.3.9	save	75
6.34.3.10	update	75
6.34.4	Member Data Documentation	76
6.34.4.1	mFilePath	76
6.34.4.2	mLoadedUsers	76

6.34.4.3 mLogger . . . . .	76
----------------------------	----

<b>Index</b>	<b>77</b>
--------------	-----------

## Chapter 1

# Main Page

iLearnRW developer repository



## Chapter 2

# Log format

### Actions

- Add difficult word
- Remove difficult word

Profile	Marker	TimeStamp	Action	Value
child1	teacher1	2013-06-14	ADD	cat
child1	child1	2013-06-14	REMOVE	cat





## Chapter 3

# About

This document was generated using Doxygen from a number of (mainly) stub classes.

The documentation is published in the iLearnRW git repository. Currently located in the `playground/Data-Logger` directory.

To be able to build the documentation the following is required:

- Doxygen
- Latex
- Graphviz



## Chapter 4

# Todo List

### **Class** `ilearnrw.datalogger.UserStore`

Replace storage backed with SQL database.

### **Member** `ilearnrw.datalogger.UserStore.update (User user)`

Don't allow multiple users with the same username.

### **Member** `ilearnrw.user.UserDetails.username`

Something to think about is whether this should be case-sensitive and if unicode should be supported. Unicode can create some headaches, there was a blog post about this recently (of which i only read the headline) see:

<http://labs.spotify.com/2013/06/18/creative-usernames/>



## Chapter 5

# Namespace Documentation

### 5.1 Package ilearnrw

#### Packages

- package [datalogger](#)  
*[DataLogger](#) package.*
- package [prototype](#)
- package [user](#)

### 5.2 Package ilearnrw.datalogger

[DataLogger](#) package.

#### Classes

- class [DataLogger](#)
- interface [IDataLogger](#)
- interface [ILoginProvider](#)
- interface [IProfileAccessUpdater](#)  
*Interface to update profiles in the iLearnRW system.*
- class [LoginDetails](#)
- class [ProfileAccessUpdater](#)
- class [UserStore](#)  
*The [UserStore](#) is used to create and store user objects persistantly.*

#### 5.2.1 Detailed Description

[DataLogger](#) package.

#### Note

Exactly what package the [IProfileAccessUpdater](#) should be part of is still unknown.

## 5.3 Package ilearnrw.prototype

### Packages

- package [application](#)

## 5.4 Package ilearnrw.prototype.application

### Classes

- class [ConsoleMenu](#)
- class [ConsoleMenuAction](#)
- class [DatabaseManager](#)
- class [Program](#)
- class [UserLogin](#)
  - Main menu of a logged in user.*
- class [UserProblems](#)

## 5.5 Package ilearnrw.user

### Classes

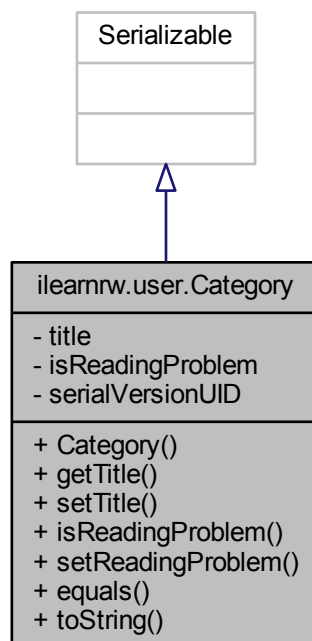
- class [Category](#)
- enum [GameAction](#)
- class [IlearnException](#)
- enum [LanguageCode](#)
- class [ProblemDefinition](#)
- class [ProblemDefinitionIndex](#)
- class [Subcategory](#)
- class [Tester](#)
- class [User](#)
- class [UserAction](#)
- class [UserDetails](#)
- class [UserPreferences](#)
- class [UserProblem](#)
- class [UserProblemsList](#)
- class [UserProfile](#)
- class [UserSession](#)

## Chapter 6

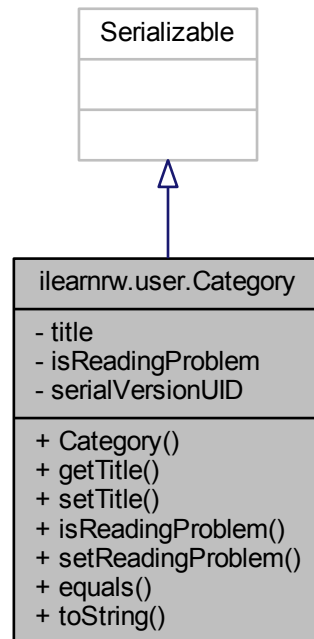
# Class Documentation

### 6.1 ilearnrw.user.Category Class Reference

Inheritance diagram for ilearnrw.user.Category:



Collaboration diagram for `ilearnrw.user.Category`:



## Public Member Functions

- `Category` (`String title`, `boolean isReadingProblem`)
- `String getTitle` ()
- `void setTitle` (`String title`)
- `boolean isReadingProblem` ()
- `void setReadingProblem` (`boolean isReadingProblem`)
- `boolean equals` (`Object obj`)
- `String toString` ()

## Private Attributes

- `String title`
- `boolean isReadingProblem`

## Static Private Attributes

- `static final long serialVersionUID` = 1L

## 6.1.1 Constructor & Destructor Documentation

### 6.1.1.1 `ilearnrw.user.Category.Category` ( `String title`, `boolean isReadingProblem` )



### 6.1.2 Member Function Documentation

6.1.2.1 boolean ilearnrw.user.Category.equals ( Object *obj* )

6.1.2.2 String ilearnrw.user.Category.getTitle ( )

6.1.2.3 boolean ilearnrw.user.Category.isReadingProblem ( )

6.1.2.4 void ilearnrw.user.Category.setReadingProblem ( boolean *isReadingProblem* )

6.1.2.5 void ilearnrw.user.Category.setTitle ( String *title* )

6.1.2.6 String ilearnrw.user.Category.toString ( )

### 6.1.3 Member Data Documentation

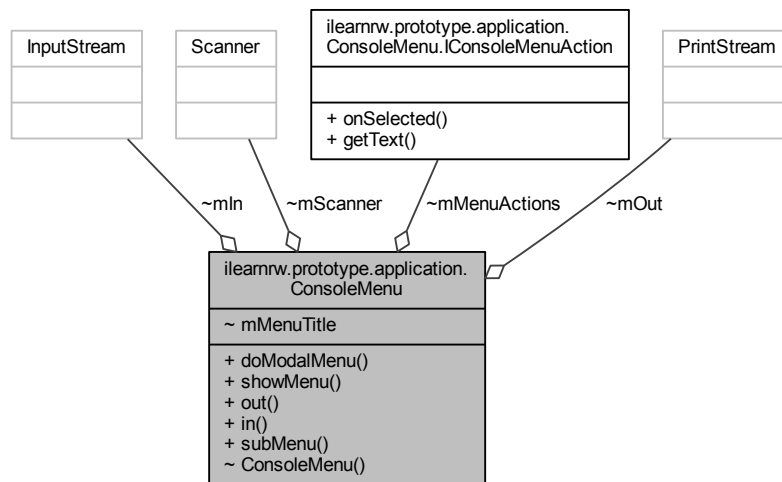
6.1.3.1 boolean ilearnrw.user.Category.isReadingProblem [private]

6.1.3.2 final long ilearnrw.user.Category.serialVersionUID = 1L [static],[private]

6.1.3.3 String ilearnrw.user.Category.title [private]

## 6.2 ilearnrw.prototype.application.ConsoleMenu Class Reference

Collaboration diagram for ilearnrw.prototype.application.ConsoleMenu:



### Classes

- enum [EConsoleMenuActionResult](#)
- interface [IConsoleMenuAction](#)

### Public Member Functions

- void [doModalMenu](#) ()

- [EConsoleMenuActionResult](#) `showMenu ()`
- `PrintStream` `out ()`
- `Scanner` `in ()`
- [ConsoleMenu](#) `subMenu (String menuTitle, IConsoleMenuAction[] menuActions)`

## Package Functions

- [ConsoleMenu](#) (`PrintStream outStream, InputStream inStream, String menuTitle, IConsoleMenuAction[] menuActions`)

## Package Attributes

- `PrintStream` `mOut`
- `InputStream` `mIn`
- `Scanner` `mScanner`
- `String` `mMenuTitle`
- [IConsoleMenuAction](#)[] `mMenuActions`

## 6.2.1 Constructor & Destructor Documentation

6.2.1.1 `ilearnrw.prototype.application.ConsoleMenu.ConsoleMenu ( PrintStream outStream, InputStream inStream, String menuTitle, IConsoleMenuAction[] menuActions )` [`package`]

## 6.2.2 Member Function Documentation

6.2.2.1 `void ilearnrw.prototype.application.ConsoleMenu.doModalMenu ( )`

6.2.2.2 `Scanner ilearnrw.prototype.application.ConsoleMenu.in ( )`

6.2.2.3 `PrintStream ilearnrw.prototype.application.ConsoleMenu.out ( )`

6.2.2.4 `EConsoleMenuActionResult ilearnrw.prototype.application.ConsoleMenu.showMenu ( )`

6.2.2.5 `ConsoleMenu ilearnrw.prototype.application.ConsoleMenu.subMenu ( String menuTitle, IConsoleMenuAction[] menuActions )`

## 6.2.3 Member Data Documentation

6.2.3.1 `InputStream ilearnrw.prototype.application.ConsoleMenu.mIn` [`package`]

6.2.3.2 `IConsoleMenuAction [] ilearnrw.prototype.application.ConsoleMenu.mMenuActions` [`package`]

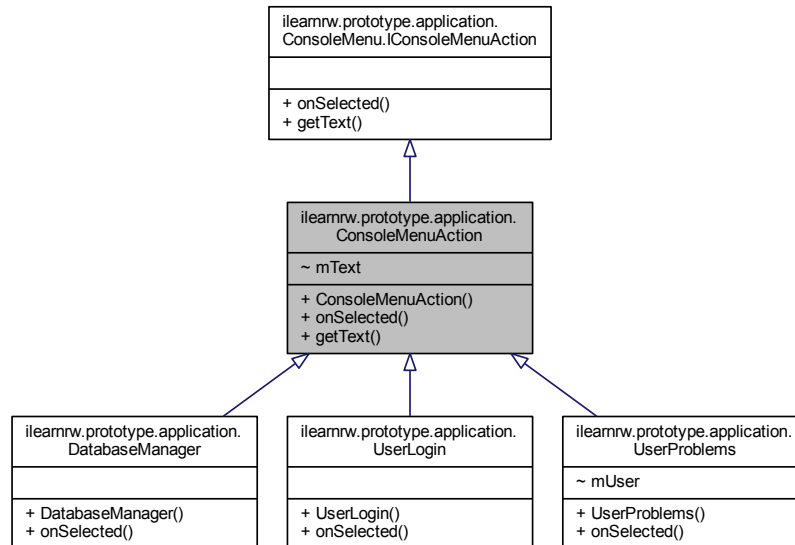
6.2.3.3 `String ilearnrw.prototype.application.ConsoleMenu.mMenuTitle` [`package`]

6.2.3.4 `PrintStream ilearnrw.prototype.application.ConsoleMenu.mOut` [`package`]

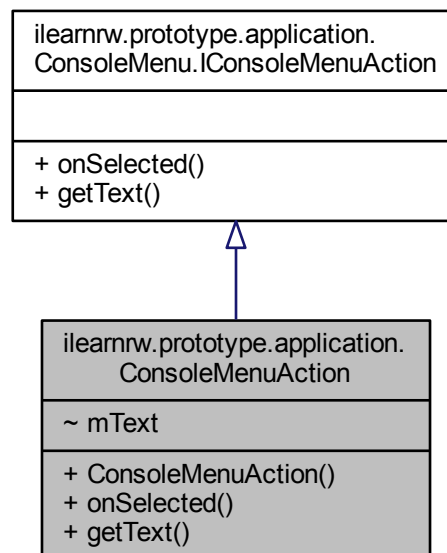
6.2.3.5 `Scanner ilearnrw.prototype.application.ConsoleMenu.mScanner` [`package`]

## 6.3 ilearnrw.prototype.application.ConsoleMenuAction Class Reference

Inheritance diagram for ilearnrw.prototype.application.ConsoleMenuAction:



Collaboration diagram for ilearnrw.prototype.application.ConsoleMenuAction:



## Public Member Functions

- [ConsoleMenuAction](#) (String text)
- [EConsoleMenuActionResult](#) onSelected ([ConsoleMenu](#) menu)
- String [getText](#) ()

## Package Attributes

- String [mText](#)

### 6.3.1 Constructor & Destructor Documentation

6.3.1.1 `ilearnrw.prototype.application.ConsoleMenuAction.ConsoleMenuAction ( String text )`

### 6.3.2 Member Function Documentation

6.3.2.1 `String ilearnrw.prototype.application.ConsoleMenuAction.getText ( )`

Implements [ilearnrw.prototype.application.ConsoleMenu.IConsoleMenuAction](#).

6.3.2.2 `EConsoleMenuActionResult ilearnrw.prototype.application.ConsoleMenuAction.onSelected ( ConsoleMenu menu )`

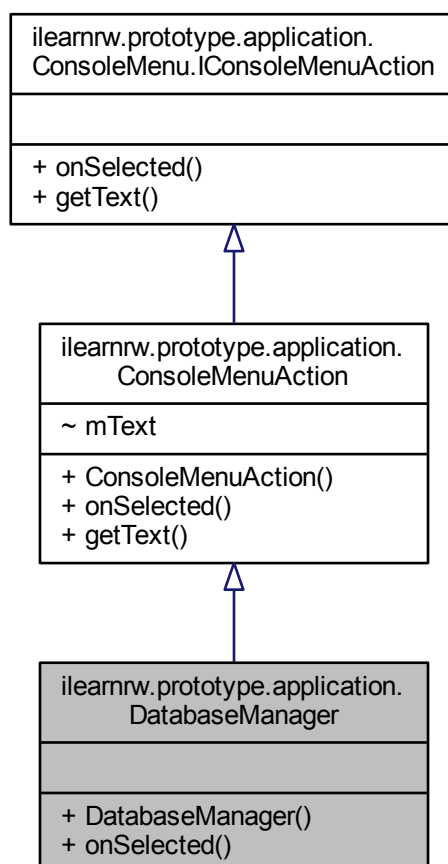
Implements [ilearnrw.prototype.application.ConsoleMenu.IConsoleMenuAction](#).

### 6.3.3 Member Data Documentation

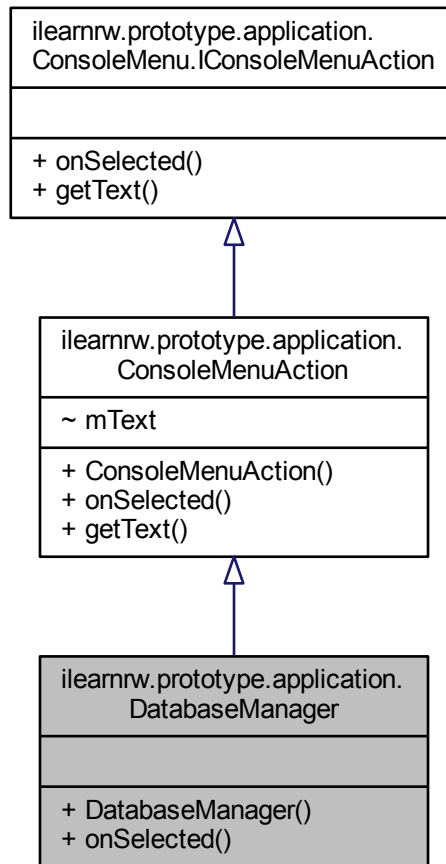
6.3.3.1 `String ilearnrw.prototype.application.ConsoleMenuAction.mText` [package]

## 6.4 ilearnrw.prototype.application.DatabaseManager Class Reference

Inheritance diagram for ilearnrw.prototype.application.DatabaseManager:



Collaboration diagram for ilearnrw.prototype.application.DatabaseManager:



## Classes

- interface [ISelectUser](#)

## Public Member Functions

- [DatabaseManager](#) (String text)
- [EConsoleMenuActionResult](#) `onSelected` ([ConsoleMenu](#) menu)

## Additional Inherited Members

### 6.4.1 Constructor & Destructor Documentation

#### 6.4.1.1 ilearnrw.prototype.application.DatabaseManager.DatabaseManager ( String text )

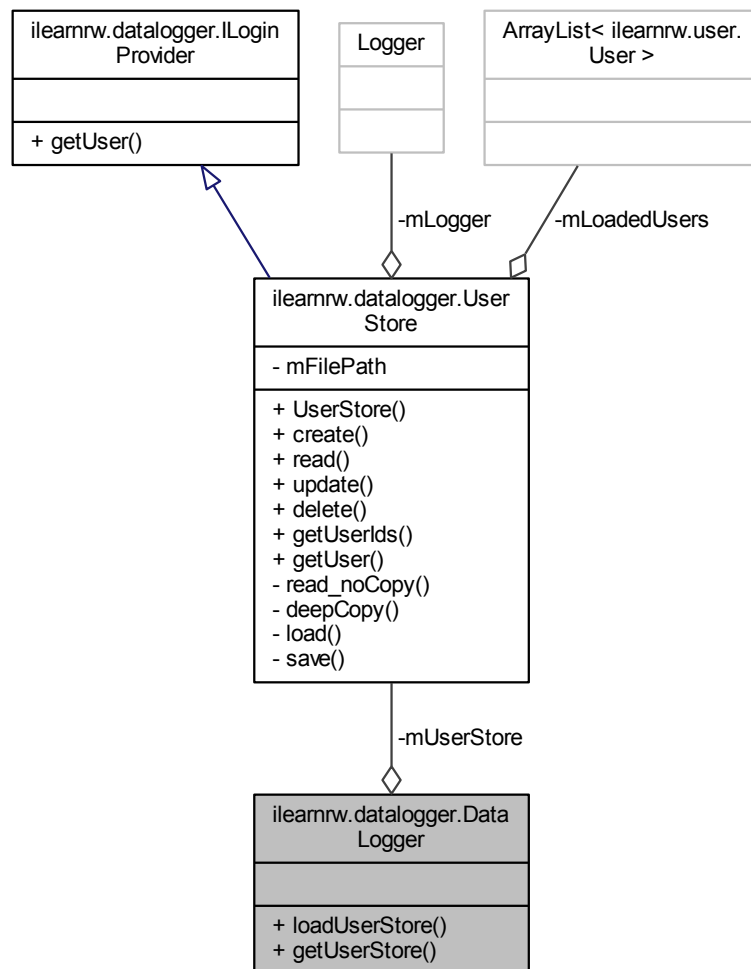
### 6.4.2 Member Function Documentation

6.4.2.1 `EConsoleMenuActionResult` ilearnrw.prototype.application.DatabaseManager.onSelected ( `ConsoleMenu menu` )

Implements [ilearnrw.prototype.application.ConsoleMenu.IConsoleMenuAction](#).

