



CLIPS

Definizione di Prodotto v2.0.0

Sommario

Descrizione approfondita dell'architettura del *progetto*_g CLIPS.

Nome del documento	Definizione di Prodotto
Versione	2.0.0
Data di redazione	2016/09/12
Redazione	Viviana Alessio Matteo Franco Andrea Grandene Luca Soldera Tommaso Panozzo
Verifica	Enrico Bellio Matteo Franco Luca Soldera
Approvazione	Tommaso Panozzo
Uso	Esterno
Lista di distribuzione	prof. Tullio Vardanega prof. Riccardo Cardin Miriade SpA

Diario delle modifiche

Versione	Riepilogo	Autore	Ruolo	Data
2.0.0	Approvazione documento	Andrea Grendene	Responsabile	2016-09-08
1.5.0	Verifica documento	Viviana Alessio	Progettista	2016-09-07
1.4.0	Aggiunta tabella tracciamento classi requisiti	Matteo Franco	Progettista	2016-09-07
1.3.0	Descrizione server - DB	Tommaso Panozzo	Progettista	2016-09-04
1.2.0	Verifica documento	Luca Soldera	Progettista	2016-09-03
1.1.2	Spiegazione scelta singleton	Andrea Grendene	Progettista	2016-09-01
1.1.1	Aggiunta diagrammi di sequenza registrazione e login	Viviana Alessio	Progettista	2016-09-01
1.1.0	Riformattazione descrizione componenti	Viviana Alessio	Progettista	2016-09-01
1.0.0	Approvazione documento	Tommaso Panozzo	Responsabile	2016-08-17
0.7.0	Verifica documento	Matteo Franco	Progettista	2016-08-12
0.6.1	Inserimento immagini per le varie componenti	Enrico Bellio	Progettista	2016-08-10
0.6.0	Verifica documento	Luca Soldera	Progettista	2016-08-03
0.5.4	Correzioni sezione "Componente CLIPS::server::dataserver"	Matteo Franco	Progettista	2016-08-02
0.5.3	Correzioni sezione "Componente CLIPS::client::datamanager"	Enrico Bellio	Progettista	2016-08-01
0.5.2	Correzioni sezione "Componente CLIPS::client::data"	Andrea Grandene	Progettista	2016-08-01
0.5.1	Correzioni sezione "Componente CLIPS::client::viewController"	Matteo Franco	Progettista	2016-08-01
0.5.0	Verifica documento	Luca Soldera	Progettista	2016-07-30
0.4.4	Stesura sezione "Componente CLIPS::server::urlrequesthandler"	Tommaso Panozzo	Progettista	2016-07-29
0.4.3	Stesura sezione "Componente CLIPS::server::dataserver"	Luca Soldera	Progettista	2016-07-29

Versione	Riepilogo	Autore	Ruolo	Data
0.4.2	Stesura sezione “Componente CLIPS::server”	Luca Soldera	Progettista	2016-07-28
0.4.1	Stesura sezione “Componente CLIPS::client::data:: urlrequest”	Tommaso Panozzo	Progettista	2016-07-27
0.4.0	Verifica documento	Luca Soldera	Progettista	2016-07-26
0.3.2	Stesura sezione “Componente CLIPS::client ::pathprogress”	Tommaso Panozzo	Progettista	2016-07-25
0.3.1	Stesura sezione “Componente CLIPS::client::data” e “Componente CLIPS::client:: data::datamanager”	Andrea Grandene	Progettista	2016-07-25
0.3.0	Verifica documento	Matteo Franco	Progettista	2016-07-24
0.2.4	Stesura sezione “Componente CLIPS::client ::authentication”	Luca Soldera	Progettista	2016-07-23
0.2.3	Ampliamento sezione “Componente CLIPS::client ::viewController”	Matteo Franco	Progettista	2016-07-23
0.2.2	Ampliamento sezione “Componente CLIPS::client::viewController”	Viviana Alessio	Progettista	2016-07-22
0.2.1	Stesura sezione “Componente CLIPS::client ::viewController”	Viviana Alessio	Progettista	2016-07-20
0.2.0	Verifica sezione “Standard di progetto”	Matteo Franco	Progettista	2016-07-19
0.1.1	Ampliata sezione “Standard di progetto”	Enrico Bellio	Progettista	2016-07-18
0.1.0	Aggiunta sezione “Standard di progetto”	Luca Soldera	Progettista	2016-07-17
0.0.2	Aggiunta sezione Struttura del database	Luca Soldera	Progettista	2016-07-15
0.0.1	Creazione documento	Tommaso Panozzo	Responsabile	2016-07-10