## 6.5 ilearnrw.datalogger.DataLogger Class Reference

Collaboration diagram for ilearnrw.datalogger.DataLogger:



### Public Member Functions

- boolean [loadUserStore](#) (String filePath)
- [UserStore](#) [getUserStore](#) ()

### Private Attributes

- [UserStore](#) [mUserStore](#) = null

### 6.5.1 Detailed Description

[DataLogger](#) class.

### 6.5.2 Member Function Documentation

#### 6.5.2.1 `UserStore` `ilearnrw.dataloader.DataLogger.getUserStore ( )`

##### Returns

The [UserStore](#) if loaded, null otherwise.

#### 6.5.2.2 `boolean` `ilearnrw.dataloader.DataLogger.loadUserStore ( String filePath )`

Loads the [UserStore](#) from disk.

##### Note

`filePath` does not have to exist. But the directory it is in has to.

##### Parameters

<i>filePath</i>	FilePath to a disk copy.
-----------------	--------------------------

##### Returns

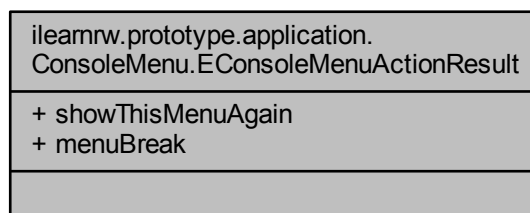
true if successful. false if already loaded.

### 6.5.3 Member Data Documentation

#### 6.5.3.1 `UserStore` `ilearnrw.dataloader.DataLogger.mUserStore = null` `[private]`

## 6.6 `ilearnrw.prototype.application.ConsoleMenu.EConsoleMenuActionResult` Enum Reference

Collaboration diagram for `ilearnrw.prototype.application.ConsoleMenu.EConsoleMenuActionResult`:





## Public Attributes

- [showThisMenuAgain](#)
- [menuBreak](#)

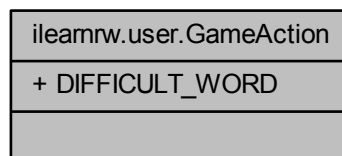
### 6.6.1 Member Data Documentation

6.6.1.1 ilearnrw.prototype.application.ConsoleMenu.EConsoleMenuActionResult.menuBreak

6.6.1.2 ilearnrw.prototype.application.ConsoleMenu.EConsoleMenuActionResult.showThisMenuAgain

## 6.7 ilearnrw.user.GameAction Enum Reference

Collaboration diagram for ilearnrw.user.GameAction:



## Public Attributes

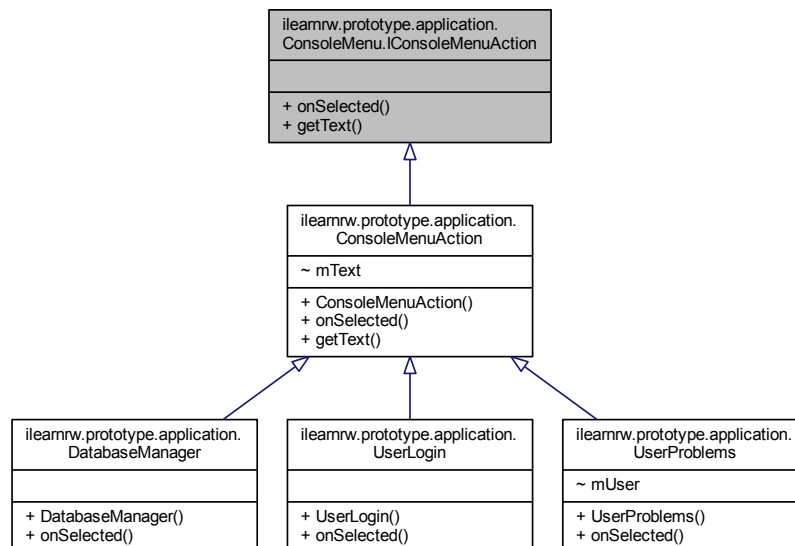
- [DIFFICULT\\_WORD](#)

### 6.7.1 Member Data Documentation

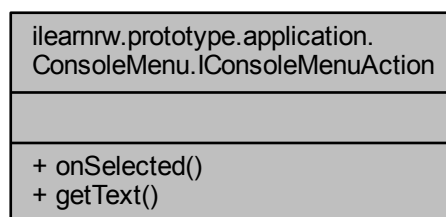
6.7.1.1 ilearnrw.user.GameAction.DIFFICULT\_WORD

## 6.8 ilearnrw.prototype.application.ConsoleMenu.IConsoleMenuAction Interface Reference

Inheritance diagram for ilearnrw.prototype.application.ConsoleMenu.IConsoleMenuAction:



Collaboration diagram for ilearnrw.prototype.application.ConsoleMenu.IConsoleMenuAction:



### Public Member Functions

- [EConsoleMenuActionResult onSelected](#) ([ConsoleMenu](#) menu)
- [String getText](#) ()

### 6.8.1 Member Function Documentation

#### 6.8.1.1 String ilearnrw.prototype.application.ConsoleMenu.IConsoleMenuAction.getText ( )

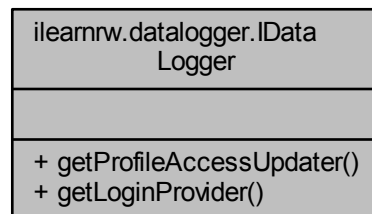
Implemented in [ilearnrw.prototype.application.ConsoleMenuAction](#).

#### 6.8.1.2 EConsoleMenuActionResult ilearnrw.prototype.application.ConsoleMenu.IConsoleMenuAction.onSelected ( ConsoleMenu *menu* )

Implemented in [ilearnrw.prototype.application.UserProblems](#), [ilearnrw.prototype.application.DatabaseManager](#), [ilearnrw.prototype.application.UserLogin](#), and [ilearnrw.prototype.application.ConsoleMenuAction](#).

## 6.9 ilearnrw.datalogger.IDataLogger Interface Reference

Collaboration diagram for ilearnrw.datalogger.IDataLogger:



### Public Member Functions

- [IProfileAccessUpdater](#) `getProfileAccessUpdater ()`
- [ILoginProvider](#) `getLoginProvider ()`

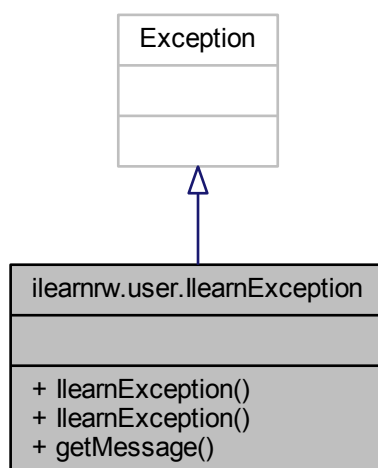
### 6.9.1 Member Function Documentation

#### 6.9.1.1 ILoginProvider ilearnrw.datalogger.IDataLogger.getLoginProvider ( )

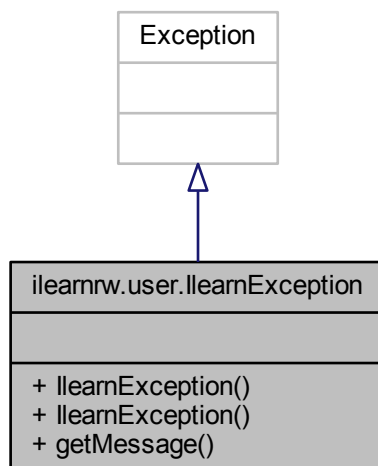
#### 6.9.1.2 IProfileAccessUpdater ilearnrw.datalogger.IDataLogger.getProfileAccessUpdater ( )

## 6.10 ilearnrw.user.IlearnException Class Reference

Inheritance diagram for ilearnrw.user.IlearnException:



Collaboration diagram for ilearnrw.user.IlearnException:



### Public Member Functions

- [IlearnException](#) (String message)
- [IlearnException](#) (String message, Throwable throwable)
- String [getMessage](#) ()

### 6.10.1 Constructor & Destructor Documentation

6.10.1.1 ilearnrw.user.IlearnException.IlearnException ( String *message* )

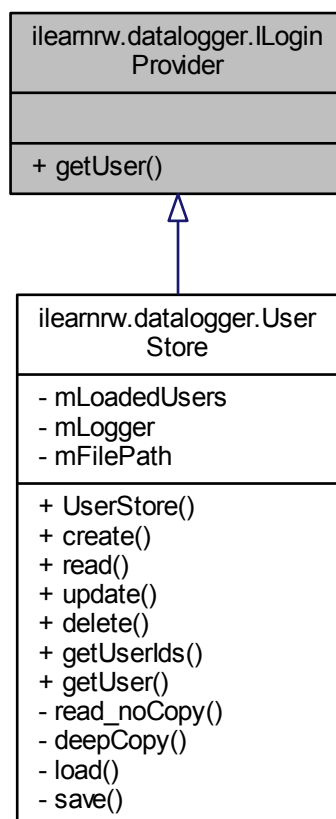
6.10.1.2 ilearnrw.user.IlearnException.IlearnException ( String *message*, Throwable *throwable* )

### 6.10.2 Member Function Documentation

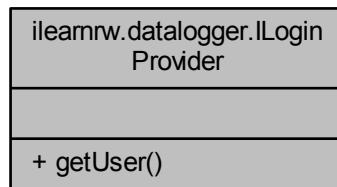
6.10.2.1 String ilearnrw.user.IlearnException.getMessage ( )

## 6.11 ilearnrw.datalogger.ILoginProvider Interface Reference

Inheritance diagram for ilearnrw.datalogger.ILoginProvider:



Collaboration diagram for ilearnrw.datalogger.ILoginProvider:



## Public Member Functions

- [User getUser](#) (String *userName*, String *password*)

*Authenticates a User.*

### 6.11.1 Member Function Documentation

#### 6.11.1.1 User `ilearnrw.datalogger.ILoginProvider.getUser ( String userName, String password )`

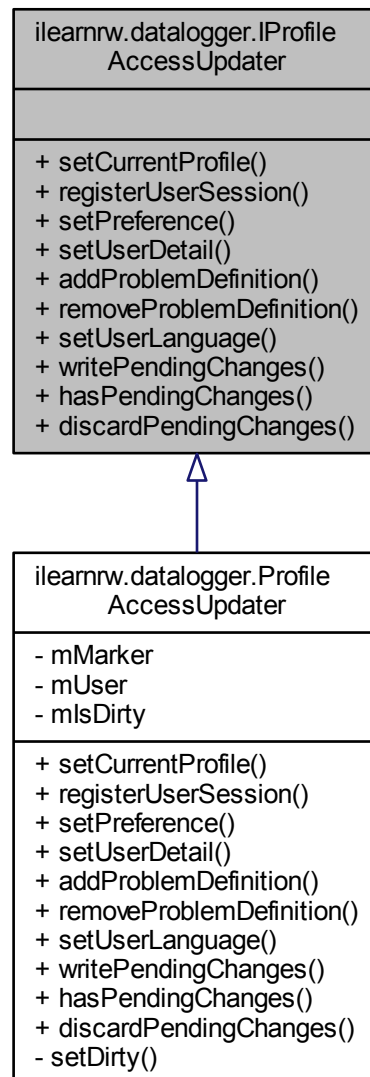
Authenticates a User.

Implemented in [ilearnrw.datalogger.UserStore](#).

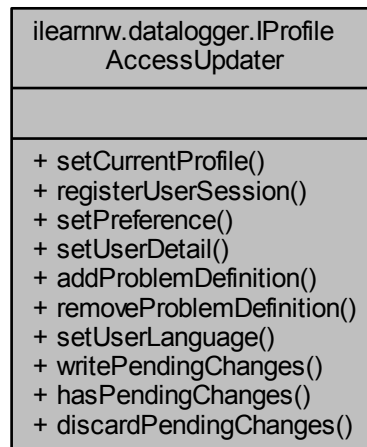
## 6.12 ilearnrw.datalogger.IProfileAccessUpdater Interface Reference

Interface to update profiles in the iLearnRW system.

Inheritance diagram for ilearnrw.datalogger.IProfileAccessUpdater:



Collaboration diagram for `ilearnrw.datalogger.IProfileAccessUpdater`:



## Classes

- class [PendingChangesAvailable](#)  
*Indicates that an action cannot be performed due to pending changes.*
- class [PermissionException](#)  
*The marker does not have permission to perform the requested change.*

## Public Member Functions

- boolean [setCurrentProfile](#) ([UserProfile](#) profile, [UserProfile](#) markerProfile) throws [PermissionException](#), [PendingChangesAvailable](#)  
*Sets the userprofile to work with and who will be the *marker* for the changes.*
- boolean [registerUserSession](#) ([UserSession](#) session)  
*Registers a new User session in the log.*
- boolean [setPreference](#) ([UserPreferences](#) preferences)  
*Registers a change to the users preferences.*
- boolean [setUserDetail](#) ([UserDetails](#) userDetails)  
*Updates a UserDetail.*
- boolean [addProblemDefinition](#) (String URI)  
*Adds a predefined ProblemDefinition to the user profile.*
- boolean [removeProblemDefinition](#) (String URI)  
*Removes a ProblemDefinition.*
- boolean [setUserLanguage](#) ([LanguageCode](#) Language)  
*Sets the userlanguage.*
- [UserProfile](#) [writePendingChanges](#) ()  
*Writes unwritten changes to the back end (persistent storage).*
- boolean [hasPendingChanges](#) ()  
*Checks if there are any changes pending.*
- boolean [discardPendingChanges](#) ()  
*Discards all pending changes.*



### 6.12.1 Detailed Description

Interface to update profiles in the iLearnRW system.

This interface assumes that `Apps` would get a read-only copy of the users profile and would then have to use this interface to make changes to the profile. When the profile has been updated a new readonly copy would have to be fetched.

A nice side-effect is that we can make sure to log all changes to the profile.

Updater also manages

Registry - Handles/stores users/profiles.

UserAdd/delete etc.. Get user (given userid, password)

### 6.12.2 Member Function Documentation

#### 6.12.2.1 `boolean ilearnrw.datalogger.IProfileAccessUpdater.addProblemDefinition ( String URI )`

Adds a predefined `ProblemDefinition` to the user profile.

Implemented in [ilearnrw.datalogger.ProfileAccessUpdater](#).

#### 6.12.2.2 `boolean ilearnrw.datalogger.IProfileAccessUpdater.discardPendingChanges ( )`

Discards all pending changes.

Implemented in [ilearnrw.datalogger.ProfileAccessUpdater](#).

#### 6.12.2.3 `boolean ilearnrw.datalogger.IProfileAccessUpdater.hasPendingChanges ( )`

Checks if there are any changes pending.

Implemented in [ilearnrw.datalogger.ProfileAccessUpdater](#).

#### 6.12.2.4 `boolean ilearnrw.datalogger.IProfileAccessUpdater.registerUserSession ( UserSession session )`

Registers a new User session in the log.

Implemented in [ilearnrw.datalogger.ProfileAccessUpdater](#).

#### 6.12.2.5 `boolean ilearnrw.datalogger.IProfileAccessUpdater.removeProblemDefinition ( String URI )`

Removes a `ProblemDefinition`.

Implemented in [ilearnrw.datalogger.ProfileAccessUpdater](#).

#### 6.12.2.6 `boolean ilearnrw.datalogger.IProfileAccessUpdater.setCurrentProfile ( UserProfile profile, UserProfile markerProfile ) throws PermissionException, PendingChangesAvailable`

Sets the userprofile to work with and who will be the `marker` for the changes.

Doing this once saves the application from passing the userprofile to all functions.

The `marker` can be the same as the profile to be edited.

## Exceptions

<a href="#">PermissionException</a>	Thrown when the admin profile does not have permissions to update the profile.
<a href="#">PendingChangesAvailable</a>	Thrown when there are pending changes are available.

## Parameters

<i>profile</i>	Profile of the user to update
<i>markerProfile</i>	Profile of the user who is doing the update

Implemented in [ilearnrw.datalogger.ProfileAccessUpdater](#).

#### 6.12.2.7 boolean ilearnrw.datalogger.IProfileAccessUpdater.setPreference ( **UserPreferences** *preferences* )

Registers a change to the users preferences.

Should changes to user preferences be logged so that teachers/experts can be notified/see that a preference has been changed?

Implemented in [ilearnrw.datalogger.ProfileAccessUpdater](#).

#### 6.12.2.8 boolean ilearnrw.datalogger.IProfileAccessUpdater.setUserDetail ( **UserDetails** *userDetails* )

Updates a UserDetail.

Implemented in [ilearnrw.datalogger.ProfileAccessUpdater](#).

#### 6.12.2.9 boolean ilearnrw.datalogger.IProfileAccessUpdater.setUserLanguage ( **LanguageCode** *Language* )

Sets the userlanguage.

Implemented in [ilearnrw.datalogger.ProfileAccessUpdater](#).

#### 6.12.2.10 **UserProfile** ilearnrw.datalogger.IProfileAccessUpdater.writePendingChanges ( )

Writes unwritten changes to the back end (persistent storage).

This will flush all pending operations in the same order as they are added. This is important if for example the same preference is set twice.

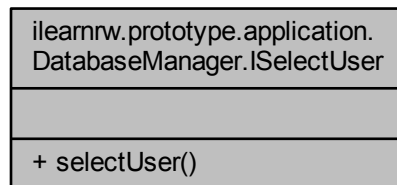
**Returns**

The new profile after the changes has been applied.

Implemented in [ilearnrw.datalogger.ProfileAccessUpdater](#).

## 6.13 ilearnrw.prototype.application.DatabaseManager.ISelectUser Interface Reference

Collaboration diagram for ilearnrw.prototype.application.DatabaseManager.ISelectUser:

**Public Member Functions**

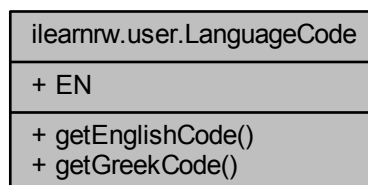
- [User selectUser](#) ([ConsoleMenu menu](#))

### 6.13.1 Member Function Documentation

6.13.1.1 **User** `ilearnrw.prototype.application.DatabaseManager.ISelectUser.selectUser ( ConsoleMenu menu )`

## 6.14 ilearnrw.user.LanguageCode Enum Reference

Collaboration diagram for ilearnrw.user.LanguageCode:

**Static Public Member Functions**

- static byte [getEnglishCode](#) ()

- static byte [getGreekCode](#) ()

### Public Attributes

- [EN](#)

### 6.14.1 Member Function Documentation

6.14.1.1 static byte `ilearnrw.user.LanguageCode.getEnglishCode ( )` [static]

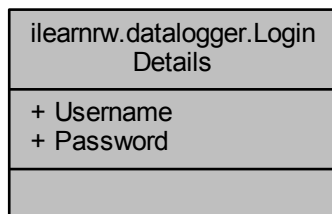
6.14.1.2 static byte `ilearnrw.user.LanguageCode.getGreekCode ( )` [static]

### 6.14.2 Member Data Documentation

6.14.2.1 `ilearnrw.user.LanguageCode.EN`

## 6.15 ilearnrw.datalogger.LoginDetails Class Reference

Collaboration diagram for `ilearnrw.datalogger.LoginDetails`:



### Public Attributes

- String [Username](#)
- String [Password](#)

### 6.15.1 Member Data Documentation

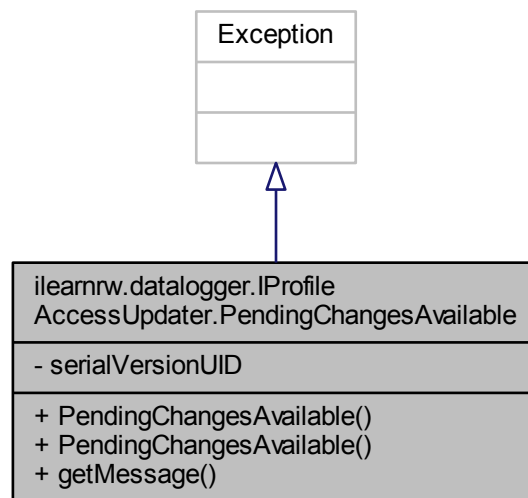
6.15.1.1 String `ilearnrw.datalogger.LoginDetails.Password`

6.15.1.2 String `ilearnrw.datalogger.LoginDetails.Username`

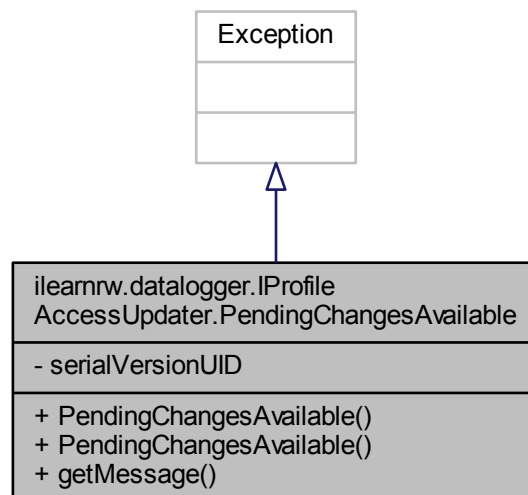
## 6.16 ilearnrw.datalogger.IProfileAccessUpdater.PendingChangesAvailable Class Reference

Indicates that an action cannot be performed due to pending changes.

Inheritance diagram for ilearnrw.datalogger.IProfileAccessUpdater.PendingChangesAvailable:



Collaboration diagram for ilearnrw.datalogger.IProfileAccessUpdater.PendingChangesAvailable:



## Public Member Functions

- [PendingChangesAvailable](#) (String message)
- [PendingChangesAvailable](#) (String message, Throwable throwable)
- String [getMessage](#) ()

## Static Private Attributes

- static final long `serialVersionUID` = 1L

### 6.16.1 Detailed Description

Indicates that an action cannot be performed due to pending changes.

If the action was to be performed the pending changes would be lost.

### 6.16.2 Constructor & Destructor Documentation

6.16.2.1 `ilearnrw.datalogger.IProfileAccessUpdater.PendingChangesAvailable.PendingChangesAvailable ( String message )`

6.16.2.2 `ilearnrw.datalogger.IProfileAccessUpdater.PendingChangesAvailable.PendingChangesAvailable ( String message, Throwable throwable )`

### 6.16.3 Member Function Documentation

6.16.3.1 `String ilearnrw.datalogger.IProfileAccessUpdater.PendingChangesAvailable.getMessage ( )`

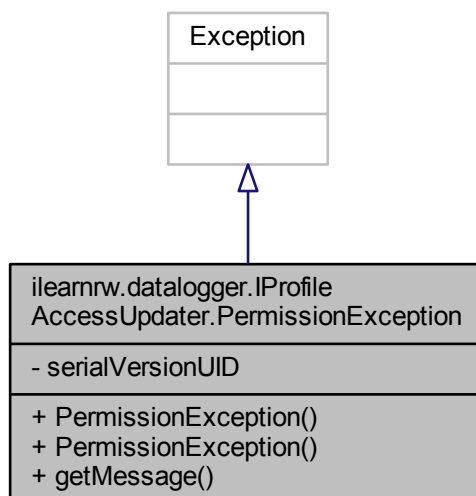
### 6.16.4 Member Data Documentation

6.16.4.1 `final long ilearnrw.datalogger.IProfileAccessUpdater.PendingChangesAvailable.serialVersionUID = 1L` `[static]`, `[private]`

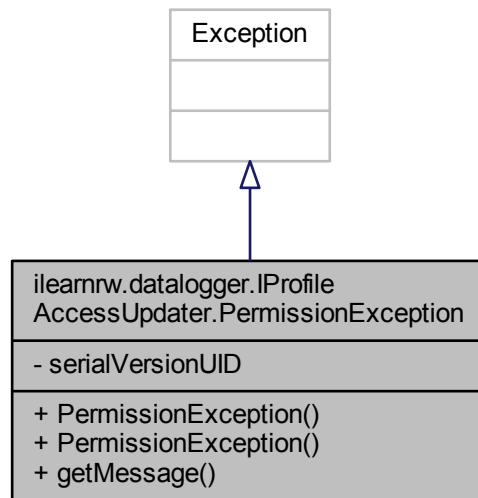
## 6.17 ilearnrw.datalogger.IProfileAccessUpdater.PermissionException Class Reference

The marker does not have permission to perform the requested change.

Inheritance diagram for `ilearnrw.datalogger.IProfileAccessUpdater.PermissionException`:



Collaboration diagram for ilearnrw.datalogger.IProfileAccessUpdater.PermissionException:



## Public Member Functions

- [PermissionException](#) (String message)
- [PermissionException](#) (String message, Throwable throwable)
- String [getMessage](#) ()

## Static Private Attributes

- static final long [serialVersionUID](#) = 1L

## 6.17.1 Detailed Description

The marker does not have permission to perform the requested change.

## 6.17.2 Constructor & Destructor Documentation

6.17.2.1 `ilearnrw.datalogger.IProfileAccessUpdater.PermissionException.PermissionException ( String message )`

6.17.2.2 `ilearnrw.datalogger.IProfileAccessUpdater.PermissionException.PermissionException ( String message, Throwable throwable )`

## 6.17.3 Member Function Documentation

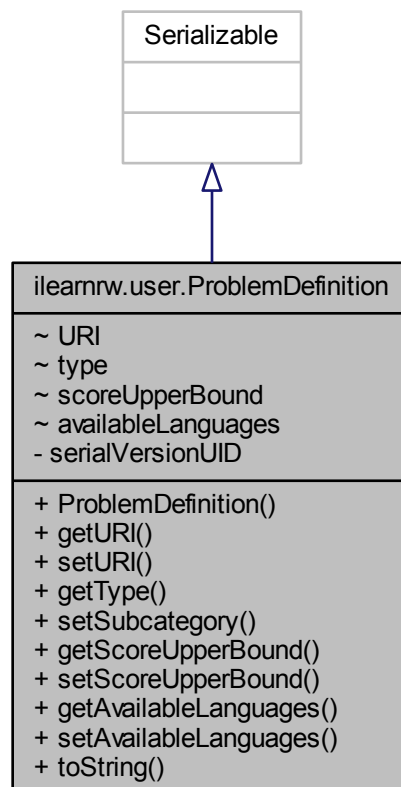
6.17.3.1 `String ilearnrw.datalogger.IProfileAccessUpdater.PermissionException.getMessage ( )`

## 6.17.4 Member Data Documentation

6.17.4.1 `final long ilearnrw.datalogger.IProfileAccessUpdater.PermissionException.serialVersionUID = 1L` `[static]`,  
`[private]`

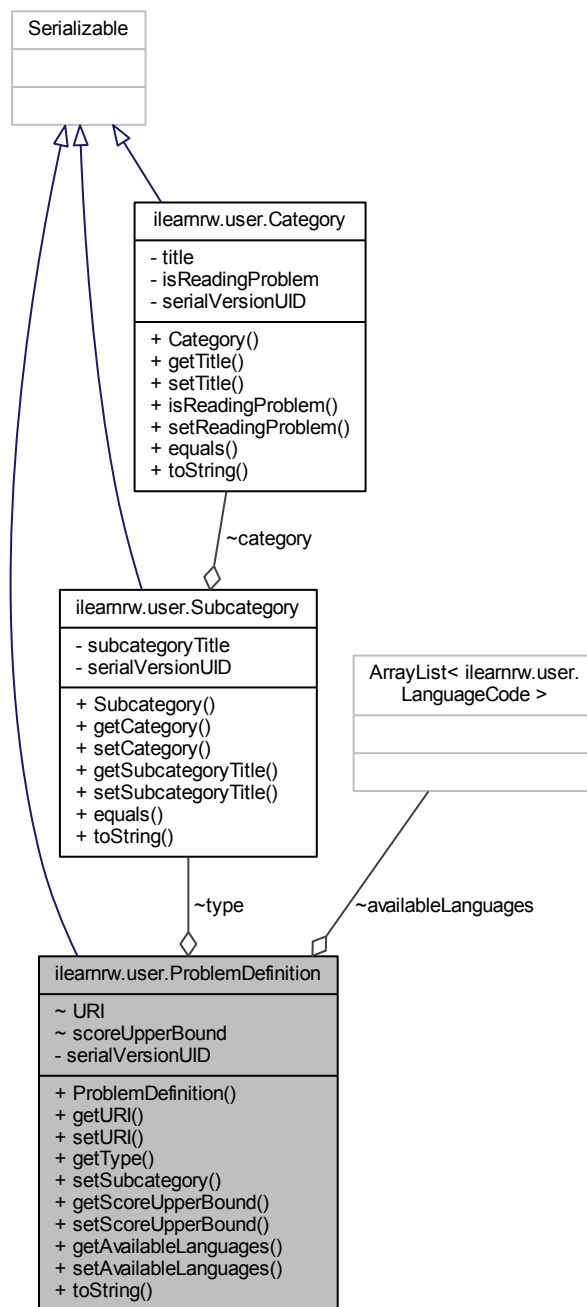
## 6.18 ilearnrw.user.ProblemDefinition Class Reference

Inheritance diagram for ilearnrw.user.ProblemDefinition:





Collaboration diagram for ilearnrw.user.ProblemDefinition:



## Public Member Functions

- `ProblemDefinition` (String `URI`, `Subcategory` `type`, int `scoreUpperBound`, ArrayList< `LanguageCode` > `availableLanguages`)
- String `getURI` ()
- void `setURI` (String `uRI`)
- `Subcategory` `getType` ()
- void `setSubcategory` (`Subcategory` `type`)

- int [getScoreUpperBound](#) ()
- void [setScoreUpperBound](#) (int [scoreUpperBound](#))
- ArrayList< [LanguageCode](#) > [getAvailableLanguages](#) ()
- void [setAvailableLanguages](#) (ArrayList< [LanguageCode](#) > [availableLanguages](#))
- String [toString](#) ()

### Package Attributes

- String [URI](#)
- [Subcategory](#) type
- int [scoreUpperBound](#)
- ArrayList< [LanguageCode](#) > [availableLanguages](#)

### Static Private Attributes

- static final long [serialVersionUID](#) = 1L

### 6.18.1 Constructor & Destructor Documentation

- 6.18.1.1 `ilearnrw.user.ProblemDefinition.ProblemDefinition ( String URI, Subcategory type, int scoreUpperBound, ArrayList< LanguageCode > availableLanguages )`

### 6.18.2 Member Function Documentation

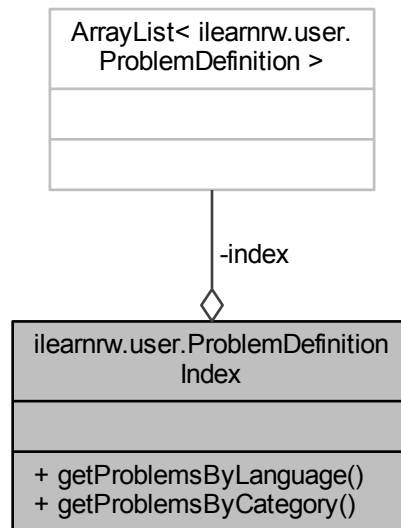
- 6.18.2.1 `ArrayList<LanguageCode> ilearnrw.user.ProblemDefinition.getAvailableLanguages ( )`
- 6.18.2.2 `int ilearnrw.user.ProblemDefinition.getScoreUpperBound ( )`
- 6.18.2.3 `Subcategory ilearnrw.user.ProblemDefinition.getType ( )`
- 6.18.2.4 `String ilearnrw.user.ProblemDefinition.getURI ( )`
- 6.18.2.5 `void ilearnrw.user.ProblemDefinition.setAvailableLanguages ( ArrayList< LanguageCode > availableLanguages )`
- 6.18.2.6 `void ilearnrw.user.ProblemDefinition.setScoreUpperBound ( int scoreUpperBound )`
- 6.18.2.7 `void ilearnrw.user.ProblemDefinition.setSubcategory ( Subcategory type )`
- 6.18.2.8 `void ilearnrw.user.ProblemDefinition.setURI ( String uRI )`
- 6.18.2.9 `String ilearnrw.user.ProblemDefinition.toString ( )`

### 6.18.3 Member Data Documentation

- 6.18.3.1 `ArrayList<LanguageCode> ilearnrw.user.ProblemDefinition.availableLanguages` [package]
- 6.18.3.2 `int ilearnrw.user.ProblemDefinition.scoreUpperBound` [package]
- 6.18.3.3 `final long ilearnrw.user.ProblemDefinition.serialVersionUID = 1L` [static],[private]
- 6.18.3.4 `Subcategory ilearnrw.user.ProblemDefinition.type` [package]
- 6.18.3.5 `String ilearnrw.user.ProblemDefinition.URI` [package]

## 6.19 ilearnrw.user.ProblemDefinitionIndex Class Reference

Collaboration diagram for ilearnrw.user.ProblemDefinitionIndex:



### Public Member Functions

- `ArrayList< ProblemDefinition > getProblemsByLanguage (LanguageCode x)`
- `ArrayList< ProblemDefinition > getProblemsByCategory (Category x)`

### Private Attributes

- `ArrayList< ProblemDefinition > index`

### 6.19.1 Member Function Documentation

6.19.1.1 `ArrayList<ProblemDefinition> ilearnrw.user.ProblemDefinitionIndex.getProblemsByCategory ( Category x )`

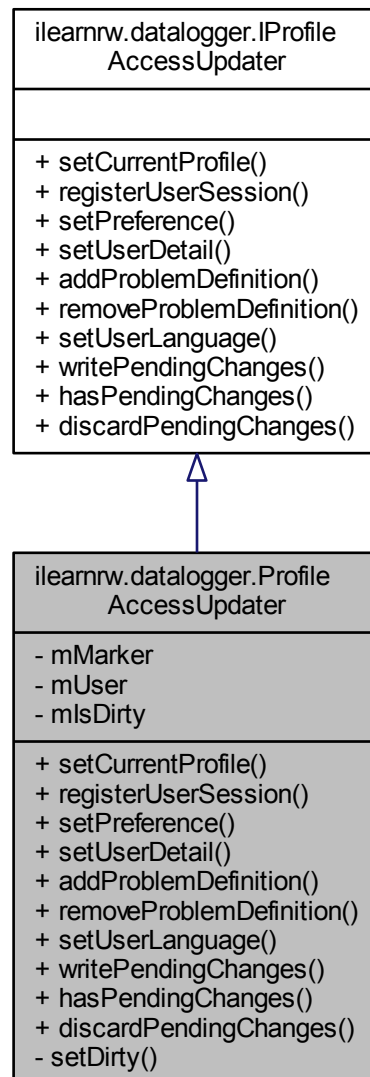
6.19.1.2 `ArrayList<ProblemDefinition> ilearnrw.user.ProblemDefinitionIndex.getProblemsByLanguage ( LanguageCode x )`