Indice

1	Introduzione	10
1.1	Scopo del documento	10
1.2	Scopo del <i>prodotto</i> _g	10
1.3	Glossario	10
1.4	Riferimenti	10
1.4.1	Normativi	10
1.4.2	Informativi	10
2	Standard di Progetto	12
2.1	Standard di progettazione architettuale	12
2.2	Standard di documentazione del codice	12
2.3	Standard di denominazione di entità e relazioni	12
2.4	Standard di programmazione	12
2.5	Strumenti di lavoro	12
3	Specifica delle componenti	13
3.1	Componente CLIPS	13
3.1.1	Informazioni sul package	13
3.2	Componente CLIPS::client	13
3.2.1	Informazioni sul package	13
3.3	Componente CLIPS::client::data	14
3.3.1	Informazioni sul package	14
3.3.2	Classi	15
3.3.2.1	CLIPS::client::data::/textitAbstractQuiz	15
3.3.2.2	CLIPS::client::data::/textitTest	15
3.3.2.3	CLIPS::client::data::AppInfo	16
3.3.2.4	CLIPS::client::data::Beacon	17
3.3.2.5	CLIPS::client::data::BiggerShape	17
3.3.2.6	CLIPS::client::data::BiggerShapeTest	17
3.3.2.7	CLIPS::client::data::Building	18
3.3.2.8	CLIPS::client::data::GameCollection	18
3.3.2.9	CLIPS::client::data::LeaderBoard	19
3.3.2.10	CLIPS::client::data::LinearScoringAlgorithm	19
3.3.2.11	CLIPS::client::data::LoggedUser	20
3.3.2.12	CLIPS::client::data::MultipleChoiceImageQuiz	20
3.3.2.13	CLIPS::client::data::MultipleChoiceTest	20
3.3.2.14	CLIPS::client::data::MultipleChoiceTextQuiz	21
3.3.2.15	CLIPS::client::data::Path	21
3.3.2.16	CLIPS::client::data::PathInfo	21
3.3.2.17	CLIPS::client::data::PathProgress	22
3.3.2.18	CLIPS::client::data::PathResult	23
3.3.2.19	CLIPS::client::data::Proof	23
3.3.2.20	CLIPS::client::data::ProofResult	24
3.3.2.21	CLIPS::client::data::Proximity	24
3.3.2.22	CLIPS::client::data::Score	25
3.3.2.23	CLIPS::client::data::ScoringAlgorithmFactory	25
3.3.2.24	CLIPS::client::data::Step	25
3.3.2.25	CLIPS::client::data::TestBuilder	26
3.3.2.26	CLIPS::client::data::TrueFalseImageQuiz	27
3.3.2.27	CLIPS::client::data::TrueFalseTest	27
3.3.2.28	CLIPS::client::data::TrueFalseTextQuiz	28
3.3.2.29	CLIPS::client::data::Utility	28
3.4	Componente CLIPS::client::data::datamanager	29