### 6.19.2 Member Data Documentation

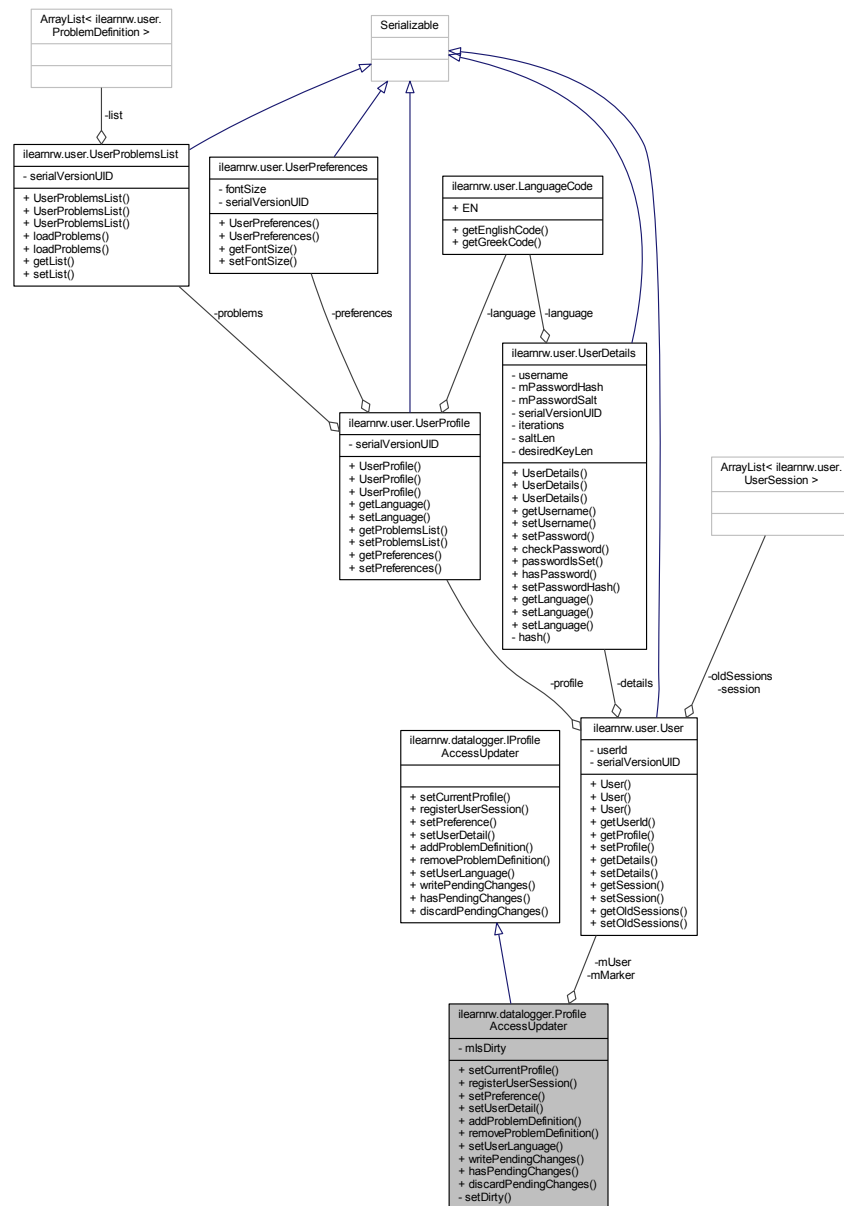
6.19.2.1 `ArrayList<ProblemDefinition> ilearnrw.user.ProblemDefinitionIndex.index [private]`

## 6.20 ilearnrw.datalogger.ProfileAccessUpdater Class Reference

Inheritance diagram for ilearnrw.datalogger.ProfileAccessUpdater:



Collaboration diagram for ilearnrw.datalogger.ProfileAccessUpdater:



## Public Member Functions

- boolean [setCurrentProfile](#) ([UserProfile](#) profile, [UserProfile](#) markerProfile) throws `PermissionException`, `PendingChangesAvailable`  
*Sets the userprofile to work with and who will be the `marker` for the changes.*
- boolean [registerUserSession](#) ([UserSession](#) session)  
*Registers a new User session in the log.*
- boolean [setPreference](#) ([UserPreferences](#) pref)  
*Registers a change to the users preferences.*
- boolean [setUserDetail](#) ([UserDetails](#) userDetails)  
*Updates a UserDetail.*
- boolean [addProblemDefinition](#) (String URI)

- Adds a predefined ProblemDefinition to the user profile.*

  - boolean [removeProblemDefinition](#) (String URI)

*Removes a ProblemDefinition.*
- boolean [setUserLanguage](#) (LanguageCode Language)

*Sets the userlanguage.*
- [UserProfile](#) [writePendingChanges](#) ()

*Writes unwritten changes to the back end (persistent storage).*
- boolean [hasPendingChanges](#) ()

*Checks if there are any changes pending.*
- boolean [discardPendingChanges](#) ()

*Discards all pending changes.*

## Private Member Functions

- void [setDirty](#) ()

## Private Attributes

- [User](#) [mMarker](#)
- The currently selected marker profile.*
- [User](#) [mUser](#)
- The profile to be updated.*
- boolean [mIsDirty](#)
- Access to storage-backend.*

## 6.20.1 Detailed Description

### Note

This class currently does not make use of the potential to log every action and the marker.

## 6.20.2 Member Function Documentation

### 6.20.2.1 boolean [ilearnrw.datalogger.ProfileAccessUpdater.addProblemDefinition](#) ( String URI )

Adds a predefined ProblemDefinition to the user profile.

Implements [ilearnrw.datalogger.IProfileAccessUpdater](#).

### 6.20.2.2 boolean [ilearnrw.datalogger.ProfileAccessUpdater.discardPendingChanges](#) ( )

Discards all pending changes.

Implements [ilearnrw.datalogger.IProfileAccessUpdater](#).

### 6.20.2.3 boolean [ilearnrw.datalogger.ProfileAccessUpdater.hasPendingChanges](#) ( )

Checks if there are any changes pending.

Implements [ilearnrw.datalogger.IProfileAccessUpdater](#).

**6.20.2.4** boolean ilearnrw.datalogger.ProfileAccessUpdater.registerUserSession ( *UserSession session* )

Registers a new User session in the log.

Implements [ilearnrw.datalogger.IProfileAccessUpdater](#).

**6.20.2.5** boolean ilearnrw.datalogger.ProfileAccessUpdater.removeProblemDefinition ( *String URI* )

Removes a ProblemDefinition.

Implements [ilearnrw.datalogger.IProfileAccessUpdater](#).

**6.20.2.6** boolean ilearnrw.datalogger.ProfileAccessUpdater.setCurrentProfile ( *UserProfile profile*, *UserProfile markerProfile* ) throws **PermissionException**, **PendingChangesAvailable**

Sets the userprofile to work with and who will be the `marker` for the changes.

Doing this once saves the application from passing the userprofile to all functions.

The `marker` can be the same as the profile to be edited.

**Exceptions**

<i>PermissionException</i>	Thrown when the admin profile does not have permissions to update the profile.
<i>PendingChangesAvailable</i>	Thrown when there are pending changes are available.

Implements [ilearnrw.datalogger.IProfileAccessUpdater](#).

**6.20.2.7** void ilearnrw.datalogger.ProfileAccessUpdater.setDirty ( ) [private]**6.20.2.8** boolean ilearnrw.datalogger.ProfileAccessUpdater.setPreference ( *UserPreferences preferences* )

Registers a change to the users preferences.

Should changes to user preferences be logged so that teachers/experts can be notified/see that a preference has been changed?

Implements [ilearnrw.datalogger.IProfileAccessUpdater](#).

**6.20.2.9** boolean ilearnrw.datalogger.ProfileAccessUpdater.setUserDetail ( *UserDetails userDetails* )

Updates a UserDetail.

Implements [ilearnrw.datalogger.IProfileAccessUpdater](#).

**6.20.2.10** boolean ilearnrw.datalogger.ProfileAccessUpdater.setUserLanguage ( *LanguageCode language* )

Sets the userlanguage.

Implements [ilearnrw.datalogger.IProfileAccessUpdater](#).

**6.20.2.11** **UserProfile** ilearnrw.datalogger.ProfileAccessUpdater.writePendingChanges ( )

Writes unwritten changes to the back end (persistent storage).

This will flush all pending operations in the same order as they are added. This is important if for example the same preference is set twice.

**Returns**

The new profile after the changes has been applied.

Implements [ilearnrw.datalogger.IProfileAccessUpdater](#).

### 6.20.3 Member Data Documentation

**6.20.3.1** `boolean ilearnrw.datalogger.ProfileAccessUpdater.mIsDirty` `[private]`

Access to storage-backend.

**6.20.3.2** `User ilearnrw.datalogger.ProfileAccessUpdater.mMarker` `[private]`

The currently selected marker profile.

The marker profile is used to show who made a certain change/log in the system

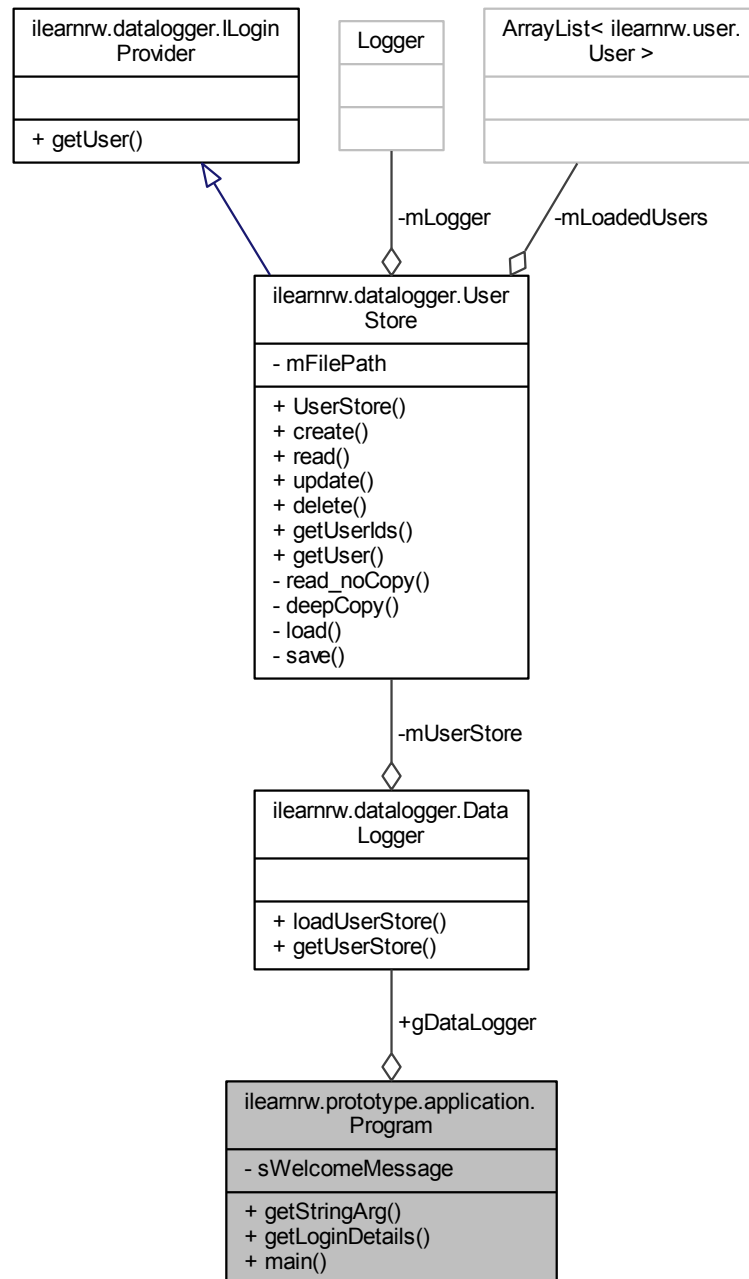
**6.20.3.3** `User ilearnrw.datalogger.ProfileAccessUpdater.mUser` `[private]`

The profile to be updated.



## 6.21 ilearnrw.prototype.application.Program Class Reference

Collaboration diagram for ilearnrw.prototype.application.Program:



### Static Public Member Functions

- static String [getStringArg](#) (String name, String[] args) throws Exception
- static [LoginDetails](#) [getLoginDetails](#) ()
- static void [main](#) (String[] args)

## Static Public Attributes

- static `ILogger gDataLogger = new ILogger()`

## Static Private Attributes

- static `String sWelcomeMessage`

## 6.21.1 Member Function Documentation

6.21.1.1 `static LoginDetails ilearnrw.prototype.application.Program.getLoginDetails ( ) [static]`

6.21.1.2 `static String ilearnrw.prototype.application.Program.getStringArg ( String name, String[] args ) throws Exception [static]`

6.21.1.3 `static void ilearnrw.prototype.application.Program.main ( String[] args ) [static]`

### Parameters

<i>args</i>	Commandline arguments.
-------------	------------------------

Commandline arguments:

–db <filePath> Database file to use (required)

## 6.21.2 Member Data Documentation

6.21.2.1 `ILogger ilearnrw.prototype.application.Program.gDataLogger = new ILogger()` [static]

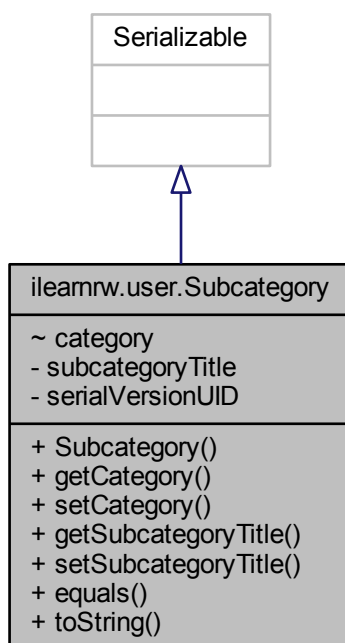
6.21.2.2 `String ilearnrw.prototype.application.Program.sWelcomeMessage` [static],[private]

### Initial value:

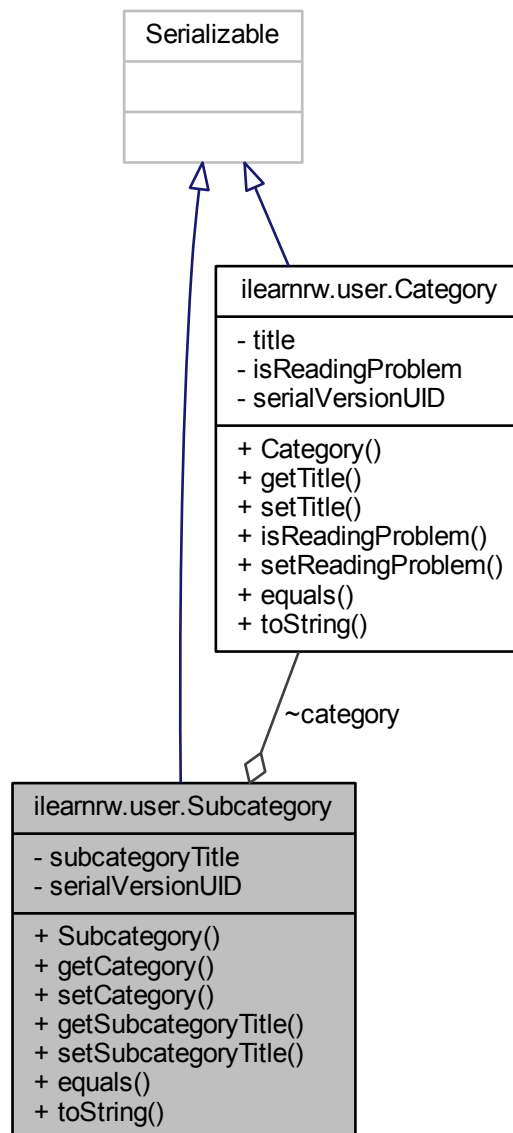
```
= "Welcome to the iLearnRW prototype application\n"
+ "\n"
+ "Developer notes:\n"
+ "\tUsernames are not unique.\n"
+ "\tUsernames cannot contain spaces.\n"
+ "\n"
```

## 6.22 ilearnrw.user.Subcategory Class Reference

Inheritance diagram for ilearnrw.user.Subcategory:



Collaboration diagram for `ilearnrw.user.Subcategory`:



## Public Member Functions

- `Subcategory` (`String subcategoryTitle`, `Category category`)
- `Category getCategory` ()
- `void setCategory` (`Category category`)
- `String getSubcategoryTitle` ()
- `void setSubcategoryTitle` (`String subcategoryTitle`)
- `boolean equals` (`Object obj`)
- `String toString` ()

### Package Attributes

- [Category category](#)

### Private Attributes

- String [subcategoryTitle](#)

### Static Private Attributes

- static final long [serialVersionUID](#) = 1L

## 6.22.1 Constructor & Destructor Documentation

6.22.1.1 `ilearnrw.user.Subcategory.Subcategory ( String subcategoryTitle, Category category )`

## 6.22.2 Member Function Documentation

6.22.2.1 `boolean ilearnrw.user.Subcategory.equals ( Object obj )`

6.22.2.2 `Category ilearnrw.user.Subcategory.getCategory ( )`

6.22.2.3 `String ilearnrw.user.Subcategory.getSubcategoryTitle ( )`

6.22.2.4 `void ilearnrw.user.Subcategory.setCategory ( Category category )`

6.22.2.5 `void ilearnrw.user.Subcategory.setSubcategoryTitle ( String subcategoryTitle )`

6.22.2.6 `String ilearnrw.user.Subcategory.toString ( )`

## 6.22.3 Member Data Documentation

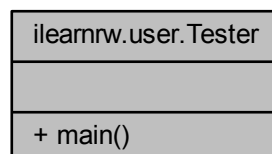
6.22.3.1 `Category ilearnrw.user.Subcategory.category` [package]

6.22.3.2 `final long ilearnrw.user.Subcategory.serialVersionUID = 1L` [static],[private]

6.22.3.3 `String ilearnrw.user.Subcategory.subcategoryTitle` [private]

## 6.23 ilearnrw.user.Tester Class Reference

Collaboration diagram for ilearnrw.user.Tester:



## Static Public Member Functions

- static void [main](#) (String[] args)

### 6.23.1 Member Function Documentation

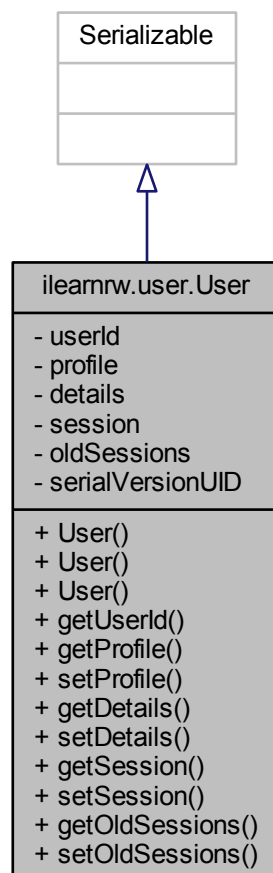
6.23.1.1 static void `ilearnrw.user.Tester.main ( String[] args )` [`static`]

#### Parameters

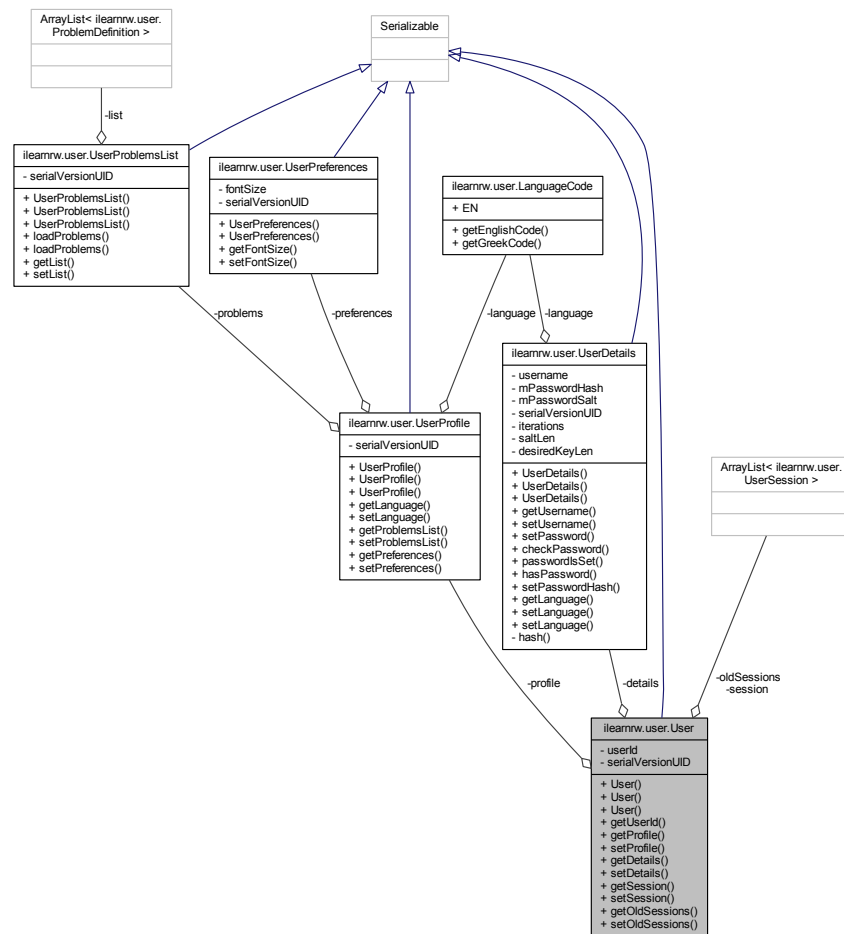
<i>args</i>	
-------------	--

## 6.24 ilearnrw.user.User Class Reference

Inheritance diagram for `ilearnrw.user.User`:



Collaboration diagram for ilearnrw.user.User:



## Public Member Functions

- [User](#) (int [userId](#), [UserProfile](#) [profile](#), [UserDetails](#) [details](#), [ArrayList](#)< [UserSession](#) > [oldSessions](#))
- [User](#) ([LanguageCode](#) [lc](#))
- [User](#) (int [userId](#))
- int [getUserId](#) ()
- [UserProfile](#) [getProfile](#) ()
- void [setProfile](#) ([UserProfile](#) [profile](#))
- [UserDetails](#) [getDetails](#) ()
- void [setDetails](#) ([UserDetails](#) [details](#))
- [ArrayList](#)< [UserSession](#) > [getSession](#) ()
- void [setSession](#) ([ArrayList](#)< [UserSession](#) > [session](#))
- [ArrayList](#)< [UserSession](#) > [getOldSessions](#) ()
- void [setOldSessions](#) ([ArrayList](#)< [UserSession](#) > [oldSessions](#))

## Private Attributes

- int [userId](#)
- [UserProfile](#) [profile](#)
- [UserDetails](#) [details](#)

- ArrayList< [UserSession](#) > [session](#)
- ArrayList< [UserSession](#) > [oldSessions](#)

### Static Private Attributes

- static final long [serialVersionUID](#) = 1L

### 6.24.1 Constructor & Destructor Documentation

6.24.1.1 `ilearnrw.user.User.User ( int userId, UserProfile profile, UserDetails details, ArrayList< UserSession > oldSessions )`

6.24.1.2 `ilearnrw.user.User.User ( LanguageCode lc )`

6.24.1.3 `ilearnrw.user.User.User ( int userId )`

### 6.24.2 Member Function Documentation

6.24.2.1 `UserDetails ilearnrw.user.User.getDetails ( )`

6.24.2.2 `ArrayList<UserSession> ilearnrw.user.User.getOldSessions ( )`

6.24.2.3 `UserProfile ilearnrw.user.User.getProfile ( )`

6.24.2.4 `ArrayList<UserSession> ilearnrw.user.User.getSession ( )`

6.24.2.5 `int ilearnrw.user.User.getUserId ( )`

6.24.2.6 `void ilearnrw.user.User.setDetails ( UserDetails details )`

6.24.2.7 `void ilearnrw.user.User.setOldSessions ( ArrayList< UserSession > oldSessions )`

6.24.2.8 `void ilearnrw.user.User.setProfile ( UserProfile profile )`

6.24.2.9 `void ilearnrw.user.User.setSession ( ArrayList< UserSession > session )`

### 6.24.3 Member Data Documentation

6.24.3.1 `UserDetails ilearnrw.user.User.details [private]`

6.24.3.2 `ArrayList<UserSession> ilearnrw.user.User.oldSessions [private]`

6.24.3.3 `UserProfile ilearnrw.user.User.profile [private]`

6.24.3.4 `final long ilearnrw.user.User.serialVersionUID = 1L [static],[private]`

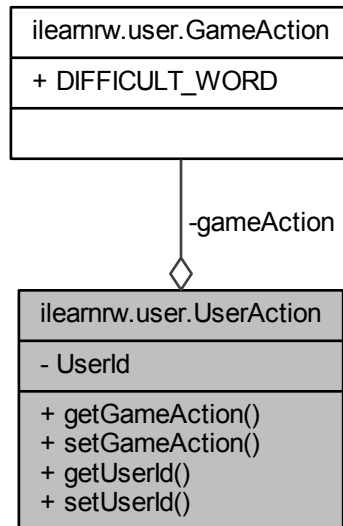
6.24.3.5 `ArrayList<UserSession> ilearnrw.user.User.session [private]`

6.24.3.6 `int ilearnrw.user.User.userId [private]`



## 6.25 ilearnrw.user.UserAction Class Reference

Collaboration diagram for ilearnrw.user.UserAction:



### Public Member Functions

- [GameAction](#) [getGameAction](#) ( )
- void [setGameAction](#) ([GameAction](#) [gameAction](#))
- int [getUserId](#) ( )
- void [setUserId](#) (int [userId](#))

### Private Attributes

- [GameAction](#) [gameAction](#)
- int [UserId](#)

### 6.25.1 Member Function Documentation

6.25.1.1 [GameAction](#) [ilearnrw.user.UserAction.getGameAction](#) ( )

6.25.1.2 int [ilearnrw.user.UserAction.getUserId](#) ( )

6.25.1.3 void [ilearnrw.user.UserAction.setGameAction](#) ( [GameAction](#) [gameAction](#) )

6.25.1.4 void [ilearnrw.user.UserAction.setUserId](#) ( int [userId](#) )

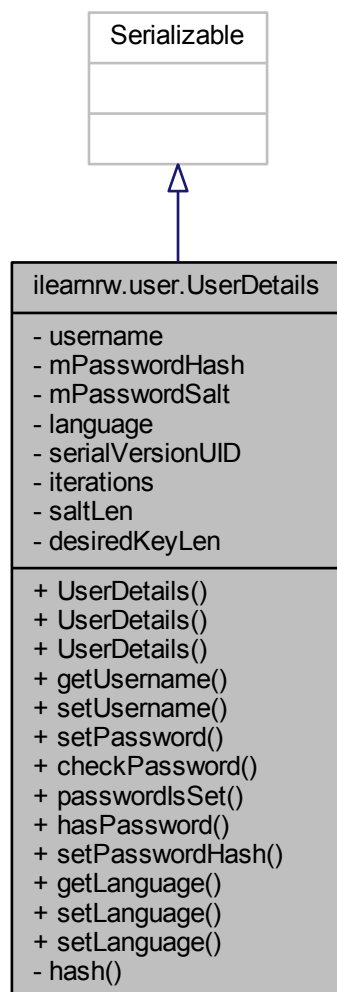
### 6.25.2 Member Data Documentation

6.25.2.1 [GameAction](#) [ilearnrw.user.UserAction.gameAction](#) [private]

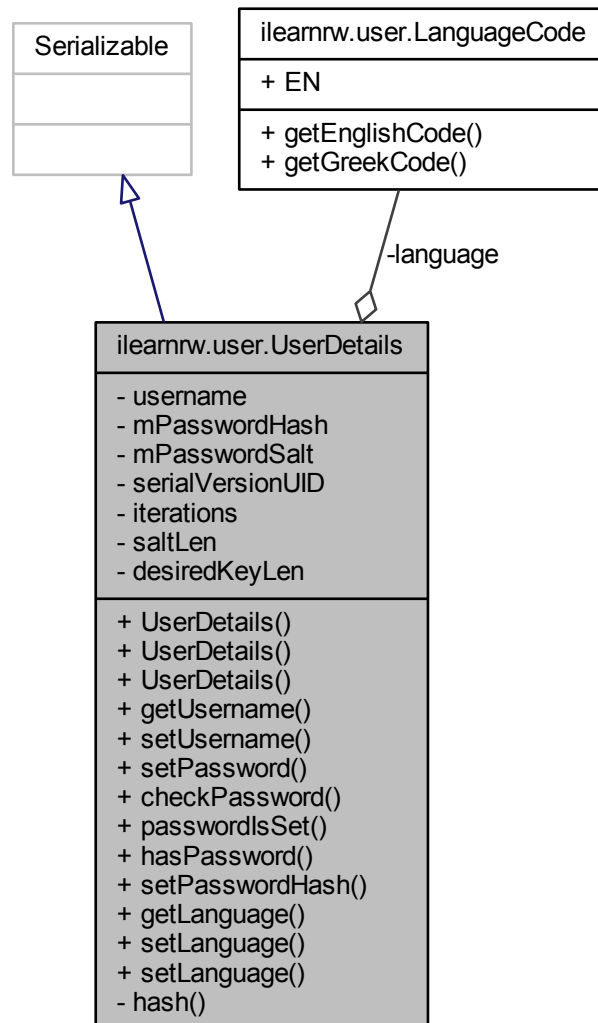
6.25.2.2 `int ilearnrw.user.UserAction.UserId` `[private]`

## 6.26 ilearnrw.user.UserDetails Class Reference

Inheritance diagram for ilearnrw.user.UserDetails:



Collaboration diagram for ilearnrw.user.UserDetails:



## Public Member Functions

- `UserDetails` (String `username`, int `password`, `LanguageCode` `language`)
- `UserDetails` (`LanguageCode` `language`)
- `UserDetails` (int `userId`)
- String `getUsername` ()
- void `setUsername` (String `username`)
- final void `setPassword` (String `plaintextPassword`)
- final boolean `checkPassword` (String `plaintextPassword`)
- boolean `passwordIsSet` ()
- boolean `hasPassword` ()
- final void `setPasswordHash` (byte[] `hash`, byte[] `salt`)
- `LanguageCode` `getLanguage` ()
- void `setLanguage` (`LanguageCode` `language`)
- void `setLanguage` (byte `languageCode`)

## Static Private Member Functions

- static byte[] [hash](#) (String plaintextPassword, byte[] salt) throws NoSuchAlgorithmException, InvalidKeySpecException

## Private Attributes

- String [username](#)
- byte[] [mPasswordHash](#)  
*Stores hash(plainTextPassword + mPasswordSalt)*
- byte[] [mPasswordSalt](#)  
*Stores the salt used to hash the password.*
- [LanguageCode](#) [language](#)

## Static Private Attributes

- static final long [serialVersionUID](#) = 1L
- static final int [iterations](#) = 10\*1024
- static final int [saltLen](#) = 32
- static final int [desiredKeyLen](#) = 256

### 6.26.1 Constructor & Destructor Documentation

6.26.1.1 `ilearnrw.user.UserDetails.UserDetails ( String username, int password, LanguageCode language )`

6.26.1.2 `ilearnrw.user.UserDetails.UserDetails ( LanguageCode language )`

6.26.1.3 `ilearnrw.user.UserDetails.UserDetails ( int userId )`

### 6.26.2 Member Function Documentation

6.26.2.1 `final boolean ilearnrw.user.UserDetails.checkPassword ( String plaintextPassword )`

6.26.2.2 `LanguageCode ilearnrw.user.UserDetails.getLanguage ( )`

6.26.2.3 `String ilearnrw.user.UserDetails.getUsername ( )`

6.26.2.4 `static byte [] ilearnrw.user.UserDetails.hash ( String plaintextPassword, byte[] salt ) throws  
NoSuchAlgorithmException, InvalidKeySpecException [static],[private]`

6.26.2.5 `boolean ilearnrw.user.UserDetails.hasPassword ( )`

6.26.2.6 `boolean ilearnrw.user.UserDetails.passwordIsSet ( )`

6.26.2.7 `void ilearnrw.user.UserDetails.setLanguage ( LanguageCode language )`

#### Parameters

<i>the</i>	language code as a enum item
------------	------------------------------

6.26.2.8 `void ilearnrw.user.UserDetails.setLanguage ( byte languageCode )`

## Parameters

<i>the</i>	language code as a enum item
------------	------------------------------

6.26.2.9    `final void ilearnrw.user.UserDetails.setPassword ( String plaintextPassword )`

6.26.2.10   `final void ilearnrw.user.UserDetails.setPasswordHash ( byte[] hash, byte[] salt )`

6.26.2.11   `void ilearnrw.user.UserDetails.setUsername ( String username )`

### 6.26.3 Member Data Documentation

6.26.3.1    `final int ilearnrw.user.UserDetails.desiredKeyLen = 256`    `[static], [private]`

6.26.3.2    `final int ilearnrw.user.UserDetails.iterations = 10*1024`    `[static], [private]`

6.26.3.3    `LanguageCode ilearnrw.user.UserDetails.language`    `[private]`

6.26.3.4    `byte [] ilearnrw.user.UserDetails.mPasswordHash`    `[private]`

Stores hash(plainTextPassword + mPasswordSalt)

#### Note

In the future we might want to change this to store the value in a base64 encoded String;

6.26.3.5    `byte [] ilearnrw.user.UserDetails.mPasswordSalt`    `[private]`

Stores the salt used to hash the password.

#### Note

In the future we might want to change this to store the value in a base64 encoded String;

6.26.3.6    `final int ilearnrw.user.UserDetails.saltLen = 32`    `[static], [private]`

6.26.3.7    `final long ilearnrw.user.UserDetails.serialVersionUID = 1L`    `[static], [private]`

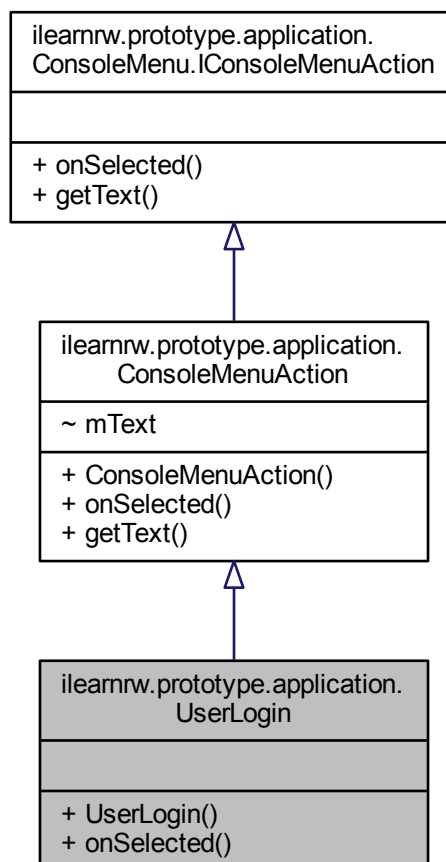
6.26.3.8    `String ilearnrw.user.UserDetails.username`    `[private]`

**Todo** Something to think about is whether this should be case-sensitive and if unicode should be supported. Unicode can create some headaches, there was a blog post about this recently (of which i only read the headline) see: <http://labs.spotify.com/2013/06/18/creative-usernames/>

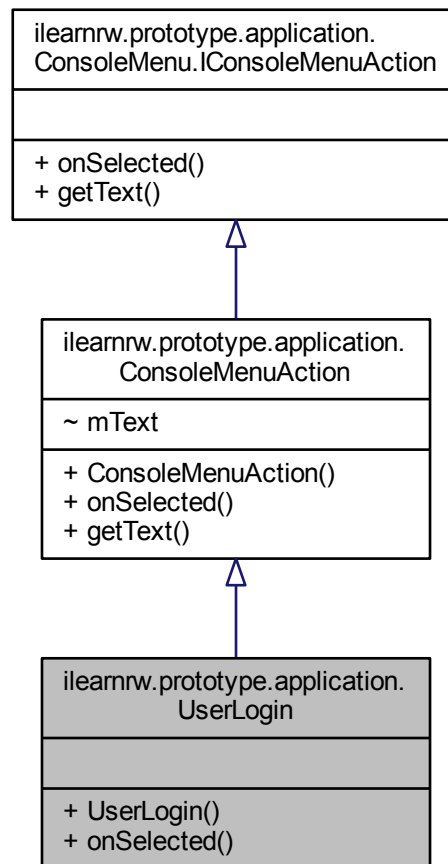
## 6.27 ilearnrw.prototype.application.UserLogin Class Reference

Main menu of a logged in user.