3.4.1	Informazioni sul package	29
3.4.2	Classi	29
3.4.2.1	CLIPS::client::data::datamanager::/textitSaveDataManager	29
3.4.2.2	CLIPS::client::data::datamanager::AbstractDataManagerListener	31
3.4.2.3	CLIPS::client::data::datamanager::AppInfoDataRequest . .	32
3.4.2.4	CLIPS::client::data::datamanager::BuildingsDataRequest .	33
3.4.2.5	CLIPS::client::data::datamanager::CheckResult	34
3.4.2.6	CLIPS::client::data::datamanager::DataManager	34
3.4.2.7	CLIPS::client::data::datamanager::DataRequestMaker . . .	35
3.4.2.8	CLIPS::client::data::datamanager::DBHandler	37
3.4.2.9	CLIPS::client::data::datamanager::GetRankingDataRequest	45
3.4.2.10	CLIPS::client::data::datamanager::GetResultsDataRequest	46
3.4.2.11	CLIPS::client::data::datamanager::LoginManager	46
3.4.2.12	CLIPS::client::data::datamanager::PathDataRequest	48
3.4.2.13	CLIPS::client::data::datamanager::SaveResultDataRequest	49
3.5	Componente CLIPS::client::data::urlrequest	50
3.5.1	Informazioni sul package	50
3.5.2	Classi	51
3.5.2.1	CLIPS::client::data::urlrequest::AbstractUrlRequestListener	51
3.5.2.2	CLIPS::client::data::urlrequest::AppInfoRequest	52
3.5.2.3	CLIPS::client::data::urlrequest::BuildingsRequest	52
3.5.2.4	CLIPS::client::data::urlrequest::ChangeProfileDataRequest	53
3.5.2.5	CLIPS::client::data::urlrequest::CheckRequest	53
3.5.2.6	CLIPS::client::data::urlrequest::ForgotPasswordRequest .	54
3.5.2.7	CLIPS::client::data::urlrequest::GetProfileDataRequest .	54
3.5.2.8	CLIPS::client::data::urlrequest::GetRankingRequest . . .	54
3.5.2.9	CLIPS::client::data::urlrequest::GetResultsRequest . . .	55
3.5.2.10	CLIPS::client::data::urlrequest::LoginRequest	55
3.5.2.11	CLIPS::client::data::urlrequest::LogoutRequest	56
3.5.2.12	CLIPS::client::data::urlrequest::PathRequest	56
3.5.2.13	CLIPS::client::data::urlrequest::RegistrationRequest . .	56
3.5.2.14	CLIPS::client::data::urlrequest::RequestMaker	57
3.5.2.15	CLIPS::client::data::urlrequest::SaveResultRequest . . .	60
3.5.2.16	CLIPS::client::data::urlrequest::ServerError	61
3.5.2.17	CLIPS::client::data::urlrequest::URLDataConstants	61
3.5.2.18	CLIPS::client::data::urlrequest::URLRequest	61
3.6	Componente CLIPS::client::pathprogress	62
3.6.1	Informazioni sul package	62
3.6.2	Classi	63
3.6.2.1	CLIPS::client::pathprogress::BeaconDiscover	63
3.6.2.2	CLIPS::client::pathprogress::BeaconDiscoverDelegate . . .	63
3.6.2.3	CLIPS::client::pathprogress::GPSListener	64
3.6.2.4	CLIPS::client::pathprogress::GPSManager	64
3.6.2.5	CLIPS::client::pathprogress::PathProgressController . . .	65
3.6.2.6	CLIPS::client::pathprogress::PathProgressControllerDelegate	65
3.6.2.7	CLIPS::client::pathprogress::PathProgressMaker	66
3.6.2.8	CLIPS::client::pathprogress::RawBeacon	66
3.7	Componente CLIPS::client::viewController	67
3.7.1	Informazioni sul package	67
3.7.2	Classi	67
3.7.2.1	CLIPS::client::viewController::ResultAdapter	67
3.8	Componente CLIPS::client::viewController::authentication	67
3.8.1	Informazioni sul package	67
3.8.2	Classi	68
3.8.2.1	CLIPS::client::viewController::authentication::AccountActivity	68
3.8.2.2	CLIPS::client::viewController::authentication::AccountView	69