Inheritance diagram for ilearnrw.prototype.application.UserLogin:



Collaboration diagram for ilearnrw.prototype.application.UserLogin:



## Public Member Functions

- [UserLogin](#) (String text)
- [EConsoleMenuActionResult onSelected](#) ([ConsoleMenu](#) menu)

## Additional Inherited Members

### 6.27.1 Detailed Description

Main menu of a logged in user.

### 6.27.2 Constructor & Destructor Documentation

6.27.2.1 `ilearnrw.prototype.application.UserLogin.UserLogin ( String text )`

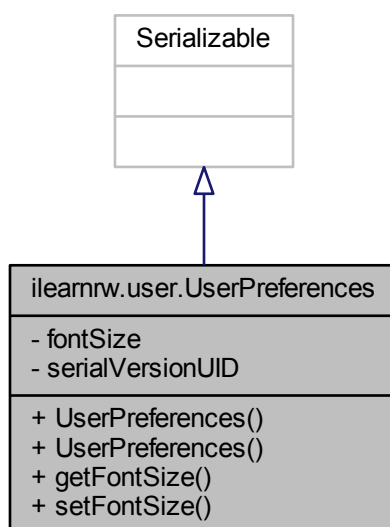
### 6.27.3 Member Function Documentation

6.27.3.1 **EConsoleMenuActionResult** `ilearnrw.prototype.application.UserLogin.onSelected ( ConsoleMenu menu )`

Implements [ilearnrw.prototype.application.ConsoleMenu.IConsoleMenuAction](#).

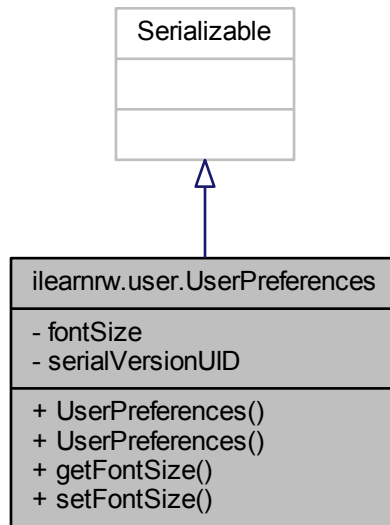
## 6.28 ilearnrw.user.UserPreferences Class Reference

Inheritance diagram for `ilearnrw.user.UserPreferences`:





Collaboration diagram for ilearnrw.user.UserPreferences:



### Public Member Functions

- [UserPreferences](#) ()
- [UserPreferences](#) (int [fontSize](#))
- int [getFontSize](#) ()
- void [setFontSize](#) (int [fontSize](#))

### Private Attributes

- int [fontSize](#)

### Static Private Attributes

- static final long [serialVersionUID](#) = 1L

## 6.28.1 Constructor & Destructor Documentation

6.28.1.1 `ilearnrw.user.UserPreferences.UserPreferences ( )`

6.28.1.2 `ilearnrw.user.UserPreferences.UserPreferences ( int fontSize )`

## 6.28.2 Member Function Documentation

6.28.2.1 `int ilearnrw.user.UserPreferences.getFontSize ( )`

6.28.2.2 `void ilearnrw.user.UserPreferences.setFontSize ( int fontSize )`

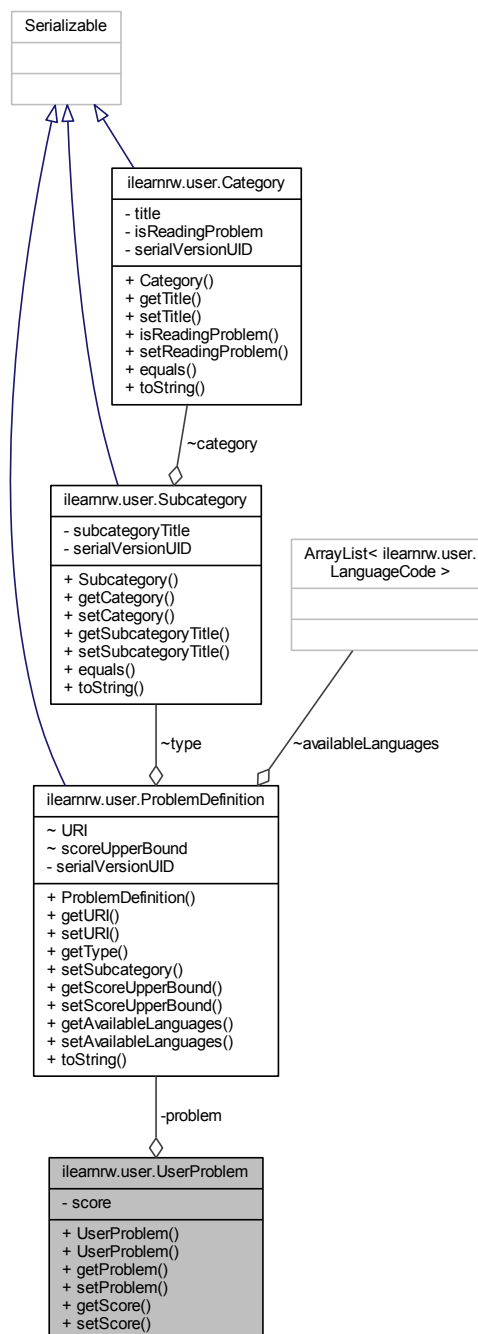
### 6.28.3 Member Data Documentation

6.28.3.1 `int ilearnrw.user.UserPreferences.fontSize` `[private]`

6.28.3.2 `final long ilearnrw.user.UserPreferences.serialVersionUID = 1L` `[static], [private]`

## 6.29 ilearnrw.user.UserProblem Class Reference

Collaboration diagram for `ilearnrw.user.UserProblem`:



## Public Member Functions

- [UserProblem](#) ([ProblemDefinition](#) *problem*, int *score*)
- [UserProblem](#) ([ProblemDefinition](#) *problem*)
- [ProblemDefinition](#) [getProblem](#) ()
- void [setProblem](#) ([ProblemDefinition](#) *problem*)
- int [getScore](#) ()
- void [setScore](#) (int *score*)

## Private Attributes

- [ProblemDefinition](#) *problem*
- int *score*

### 6.29.1 Constructor & Destructor Documentation

6.29.1.1 `ilearnrw.user.UserProblem.UserProblem ( ProblemDefinition problem, int score )`

6.29.1.2 `ilearnrw.user.UserProblem.UserProblem ( ProblemDefinition problem )`

### 6.29.2 Member Function Documentation

6.29.2.1 `ProblemDefinition ilearnrw.user.UserProblem.getProblem ( )`

#### Returns

the [ProblemDefintion](#) of the user's problem

6.29.2.2 `int ilearnrw.user.UserProblem.getScore ( )`

#### Returns

the frequency that the user does errors related to this problem

6.29.2.3 `void ilearnrw.user.UserProblem.setProblem ( ProblemDefinition problem )`

6.29.2.4 `void ilearnrw.user.UserProblem.setScore ( int score )`

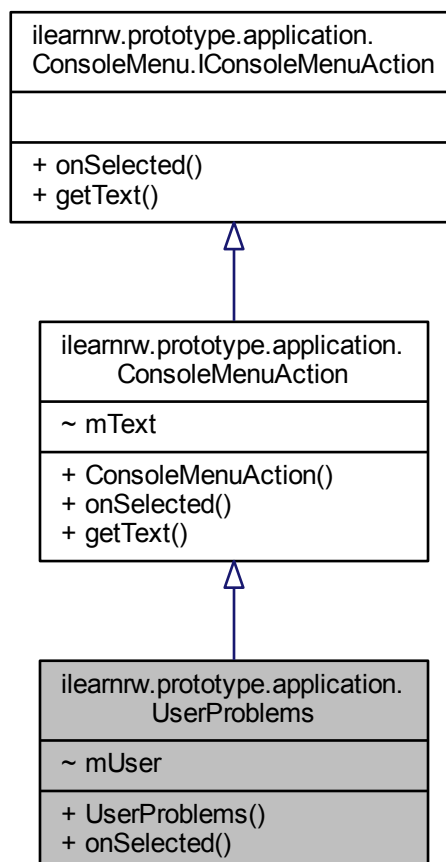
### 6.29.3 Member Data Documentation

6.29.3.1 `ProblemDefinition ilearnrw.user.UserProblem.problem [private]`

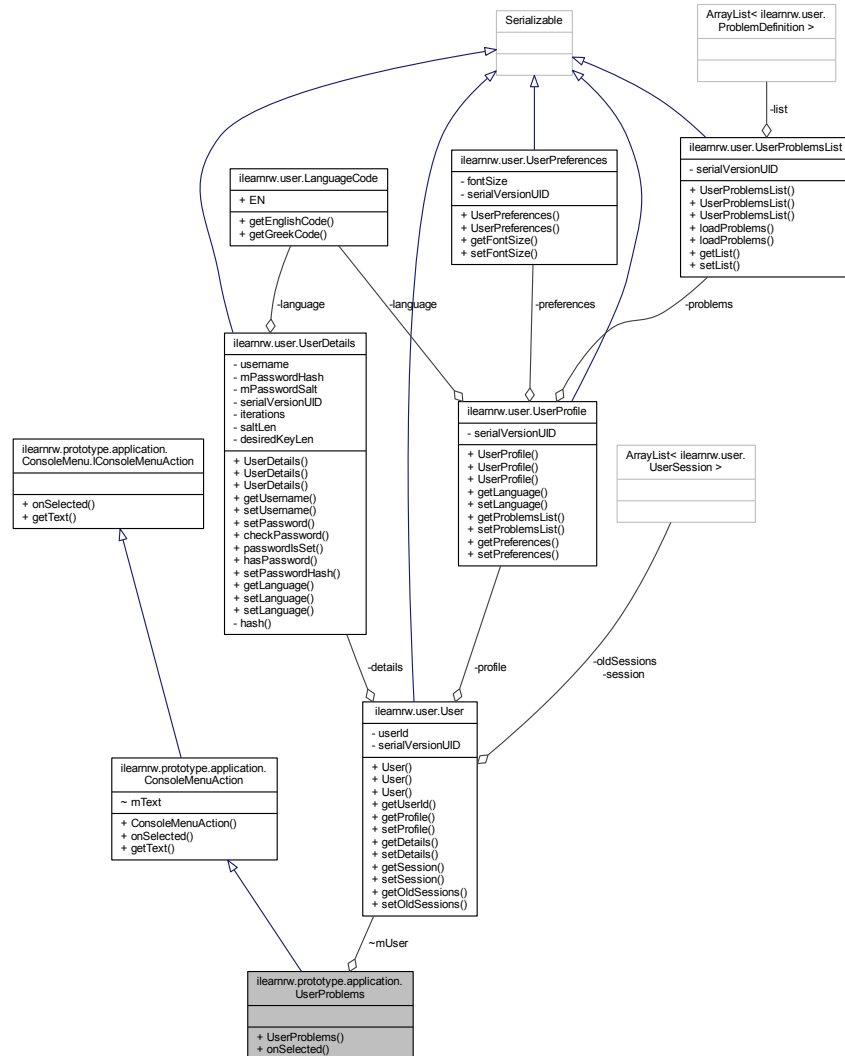
6.29.3.2 `int ilearnrw.user.UserProblem.score [private]`

### 6.30 ilearnrw.prototype.application.UserProblems Class Reference

Inheritance diagram for ilearnrw.prototype.application.UserProblems:



Collaboration diagram for ilearnrw.prototype.application.UserProblems:



## Public Member Functions

- [UserProblems](#) (String text, [User](#) user)
- [EConsoleMenuActionResult](#) onSelected ([ConsoleMenu](#) menu)

## Package Attributes

- [User](#) mUser = null

### 6.30.1 Constructor & Destructor Documentation

#### 6.30.1.1 ilearnrw.prototype.application.UserProblems.UserProblems ( String text, User user )

### 6.30.2 Member Function Documentation

### 6.30.2.1 EConsoleMenuActionResult ilearnrw.prototype.application.UserProblems.onSelected ( ConsoleMenu *menu* )

Should display a menu where you can:

- List problem definitions
- Add problem definitions to a profile from the ProblemDefinitionIndex.
- Remove problem definitions

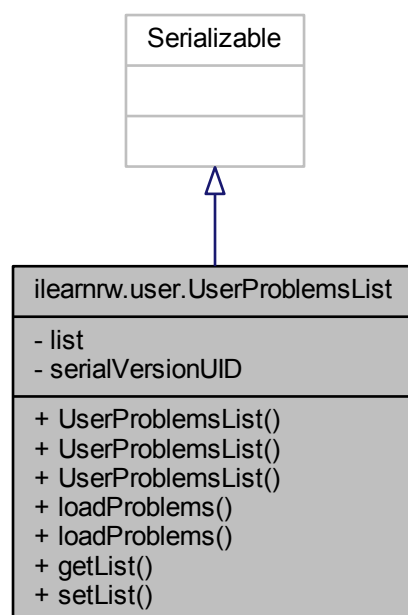
Implements [ilearnrw.prototype.application.ConsoleMenu.IConsoleMenuAction](#).

## 6.30.3 Member Data Documentation

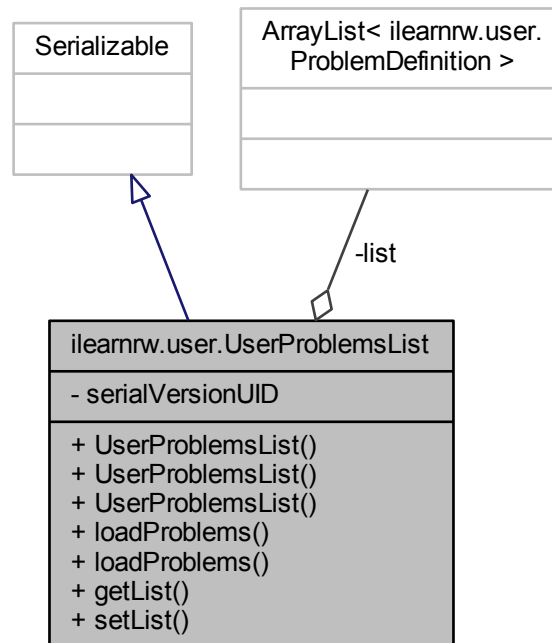
### 6.30.3.1 User ilearnrw.prototype.application.UserProblems.mUser = null [package]

## 6.31 ilearnrw.user.UserProblemsList Class Reference

Inheritance diagram for ilearnrw.user.UserProblemsList:



Collaboration diagram for ilearnrw.user.UserProblemsList:



## Public Member Functions

- [UserProblemsList](#) (int *userId*)
- [UserProblemsList](#) ([LanguageCode](#) *lc*)
- [UserProblemsList](#) ([ArrayList< ProblemDefinition >](#) *list*)
- void [loadProblems](#) ([LanguageCode](#) *lc*)
- void [loadProblems](#) (int *userId*)
- [ArrayList< ProblemDefinition >](#) [getList](#) ()
- void [setList](#) ([ArrayList< ProblemDefinition >](#) *list*)

## Private Attributes

- [ArrayList< ProblemDefinition >](#) *list*

## Static Private Attributes

- static final long [serialVersionUID](#) = 1L

### 6.31.1 Constructor & Destructor Documentation

6.31.1.1 `ilearnrw.user.UserProblemsList.UserProblemsList ( int userId )`

6.31.1.2 `ilearnrw.user.UserProblemsList.UserProblemsList ( LanguageCode lc )`

6.31.1.3 `ilearnrw.user.UserProblemsList.UserProblemsList ( ArrayList< ProblemDefinition > list )`

## 6.31.2 Member Function Documentation

6.31.2.1 `ArrayList<ProblemDefinition> ilearnrw.user.UserProblemsList.getList ( )`

6.31.2.2 `void ilearnrw.user.UserProblemsList.loadProblems ( LanguageCode lc )`

6.31.2.3 `void ilearnrw.user.UserProblemsList.loadProblems ( int userId )`

6.31.2.4 `void ilearnrw.user.UserProblemsList.setList ( ArrayList< ProblemDefinition > list )`

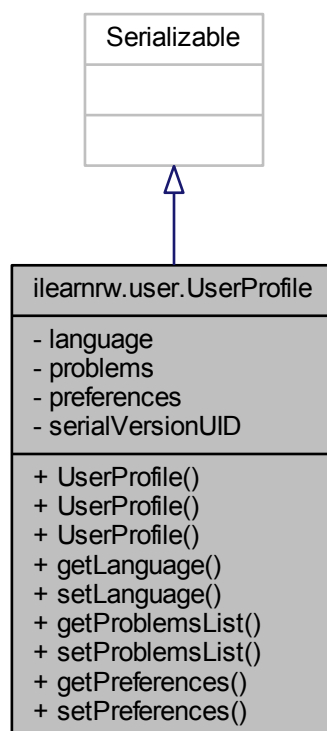
## 6.31.3 Member Data Documentation

6.31.3.1 `ArrayList<ProblemDefinition> ilearnrw.user.UserProblemsList.list [private]`

6.31.3.2 `final long ilearnrw.user.UserProblemsList.serialVersionUID = 1L [static],[private]`

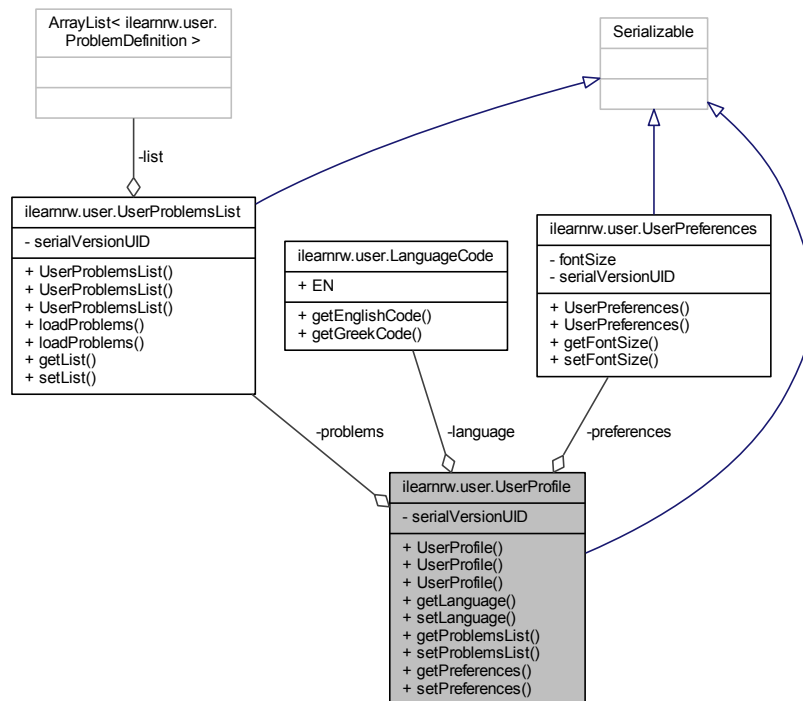
## 6.32 ilearnrw.user.UserProfile Class Reference

Inheritance diagram for `ilearnrw.user.UserProfile`:





Collaboration diagram for ilearnrw.user.UserProfile:



## Public Member Functions

- [UserProfile \(LanguageCode language, UserProblemsList problems, UserPreferences preferences\)](#)
- [UserProfile \(LanguageCode language\)](#)
- [UserProfile \(int userId\)](#)
- [LanguageCode getLanguage \(\)](#)
- [void setLanguage \(LanguageCode language\)](#)
- [UserProblemsList getProblemsList \(\)](#)
- [void setProblemsList \(UserProblemsList problems\)](#)
- [UserPreferences getPreferences \(\)](#)
- [void setPreferences \(UserPreferences preferences\)](#)

## Private Attributes

- [LanguageCode language](#)
- [UserProblemsList problems](#)
- [UserPreferences preferences](#)

## Static Private Attributes

- static final long [serialVersionUID](#) = 1L

### 6.32.1 Constructor & Destructor Documentation

6.32.1.1 `ilearnrw.user.UserProfile.UserProfile ( LanguageCode language, UserProblemsList problems, UserPreferences preferences )`

6.32.1.2 `ilearnrw.user.UserProfile.UserProfile ( LanguageCode language )`

6.32.1.3 `ilearnrw.user.UserProfile.UserProfile ( int userId )`

### 6.32.2 Member Function Documentation

6.32.2.1 `LanguageCode ilearnrw.user.UserProfile.getLanguage ( )`

6.32.2.2 `UserPreferences ilearnrw.user.UserProfile.getPreferences ( )`

6.32.2.3 `UserProblemsList ilearnrw.user.UserProfile.getProblemsList ( )`

6.32.2.4 `void ilearnrw.user.UserProfile.setLanguage ( LanguageCode language )`

6.32.2.5 `void ilearnrw.user.UserProfile.setPreferences ( UserPreferences preferences )`

6.32.2.6 `void ilearnrw.user.UserProfile.setProblemsList ( UserProblemsList problems )`

### 6.32.3 Member Data Documentation

6.32.3.1 `LanguageCode ilearnrw.user.UserProfile.language [private]`

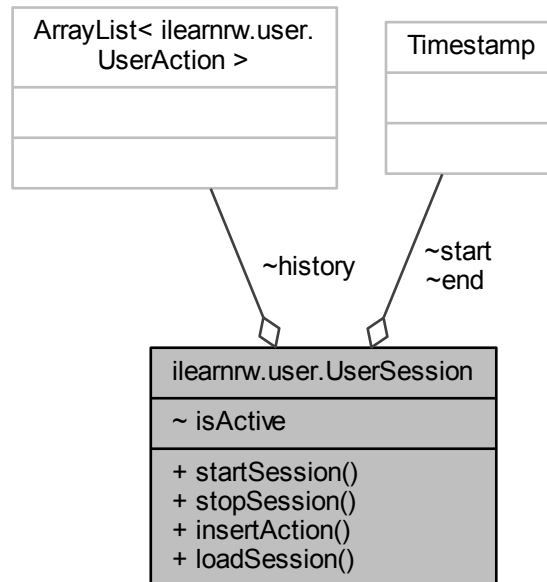
6.32.3.2 `UserPreferences ilearnrw.user.UserProfile.preferences [private]`

6.32.3.3 `UserProblemsList ilearnrw.user.UserProfile.problems [private]`

6.32.3.4 `final long ilearnrw.user.UserProfile.serialVersionUID = 1L [static],[private]`

## 6.33 ilearnrw.user.UserSession Class Reference

Collaboration diagram for ilearnrw.user.UserSession:



### Public Member Functions

- void [startSession](#) ()
- void [stopSession](#) ()
- void [insertAction](#) ([UserAction](#) x) throws [IlearnException](#)
- void [loadSession](#) (int [userId](#))

### Package Attributes

- boolean [isActive](#)
- Timestamp [start](#)
- Timestamp [end](#)
- ArrayList< [UserAction](#) > [history](#)

#### 6.33.1 Member Function Documentation

6.33.1.1 void [ilearnrw.user.UserSession.insertAction](#) ( [UserAction](#) x ) throws [IlearnException](#)

6.33.1.2 void [ilearnrw.user.UserSession.loadSession](#) ( int [userId](#) )

6.33.1.3 void [ilearnrw.user.UserSession.startSession](#) ( )

sets the timestamp and turns the boolean variable 'isActive' to true!

#### 6.33.1.4 void ilearnrw.user.UserSession.stopSession ( )

sets the timestamp and turns the boolean variable 'isActive' to true!

### 6.33.2 Member Data Documentation

#### 6.33.2.1 Timestamp ilearnrw.user.UserSession.end [package]

#### 6.33.2.2 ArrayList<UserAction> ilearnrw.user.UserSession.history [package]

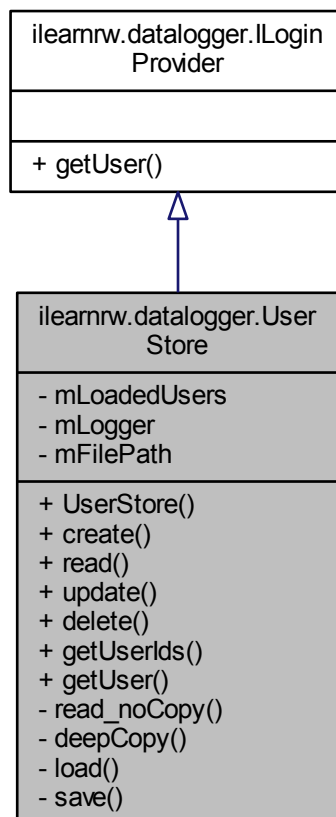
#### 6.33.2.3 boolean ilearnrw.user.UserSession.isActive [package]

#### 6.33.2.4 Timestamp ilearnrw.user.UserSession.start [package]

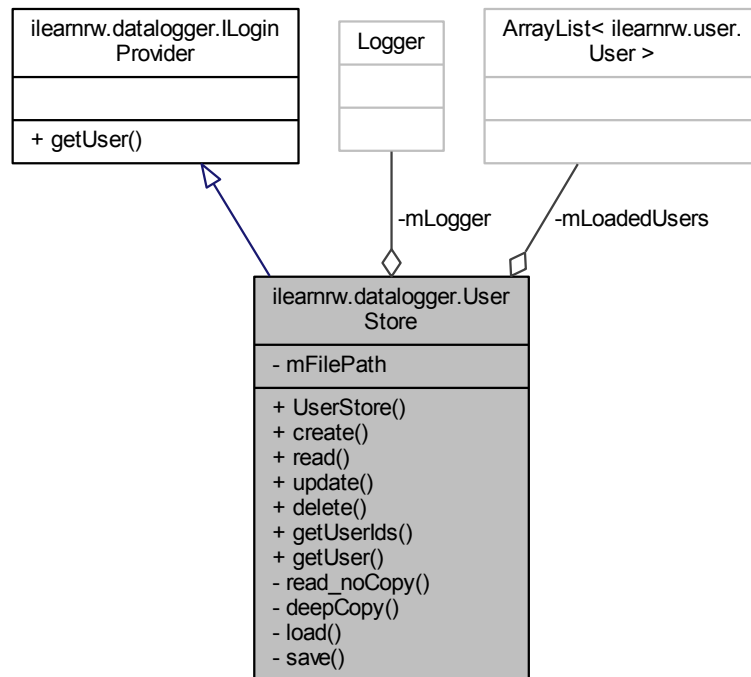
## 6.34 ilearnrw.datalogger.UserStore Class Reference

The [UserStore](#) is used to create and store user objects persistantly.

Inheritance diagram for ilearnrw.datalogger.UserStore:



Collaboration diagram for ilearnrw.datalogger.UserStore:



## Public Member Functions

- [UserStore](#) (String filePath)
- [User create](#) ()
- [User read](#) (int userId)
- [User update](#) ([User](#) user)
- boolean [delete](#) (int userId)
- LinkedList< Integer > [getUserIds](#) ()
- [User getUser](#) (String userName, String password)

*Authenticates a User.*

## Private Member Functions

- [User read\\_noCopy](#) (int userId)
  - [User deepCopy](#) ([User](#) user)
  - void [load](#) ()
- Loads the serialized objects in mFilePath into mLoadedUsers.*
- void [save](#) ()

*Serializes mLoadedUsers into mFilePath*

## Private Attributes

- ArrayList< [User](#) > [mLoadedUsers](#)
- Logger [mLogger](#)

## Static Private Attributes

- static String `mFilePath`

*The complete path to the file to read/write the user information from/to.*

### 6.34.1 Detailed Description

The `UserStore` is used to create and store user objects persistantly.

**Todo** Replace storage backed with SQL database.

#### Note

At the moment Java's built in serialization is used to store users to disk. This requires all objects to be stored to implement the `Serializable` interface.

Implements a CRUD interface for users and save/load functions to store the data persistantly.

### 6.34.2 Constructor & Destructor Documentation

#### 6.34.2.1 `ilearnrw.datalogger.UserStore.UserStore ( String filePath )`

##### Parameters

<code>filePath</code>	The full path to the file to use to store the user profiles etc.
-----------------------	--

### 6.34.3 Member Function Documentation

#### 6.34.3.1 `User ilearnrw.datalogger.UserStore.create ( )`

Create a new user.

Will generate a new `UserId` for the user.

##### Returns

A new `User` object.

#### 6.34.3.2 `User ilearnrw.datalogger.UserStore.deepCopy ( User user ) [private]`

Returns a copy of a `User` object

This is called from functions returning a user in this class since we do not want to give out the actual reference stored in the `_loadedUsers` list.

#### 6.34.3.3 `boolean ilearnrw.datalogger.UserStore.delete ( int userId )`

Deletes a user from the store.

## Parameters

<i>userId</i>	userId of the user to delete
---------------	------------------------------

## Returns

true if successful.

#### 6.34.3.4 User ilearnrw.datalogger.UserStore.getUser ( String *userName*, String *password* )

Authenticates a User.

Implements [ilearnrw.datalogger.ILoginProvider](#).

#### 6.34.3.5 LinkedList<Integer> ilearnrw.datalogger.UserStore.getUserIds ( )

## Returns

A list of all stored userid's.

#### 6.34.3.6 void ilearnrw.datalogger.UserStore.load ( ) [private]

Loads the serialized objects in `mFilePath` into `mLoadedUsers`.

#### 6.34.3.7 User ilearnrw.datalogger.UserStore.read ( int *userId* )

Reads a User from the [UserStore](#)

## Parameters

<i>userId</i>	The userId of the user to load
---------------	--------------------------------

## Returns

The user with `userId == userId`

#### 6.34.3.8 User ilearnrw.datalogger.UserStore.read\_noCopy ( int *userId* ) [private]

#### 6.34.3.9 void ilearnrw.datalogger.UserStore.save ( ) [private]

Serializes `mLoadedUsers` into `mFilePath`

#### 6.34.3.10 User ilearnrw.datalogger.UserStore.update ( User *user* )

Updates the user currently stored with the one supplied.

**Todo** Don't allow multiple users with the same username.

## Parameters

<i>user</i>	The new user copy to store.
-------------	-----------------------------

## Returns

A new copy of the user object.

#### 6.34.4 Member Data Documentation

**6.34.4.1** `String ilearnrw.datalogger.UserStore.mFilePath` `[static], [private]`

The complete path to the file to read/write the user information from/to.

**6.34.4.2** `ArrayList<User> ilearnrw.datalogger.UserStore.mLoadedUsers` `[private]`

**6.34.4.3** `Logger ilearnrw.datalogger.UserStore.mLogger` `[private]`



# Index

- addProblemDefinition
  - ilearnrw::datalogger::IProfileAccessUpdater, [29](#)
  - ilearnrw::datalogger::ProfileAccessUpdater, [42](#)
- availableLanguages
  - ilearnrw::user::ProblemDefinition, [38](#)
- Category
  - ilearnrw::user::Category, [12](#)
- category
  - ilearnrw::user::Subcategory, [49](#)
- checkPassword
  - ilearnrw::user::UserDetails, [56](#)
- ConsoleMenu
  - ilearnrw::prototype::application::ConsoleMenu, [14](#)
- ConsoleMenuAction
  - ilearnrw::prototype::application::ConsoleMenu-Action, [16](#)
- create
  - ilearnrw::datalogger::UserStore, [74](#)
- DIFFICULT\_WORD
  - ilearnrw::user::GameAction, [21](#)
- DatabaseManager
  - ilearnrw::prototype::application::DatabaseManager, [18](#)
- deepCopy
  - ilearnrw::datalogger::UserStore, [74](#)
- delete
  - ilearnrw::datalogger::UserStore, [74](#)
- desiredKeyLen
  - ilearnrw::user::UserDetails, [57](#)
- details
  - ilearnrw::user::User, [52](#)
- discardPendingChanges
  - ilearnrw::datalogger::IProfileAccessUpdater, [29](#)
  - ilearnrw::datalogger::ProfileAccessUpdater, [42](#)
- doModalMenu
  - ilearnrw::prototype::application::ConsoleMenu, [14](#)
- EN
  - ilearnrw::user::LanguageCode, [32](#)
- end
  - ilearnrw::user::UserSession, [72](#)
- equals
  - ilearnrw::user::Category, [13](#)
  - ilearnrw::user::Subcategory, [49](#)
- fontSize
  - ilearnrw::user::UserPreferences, [62](#)
- gDataLogger
  - ilearnrw::prototype::application::Program, [46](#)
- gameAction
  - ilearnrw::user::UserAction, [53](#)
- getAvailableLanguages
  - ilearnrw::user::ProblemDefinition, [38](#)
- getCategory
  - ilearnrw::user::Subcategory, [49](#)
- getDetails
  - ilearnrw::user::User, [52](#)
- getEnglishCode
  - ilearnrw::user::LanguageCode, [32](#)
- getFontSize
  - ilearnrw::user::UserPreferences, [61](#)
- getGameAction
  - ilearnrw::user::UserAction, [53](#)
- getGreekCode
  - ilearnrw::user::LanguageCode, [32](#)
- getLanguage
  - ilearnrw::user::UserDetails, [56](#)
  - ilearnrw::user::UserProfile, [70](#)
- getList
  - ilearnrw::user::UserProblemsList, [68](#)
- getLoginDetails
  - ilearnrw::prototype::application::Program, [46](#)
- getLoginProvider
  - ilearnrw::datalogger::IDataLogger, [23](#)
- getMessage
  - ilearnrw::datalogger::IProfileAccessUpdater::PendingChangesAvailable, [34](#)
  - ilearnrw::datalogger::IProfileAccessUpdater::PermissionException, [35](#)
  - ilearnrw::user::IlearnException, [25](#)
- getOldSessions
  - ilearnrw::user::User, [52](#)
- getPreferences
  - ilearnrw::user::UserProfile, [70](#)
- getProblem
  - ilearnrw::user::UserProblem, [63](#)
- getProblemsByCategory
  - ilearnrw::user::ProblemDefinitionIndex, [39](#)
- getProblemsByLanguage
  - ilearnrw::user::ProblemDefinitionIndex, [39](#)
- getProblemsList
  - ilearnrw::user::UserProfile, [70](#)
- getProfile
  - ilearnrw::user::User, [52](#)
- getProfileAccessUpdater
  - ilearnrw::datalogger::IDataLogger, [23](#)
- getScore