3.8.2.3	CLIPS::client::viewController::authentication::ChangeProfileActivity	69
3.8.2.4	CLIPS::client::viewController::authentication::ConfirmRegistration	69
3.8.2.5	CLIPS::client::viewController::authentication::ForgotPasswordActivity	69
3.8.2.6	CLIPS::client::viewController::authentication::ForgotPasswordView	70
3.8.2.7	CLIPS::client::viewController::authentication::LoginActivity	70
3.8.2.8	CLIPS::client::viewController::authentication::LoginView	70
3.8.2.9	CLIPS::client::viewController::authentication::RecoverPassword	71
3.8.2.10	CLIPS::client::viewController::authentication::RegistrationActivity	71
3.8.2.11	CLIPS::client::viewController::authentication::RegistrationView	71
3.8.2.12	CLIPS::client::viewController::authentication::UpdateUserInfoActivity	72
3.8.2.13	CLIPS::client::viewController::authentication::UpdateUserInfoView	72
3.9	Componente CLIPS::client::viewController::building	73
3.9.1	Informazioni sul package	73
3.9.2	Classi	74
3.9.2.1	CLIPS::client::viewController::building::BuildingActivity	74
3.9.2.2	CLIPS::client::viewController::building::BuildingRow	74
3.9.2.3	CLIPS::client::viewController::building::BuildingSearchActivity	74
3.9.2.4	CLIPS::client::viewController::building::BuildingSearchView	75
3.9.2.5	CLIPS::client::viewController::building::BuildingView	75
3.9.2.6	CLIPS::client::viewController::building::PathActivity	76
3.9.2.7	CLIPS::client::viewController::building::PathRow	77
3.9.2.8	CLIPS::client::viewController::building::PathView	77
3.9.2.9	CLIPS::client::viewController::building::SearchCriteriaRow	77
3.10	Componente CLIPS::client::viewController::games	77
3.10.1	Informazioni sul package	77
3.10.2	Classi	79
3.10.2.1	CLIPS::client::viewController::games::MultipleChoiceQuizActivity	79
3.10.2.2	CLIPS::client::viewController::games::MultipleChoiceQuizView	79
3.10.2.3	CLIPS::client::viewController::games::ProofActivity	80
3.10.2.4	CLIPS::client::viewController::games::ProofResultActivity	80
3.10.2.5	CLIPS::client::viewController::games::ProofResultRow	80
3.10.2.6	CLIPS::client::viewController::games::ProofResultView	80
3.10.2.7	CLIPS::client::viewController::games::ProofView	81
3.10.2.8	CLIPS::client::viewController::games::QuizActivity	81
3.10.2.9	CLIPS::client::viewController::games::QuizResultView	81
3.10.2.10	CLIPS::client::viewController::games::QuizView	81
3.10.2.11	CLIPS::client::viewController::games::SearchNewStepActivity	82
3.10.2.12	CLIPS::client::viewController::games::SearchNewStepView	82
3.10.2.13	CLIPS::client::viewController::games::TestResultView	82
3.10.2.14	CLIPS::client::viewController::games::TestView	83
3.10.2.15	CLIPS::client::viewController::games::TrueFalseQuizActivity	83
3.10.2.16	CLIPS::client::viewController::games::TrueFalseQuizView	83
3.11	Componente CLIPS::client::viewController::savedresults	84
3.11.1	Informazioni sul package	84
3.11.2	Classi	85
3.11.2.1	CLIPS::client::viewController::savedresults::PlayerRow	85
3.11.2.2	CLIPS::client::viewController::savedresults::RankingActivity	85
3.11.2.3	CLIPS::client::viewController::savedresults::RankingView	85
3.11.2.4	CLIPS::client::viewController::savedresults::ResultActivity	85
3.11.2.5	CLIPS::client::viewController::savedresults::ResultRow	86
3.11.2.6	CLIPS::client::viewController::savedresults::ResultView	86
3.11.2.7	CLIPS::client::viewController::savedresults::SavedResultsActivity	87
3.11.2.8	CLIPS::client::viewController::savedresults::SavedResultView	87
3.12	Componente CLIPS::client::viewController::utility	87
3.12.1	Informazioni sul package	87
3.12.2	Classi	88