- ilearnrw::user::UserProblem, 63
- getScoreUpperBound
  - ilearnrw::user::ProblemDefinition, 38
- getSession
  - ilearnrw::user::User, 52
- getStringArg
  - ilearnrw::prototype::application::Program, 46
- getSubcategoryTitle
  - ilearnrw::user::Subcategory, 49
- getText
  - ilearnrw::prototype::application::ConsoleMenu::IConsoleMenuAction, 22
  - ilearnrw::prototype::application::ConsoleMenuAction, 16
- getTitle
  - ilearnrw::user::Category, 13
- getType
  - ilearnrw::user::ProblemDefinition, 38
- getURI
  - ilearnrw::user::ProblemDefinition, 38
- getUser
  - ilearnrw::datalogger::ILoginProvider, 26
  - ilearnrw::datalogger::UserStore, 75
- getUserId
  - ilearnrw::user::User, 52
  - ilearnrw::user::UserAction, 53
- getUserIds
  - ilearnrw::datalogger::UserStore, 75
- getUserStore
  - ilearnrw::datalogger::DataLogger, 20
- getUsername
  - ilearnrw::user::UserDetails, 56
- hasPassword
  - ilearnrw::user::UserDetails, 56
- hasPendingChanges
  - ilearnrw::datalogger::IProfileAccessUpdater, 29
  - ilearnrw::datalogger::ProfileAccessUpdater, 42
- hash
  - ilearnrw::user::UserDetails, 56
- history
  - ilearnrw::user::UserSession, 72
- ILearnException
  - ilearnrw::user::ILearnException, 25
- ilearnrw, 9
- ilearnrw.datalogger, 9
- ilearnrw.datalogger.DataLogger, 19
- ilearnrw.datalogger.IDataLogger, 23
- ilearnrw.datalogger.ILoginProvider, 25
- ilearnrw.datalogger.IProfileAccessUpdater, 26
- ilearnrw.datalogger.IProfileAccessUpdater.PendingChangesAvailable, 32
- ilearnrw.datalogger.IProfileAccessUpdater.PermissionException, 34
- ilearnrw.datalogger.LoginDetails, 32
- ilearnrw.datalogger.ProfileAccessUpdater, 40
- ilearnrw.datalogger.UserStore, 72
- ilearnrw.prototype, 10
- ilearnrw.prototype.application, 10
- ilearnrw.prototype.application.ConsoleMenu, 13
- ilearnrw.prototype.application.ConsoleMenu.EConsoleMenuActionResult, 20
- ilearnrw.prototype.application.ConsoleMenu.IConsoleMenuAction, 22
- ilearnrw.prototype.application.ConsoleMenuAction, 15
- ilearnrw.prototype.application.DatabaseManager, 17
- ilearnrw.prototype.application.DatabaseManager.ISelectUser, 31
- ilearnrw.prototype.application.Program, 45
- ilearnrw.prototype.application.UserLogin, 57
- ilearnrw.prototype.application.UserProblems, 64
- ilearnrw.user, 10
- ilearnrw.user.Category, 11
- ilearnrw.user.GameAction, 21
- ilearnrw.user.ILearnException, 24
- ilearnrw.user.LanguageCode, 31
- ilearnrw.user.ProblemDefinition, 36
- ilearnrw.user.ProblemDefinitionIndex, 39
- ilearnrw.user.Subcategory, 47
- ilearnrw.user.Tester, 49
- ilearnrw.user.User, 50
- ilearnrw.user.UserAction, 53
- ilearnrw.user.UserDetails, 54
- ilearnrw.user.UserPreferences, 60
- ilearnrw.user.UserProblem, 62
- ilearnrw.user.UserProblemsList, 66
- ilearnrw.user.UserProfile, 68
- ilearnrw.user.UserSession, 71
- ilearnrw::datalogger::DataLogger
  - getUserStore, 20
  - loadUserStore, 20
  - mUserStore, 20
- ilearnrw::datalogger::IDataLogger
  - getLoginProvider, 23
  - getProfileAccessUpdater, 23
- ilearnrw::datalogger::ILoginProvider
  - getUser, 26
- ilearnrw::datalogger::IProfileAccessUpdater
  - addProblemDefinition, 29
  - discardPendingChanges, 29
  - hasPendingChanges, 29
  - registerUserSession, 29
  - removeProblemDefinition, 29
  - setCurrentProfile, 29
  - setPreference, 30
  - setUserDetail, 30
  - setUserLanguage, 30
  - writePendingChanges, 30
- ilearnrw::datalogger::IProfileAccessUpdater::PendingChangesAvailable
  - getMessage, 34
  - PendingChangesAvailable, 34
  - serialVersionUID, 34
- ilearnrw::datalogger::IProfileAccessUpdater::PermissionException
  - getMessage, 35

- PermissionException, 35
  - serialVersionUID, 35
- ilearnrw::datalogger::LoginDetails
  - Password, 32
  - Username, 32
- ilearnrw::datalogger::ProfileAccessUpdater
  - addProblemDefinition, 42
  - discardPendingChanges, 42
  - hasPendingChanges, 42
  - mIsDirty, 44
  - mMarker, 44
  - mUser, 44
  - registerUserSession, 42
  - removeProblemDefinition, 43
  - setCurrentProfile, 43
  - setDirty, 43
  - setPreference, 43
  - setUserDetail, 43
  - setUserLanguage, 43
  - writePendingChanges, 43
- ilearnrw::datalogger::UserStore
  - create, 74
  - deepCopy, 74
  - delete, 74
  - getUser, 75
  - getUserIds, 75
  - load, 75
  - mFilePath, 76
  - mLoadedUsers, 76
  - mLogger, 76
  - read, 75
  - read\_noCopy, 75
  - save, 75
  - update, 75
  - UserStore, 74
- ilearnrw::prototype::application::ConsoleMenu
  - ConsoleMenu, 14
  - doModalMenu, 14
  - in, 14
  - mIn, 14
  - mMenuActions, 14
  - mMenuTitle, 14
  - mOut, 14
  - mScanner, 14
  - out, 14
  - showMenu, 14
  - subMenu, 14
- ilearnrw::prototype::application::ConsoleMenu::E-ConsoleMenuActionResult
  - menuBreak, 21
  - showThisMenuAgain, 21
- ilearnrw::prototype::application::ConsoleMenu::IConsoleMenuAction
  - getText, 22
  - onSelected, 22
- ilearnrw::prototype::application::ConsoleMenuAction
  - ConsoleMenuAction, 16
  - getText, 16
  - mText, 16
  - onSelected, 16
- ilearnrw::prototype::application::DatabaseManager
  - DatabaseManager, 18
  - onSelected, 18
- ilearnrw::prototype::application::DatabaseManager::I-SelectUser
  - selectUser, 31
- ilearnrw::prototype::application::Program
  - gDataLogger, 46
  - getLoginDetails, 46
  - getStringArg, 46
  - main, 46
  - sWelcomeMessage, 46
- ilearnrw::prototype::application::UserLogin
  - onSelected, 59
  - UserLogin, 59
- ilearnrw::prototype::application::UserProblems
  - mUser, 66
  - onSelected, 65
  - UserProblems, 65
- ilearnrw::user::Category
  - Category, 12
  - equals, 13
  - getTitle, 13
  - isReadingProblem, 13
  - serialVersionUID, 13
  - setReadingProblem, 13
  - setTitle, 13
  - title, 13
  - toString, 13
- ilearnrw::user::GameAction
  - DIFFICULT\_WORD, 21
- ilearnrw::user::IlearnException
  - getMessage, 25
  - IlearnException, 25
- ilearnrw::user::LanguageCode
  - EN, 32
  - getEnglishCode, 32
  - getGreekCode, 32
- ilearnrw::user::ProblemDefinition
  - availableLanguages, 38
  - getAvailableLanguages, 38
  - getScoreUpperBound, 38
  - getType, 38
  - getURI, 38
  - ProblemDefinition, 38
  - scoreUpperBound, 38
  - serialVersionUID, 38
  - setAvailableLanguages, 38
  - setScoreUpperBound, 38
  - setSubcategory, 38
  - setURI, 38
  - toString, 38
  - type, 38
  - URI, 38
- ilearnrw::user::ProblemDefinitionIndex
  - getProblemsByCategory, 39

- getProblemsByLanguage, 39
- index, 39
- ilearnrw::user::Subcategory
  - category, 49
  - equals, 49
  - getCategory, 49
  - getSubcategoryTitle, 49
  - serialVersionUID, 49
  - setCategory, 49
  - setSubcategoryTitle, 49
  - Subcategory, 49
  - subcategoryTitle, 49
  - toString, 49
- ilearnrw::user::Tester
  - main, 50
- ilearnrw::user::User
  - details, 52
  - getDetails, 52
  - getOldSessions, 52
  - getProfile, 52
  - getSession, 52
  - getUserId, 52
  - oldSessions, 52
  - profile, 52
  - serialVersionUID, 52
  - session, 52
  - setDetails, 52
  - setOldSessions, 52
  - setProfile, 52
  - setSession, 52
  - User, 52
  - userId, 52
- ilearnrw::user::UserAction
  - gameAction, 53
  - getGameAction, 53
  - getUserId, 53
  - setGameAction, 53
  - setUserId, 53
  - UserId, 53
- ilearnrw::user::UserDetails
  - checkPassword, 56
  - desiredKeyLen, 57
  - getLanguage, 56
  - getUsername, 56
  - hasPassword, 56
  - hash, 56
  - iterations, 57
  - language, 57
  - mPasswordHash, 57
  - mPasswordSalt, 57
  - passwordIsSet, 56
  - saltLen, 57
  - serialVersionUID, 57
  - setLanguage, 56
  - setPassword, 57
  - setPasswordHash, 57
  - setUsername, 57
  - UserDetails, 56
  - username, 57
- ilearnrw::user::UserPreferences
  - fontSize, 62
  - getFontSize, 61
  - serialVersionUID, 62
  - setFontSize, 61
  - UserPreferences, 61
- ilearnrw::user::UserProblem
  - getProblem, 63
  - getScore, 63
  - problem, 63
  - score, 63
  - setProblem, 63
  - setScore, 63
  - UserProblem, 63
- ilearnrw::user::UserProblemsList
  - getList, 68
  - list, 68
  - loadProblems, 68
  - serialVersionUID, 68
  - setList, 68
  - UserProblemsList, 67
- ilearnrw::user::UserProfile
  - getLanguage, 70
  - getPreferences, 70
  - getProblemsList, 70
  - language, 70
  - preferences, 70
  - problems, 70
  - serialVersionUID, 70
  - setLanguage, 70
  - setPreferences, 70
  - setProblemsList, 70
  - UserProfile, 70
- ilearnrw::user::UserSession
  - end, 72
  - history, 72
  - insertAction, 71
  - isActive, 72
  - loadSession, 71
  - start, 72
  - startSession, 71
  - stopSession, 71
- in
  - ilearnrw::prototype::application::ConsoleMenu, 14
- index
  - ilearnrw::user::ProblemDefinitionIndex, 39
- insertAction
  - ilearnrw::user::UserSession, 71
- isActive
  - ilearnrw::user::UserSession, 72
- isReadingProblem
  - ilearnrw::user::Category, 13
- iterations
  - ilearnrw::user::UserDetails, 57
- language
  - ilearnrw::user::UserDetails, 57
  - ilearnrw::user::UserProfile, 70

- list
  - ilearnrw::user::UserProblemsList, 68
- load
  - ilearnrw::datalogger::UserStore, 75
- loadProblems
  - ilearnrw::user::UserProblemsList, 68
- loadSession
  - ilearnrw::user::UserSession, 71
- loadUserStore
  - ilearnrw::datalogger::DataLogger, 20
- mFilePath
  - ilearnrw::datalogger::UserStore, 76
- mIn
  - ilearnrw::prototype::application::ConsoleMenu, 14
- mIsDirty
  - ilearnrw::datalogger::ProfileAccessUpdater, 44
- mLoadedUsers
  - ilearnrw::datalogger::UserStore, 76
- mLogger
  - ilearnrw::datalogger::UserStore, 76
- mMarker
  - ilearnrw::datalogger::ProfileAccessUpdater, 44
- mMenuActions
  - ilearnrw::prototype::application::ConsoleMenu, 14
- mMenuTitle
  - ilearnrw::prototype::application::ConsoleMenu, 14
- mOut
  - ilearnrw::prototype::application::ConsoleMenu, 14
- mPasswordHash
  - ilearnrw::user::UserDetails, 57
- mPasswordSalt
  - ilearnrw::user::UserDetails, 57
- mScanner
  - ilearnrw::prototype::application::ConsoleMenu, 14
- mText
  - ilearnrw::prototype::application::ConsoleMenu-Action, 16
- mUser
  - ilearnrw::datalogger::ProfileAccessUpdater, 44
  - ilearnrw::prototype::application::UserProblems, 66
- mUserStore
  - ilearnrw::datalogger::DataLogger, 20
- main
  - ilearnrw::prototype::application::Program, 46
  - ilearnrw::user::Tester, 50
- menuBreak
  - ilearnrw::prototype::application::ConsoleMenu::E-ConsoleMenuActionResult, 21
- oldSessions
  - ilearnrw::user::User, 52
- onSelected
  - ilearnrw::prototype::application::ConsoleMenu::I-ConsoleMenuAction, 22
  - ilearnrw::prototype::application::ConsoleMenu-Action, 16
  - ilearnrw::prototype::application::DatabaseManager, 18
- ilearnrw::prototype::application::UserLogin, 59
- ilearnrw::prototype::application::UserProblems, 65
- out
  - ilearnrw::prototype::application::ConsoleMenu, 14
- Password
  - ilearnrw::datalogger::LoginDetails, 32
- passwordIsSet
  - ilearnrw::user::UserDetails, 56
- PendingChangesAvailable
  - ilearnrw::datalogger::IProfileAccessUpdater::PendingChangesAvailable, 34
- PermissionException
  - ilearnrw::datalogger::IProfileAccessUpdater::PermissionException, 35
- preferences
  - ilearnrw::user::UserProfile, 70
- problem
  - ilearnrw::user::UserProblem, 63
- ProblemDefinition
  - ilearnrw::user::ProblemDefinition, 38
- problems
  - ilearnrw::user::UserProfile, 70
- profile
  - ilearnrw::user::User, 52
- read
  - ilearnrw::datalogger::UserStore, 75
- read\_noCopy
  - ilearnrw::datalogger::UserStore, 75
- registerUserSession
  - ilearnrw::datalogger::IProfileAccessUpdater, 29
  - ilearnrw::datalogger::ProfileAccessUpdater, 42
- removeProblemDefinition
  - ilearnrw::datalogger::IProfileAccessUpdater, 29
  - ilearnrw::datalogger::ProfileAccessUpdater, 43
- sWelcomeMessage
  - ilearnrw::prototype::application::Program, 46
- saltLen
  - ilearnrw::user::UserDetails, 57
- save
  - ilearnrw::datalogger::UserStore, 75
- score
  - ilearnrw::user::UserProblem, 63
- scoreUpperBound
  - ilearnrw::user::ProblemDefinition, 38
- selectUser
  - ilearnrw::prototype::application::DatabaseManager::ISelectUser, 31
- serialVersionUID
  - ilearnrw::datalogger::IProfileAccessUpdater::PendingChangesAvailable, 34
  - ilearnrw::datalogger::IProfileAccessUpdater::PermissionException, 35
  - ilearnrw::user::Category, 13
  - ilearnrw::user::ProblemDefinition, 38
  - ilearnrw::user::Subcategory, 49
  - ilearnrw::user::User, 52

- ilearnrw::user::UserDetails, 57
  - ilearnrw::user::UserPreferences, 62
  - ilearnrw::user::UserProblemsList, 68
  - ilearnrw::user::UserProfile, 70
- session
  - ilearnrw::user::User, 52
- setAvailableLanguages
  - ilearnrw::user::ProblemDefinition, 38
- setCategory
  - ilearnrw::user::Subcategory, 49
- setCurrentProfile
  - ilearnrw::datalogger::IProfileAccessUpdater, 29
  - ilearnrw::datalogger::ProfileAccessUpdater, 43
- setDetails
  - ilearnrw::user::User, 52
- setDirty
  - ilearnrw::datalogger::ProfileAccessUpdater, 43
- setFontSize
  - ilearnrw::user::UserPreferences, 61
- setGameAction
  - ilearnrw::user::UserAction, 53
- setLanguage
  - ilearnrw::user::UserDetails, 56
  - ilearnrw::user::UserProfile, 70
- setList
  - ilearnrw::user::UserProblemsList, 68
- setOldSessions
  - ilearnrw::user::User, 52
- setPassword
  - ilearnrw::user::UserDetails, 57
- setPasswordHash
  - ilearnrw::user::UserDetails, 57
- setPreference
  - ilearnrw::datalogger::IProfileAccessUpdater, 30
  - ilearnrw::datalogger::ProfileAccessUpdater, 43
- setPreferences
  - ilearnrw::user::UserProfile, 70
- setProblem
  - ilearnrw::user::UserProblem, 63
- setProblemsList
  - ilearnrw::user::UserProfile, 70
- setProfile
  - ilearnrw::user::User, 52
- setReadingProblem
  - ilearnrw::user::Category, 13
- setScore
  - ilearnrw::user::UserProblem, 63
- setScoreUpperBound
  - ilearnrw::user::ProblemDefinition, 38
- setSession
  - ilearnrw::user::User, 52
- setSubcategory
  - ilearnrw::user::ProblemDefinition, 38
- setSubcategoryTitle
  - ilearnrw::user::Subcategory, 49
- setTitle
  - ilearnrw::user::Category, 13
- setURI
  - ilearnrw::user::ProblemDefinition, 38
- setUserDetail
  - ilearnrw::datalogger::IProfileAccessUpdater, 30
  - ilearnrw::datalogger::ProfileAccessUpdater, 43
- setUserId
  - ilearnrw::user::UserAction, 53
- setUserLanguage
  - ilearnrw::datalogger::IProfileAccessUpdater, 30
  - ilearnrw::datalogger::ProfileAccessUpdater, 43
- setUsername
  - ilearnrw::user::UserDetails, 57
- showMenu
  - ilearnrw::prototype::application::ConsoleMenu, 14
- showThisMenuAgain
  - ilearnrw::prototype::application::ConsoleMenu::E-ConsoleMenuActionResult, 21
- start
  - ilearnrw::user::UserSession, 72
- startSession
  - ilearnrw::user::UserSession, 71
- stopSession
  - ilearnrw::user::UserSession, 71
- subMenu
  - ilearnrw::prototype::application::ConsoleMenu, 14
- Subcategory
  - ilearnrw::user::Subcategory, 49
- subcategoryTitle
  - ilearnrw::user::Subcategory, 49
- title
  - ilearnrw::user::Category, 13
- toString
  - ilearnrw::user::Category, 13
  - ilearnrw::user::ProblemDefinition, 38
  - ilearnrw::user::Subcategory, 49
- type
  - ilearnrw::user::ProblemDefinition, 38
- URI
  - ilearnrw::user::ProblemDefinition, 38
- update
  - ilearnrw::datalogger::UserStore, 75
- User
  - ilearnrw::user::User, 52
- UserDetails
  - ilearnrw::user::UserDetails, 56
- UserId
  - ilearnrw::user::UserAction, 53
- userId
  - ilearnrw::user::User, 52
- UserLogin
  - ilearnrw::prototype::application::UserLogin, 59
- UserPreferences
  - ilearnrw::user::UserPreferences, 61
- UserProblem
  - ilearnrw::user::UserProblem, 63
- UserProblems
  - ilearnrw::prototype::application::UserProblems, 65
- UserProblemsList

- [ilearnrw::user::UserProblemsList](#), [67](#)
- UserProfile
  - [ilearnrw::user::UserProfile](#), [70](#)
- UserStore
  - [ilearnrw::datalogger::UserStore](#), [74](#)
- Username
  - [ilearnrw::datalogger::LoginDetails](#), [32](#)
- username
  - [ilearnrw::user::UserDetails](#), [57](#)
- writePendingChanges
  - [ilearnrw::datalogger::IProfileAccessUpdater](#), [30](#)
  - [ilearnrw::datalogger::ProfileAccessUpdater](#), [43](#)