3.12.2.1	CLIPS::client::viewController::utility::AppInfoActivity	88
3.12.2.2	CLIPS::client::viewController::utility::AppInfoView . . .	88
3.12.2.3	CLIPS::client::viewController::utility::MenuActivity . .	89
3.12.2.4	CLIPS::client::viewController::utility::MenuView	89
3.13	Componente CLIPS::server	89
3.13.1	Informazioni sul package	89
3.14	Componente CLIPS::server::dataserver	90
3.14.1	Informazioni sul package	90
3.14.2	Classi	90
3.14.2.1	CLIPS::server::dataserver::AppInfoServer	90
3.14.2.2	CLIPS::server::dataserver::AuthTokenServer	90
3.14.2.3	CLIPS::server::dataserver::BeaconServer	91
3.14.2.4	CLIPS::server::dataserver::BuildingServer	91
3.14.2.5	CLIPS::server::dataserver::LeaderBoardServer	91
3.14.2.6	CLIPS::server::dataserver::PathResultServer	91
3.14.2.7	CLIPS::server::dataserver::PathServer	91
3.14.2.8	CLIPS::server::dataserver::ProofResultServer	91
3.14.2.9	CLIPS::server::dataserver::ProofServer	91
3.14.2.10	CLIPS::server::dataserver::ProximityServer	92
3.14.2.11	CLIPS::server::dataserver::ScoreManager	92
3.14.2.12	CLIPS::server::dataserver::StepServer	92
3.14.2.13	CLIPS::server::dataserver::UserServer	92
3.15	Componente CLIPS::server::urlrequesthandler	92
3.15.1	Informazioni sul package	92
3.15.2	Classi	92
3.15.2.1	CLIPS::server::urlrequesthandler::AppInfoProvider	92
3.15.2.2	CLIPS::server::urlrequesthandler::BuildingsProvider . . .	93
3.15.2.3	CLIPS::server::urlrequesthandler::DBHandler	94
3.15.2.4	CLIPS::server::urlrequesthandler::EmailChecker	94
3.15.2.5	CLIPS::server::urlrequesthandler::EmailSender	94
3.15.2.6	CLIPS::server::urlrequesthandler::GetUserData	94
3.15.2.7	CLIPS::server::urlrequesthandler::LoginHandler	95
3.15.2.8	CLIPS::server::urlrequesthandler::LogoutHandler	95
3.15.2.9	CLIPS::server::urlrequesthandler::PasswordChecker	95
3.15.2.10	CLIPS::server::urlrequesthandler::PasswordResetEmailSender	95
3.15.2.11	CLIPS::server::urlrequesthandler::PasswordResetHandler .	96
3.15.2.12	CLIPS::server::urlrequesthandler::PathProvider	96
3.15.2.13	CLIPS::server::urlrequesthandler::PathResultsProvider .	98
3.15.2.14	CLIPS::server::urlrequesthandler::PostUserData	98
3.15.2.15	CLIPS::server::urlrequesthandler::RankingProvider	98
3.15.2.16	CLIPS::server::urlrequesthandler::RegistrationFieldsValidator	99
3.15.2.17	CLIPS::server::urlrequesthandler::RegistrationHandler .	99
3.15.2.18	CLIPS::server::urlrequesthandler::TokenGenerator	99
3.15.2.19	CLIPS::server::urlrequesthandler::URLRequestHandler . . .	99
3.15.2.20	CLIPS::server::urlrequesthandler::URLRequestRouter . . .	100
3.15.2.21	CLIPS::server::urlrequesthandler::UserDataRequest	100
3.15.2.22	CLIPS::server::urlrequesthandler::UsernameChecker	100
3.15.2.23	CLIPS::server::urlrequesthandler::UserResultProvider . .	101
4	Struttura del database	101
5	Diagrammi di Sequenza	102
5.1	Registrazione	102
5.2	Login	102
5.3	Richiesta lista edifici abilitati	103
5.4	Scoperta nuovi <i>Beacon_g</i>	103

6	Tabella tracciamento Requisiti-Componenti	105
7	Tabella tracciamento Componenti-Requisiti	124
8	Tabella tracciamento Classi-Requisiti	142
9	Tabella tracciamento Requisiti-Classi	167
10	Algoritmo di assegnazione del punteggio	192
11	Specifica JSON client-server	193
11.1	AppInfo	193
11.2	Path	193
11.3	Algorithm	194
11.4	Proof	194
11.4.1	Scelta Multipla	195
11.4.2	Vero o Falso	195

Elenco delle tabelle

1	Tracciamento requisiti-componenti	123
2	Tracciamento componenti-requisiti	141
3	Tracciamento classi-requisiti	166
4	Tracciamento requisiti-classi	191

Elenco delle figure

1	Schema package client	13
2	Schema sintetico package client::data	14
3	Prima parte schema package client::data	14
4	Seconda parte schema package client::data	15
5	Terza parte schema package client::data	16
6	Schema sintetico package client::data::datamanager	30
7	Prima parte schema package client::data::datamanager	30
8	Seconda parte schema package client::data::datamanager	31
9	Terza parte schema package client::data::datamanager	32
10	Schema sintetico package client::data::urlrequest	50
11	Prima parte schema package client::data::urlrequest	50
12	Seconda parte schema package client::data::urlrequest	51
13	Terza parte schema package client::data::urlrequest	52
14	Schema package client::pathprogress	63
15	Schema package client::viewController	67
16	Prima parte schema package client::viewController::authentication	68
17	Seconda parte schema package client::viewController::authentication	68
18	Schema package client::viewController::building	73
19	Prima parte schema package client::viewController::games	78
20	Seconda parte schema package client::viewController::games	78
21	Terza parte schema package client::viewController::games	79
22	Schema package client::viewController::savedresults	84
23	Schema package client::viewController::utility	88
24	Schema package server	89
25	Schema package server::dataserver	90
26	Schema package server::urlrequesthandler	93
27	Struttura del database dell'applicazione	101
28	Diagramma di Sequenza della registrazione	102
29	Diagramma di Sequenza della login	102
30	Diagramma di Sequenza della richiesta di edifici abilitati ai percorsi	103
31	Diagramma di Sequenza della scoperta di nuovi beacon	104

1 Introduzione

1.1 Scopo del documento

Questo documento ha lo scopo di spiegare dettagliatamente l'implementazione delle classi del *prodotto_g* didattico.

1.2 Scopo del *prodotto_g*

Il prodotto finale consisterà di un'applicazione mobile che, interagendo con dei beacons sparsi nell'area designata, guiderà l'utente attraverso un percorso. L'utente potrà completare il percorso superando tutte le prove che gli si presenteranno nelle diverse tappe. Le prove potranno essere degli indovinelli o dei semplici giochi inerenti all'area in cui si svolge il percorso.

1.3 Glossario

Al fine di evitare ogni ambiguità nel linguaggio e massimizzare la comprensione dei documenti, i termini tecnici, gli acronimi e le abbreviazioni che necessitano di definizione sono riportati nel documento "*Glossario v3.0.0*".

Inoltre ogni occorrenza di un vocabolo presente nel Glossario sarà posta in corsivo e seguita da una 'g' minuscola a pedice (p.es. *Glossario_g*).

1.4 Riferimenti

1.4.1 Normativi

- **Capitolato d'appalto C2 - CLIPS:** Communication & Localisation with Indoor Positioning Systems.
<http://www.math.unipd.it/~tullio/IS-1/2015/Progetto/C2.pdf>
- **Norme di Progetto**
"*Norme di Progetto v4.0.0*"
- **Analisi dei Requisiti**
"*Analisi dei Requisiti v4.0.0*"

1.4.2 Informativi

- **Software Engineering (10th edition)**
Ian Sommerville
Pearson Education — Addison-Wesley
- **Guide to the Software Engineering Body of Knowledge** IEEE Computer Society.
Software Engineering Coordinating Committee
- **Slides del prof. Tullio Vardanega**
riguardo la [progettazione](#)
- **Slides del prof. Riccardo Cardin**
riguardo UML
[diagrammi delle classi](#)
[diagrammi dei package](#)
[diagrammi di attività](#)
[diagrammi di sequenza](#)

riguardo i design pattern

[design pattern strutturali](#)

[design pattern creazionali](#)

[design pattern comportamentali](#)

- [Documentazione per lo sviluppo di applicazioni per Android](#)
- [Documentazione java](#)
- [Documentazione JSON](#)
- [Documentazione PHP](#)
- [Documentazione MongoDB](#)
- [Docuentazione Android Beacon Library](#)
- [Documentazione AltBeacon](#)

2 Standard di Progetto

2.1 Standard di progettazione architettuale

Gli standard di progettazione architettuale sono specificati nel documento “*Specifica Tecnica v3.0.0*”. Per approfondimenti si faccia riferimento a tale documento.

2.2 Standard di documentazione del codice

Le metriche di documentazione del codice sono descritte dettagliatamente nel documento “*Piano di Qualifica v4.0.0*” nella sezione. Per approfondimenti si faccia riferimento a tale documento.

2.3 Standard di denominazione di entità e relazioni

Le norme tipografiche utilizzate sono definite nel documento “*Norme di Progetto v4.0.0*” nella sezione 2.1.4. Per approfondimenti si faccia riferimento a tale sezione.

2.4 Standard di programmazione

Le regole per lo standard di programmazione sono definite nel documento “*Norme di Progetto v4.0.0*” nella sezione 2.1.4. Per approfondimenti si faccia riferimento a tale documento.

2.5 Strumenti di lavoro

Gli strumenti utilizzati per lo sviluppo del progetto sono specificati nei documenti “*Norme di Progetto v4.0.0*” nella sezione 2.1.3. Per approfondimenti si faccia riferimento a tale documento.

3 Specifica delle componenti

3.1 Componente CLIPS

3.1.1 Informazioni sul package

- **Descrizione:** package generale contenente il prodotto del progetto
- **Package contenuti:**
 - CLIPS::client: componente globale per il front end del prodotto che utilizza il design pattern MVC_g . Si occupa di fornire un'interfaccia grafica dell'applicazione e di interagire con il lato server.
 - CLIPS::server: componente globale per il back end del prodotto

3.2 Componente CLIPS::client

3.2.1 Informazioni sul package

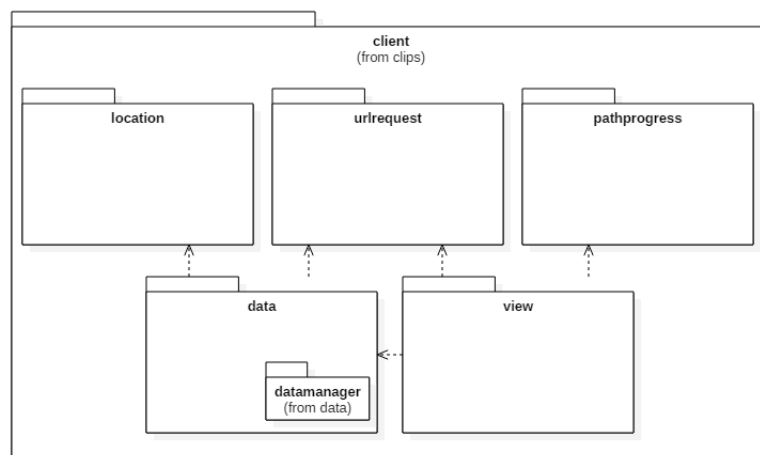


Figura 1: Schema package client

- **Descrizione:** componente globale per il front end del prodotto che utilizza il design pattern MVC_g . Si occupa di fornire un'interfaccia grafica dell'applicazione e di interagire con il lato server.
- **Padre:** CLIPS
- **Package contenuti:**
 - CLIPS::client::data: package per la gestione in locale dei dati
 - CLIPS::client::pathprogress: componente che gestisce i dati del percorso e salva i risultati ottenuti nelle prove mentre si sta giocando
 - CLIPS::client::viewController: componente che raggruppa tutte le view ed i controller relativi alle view

3.3 Componente CLIPS::client::data

3.3.1 Informazioni sul package

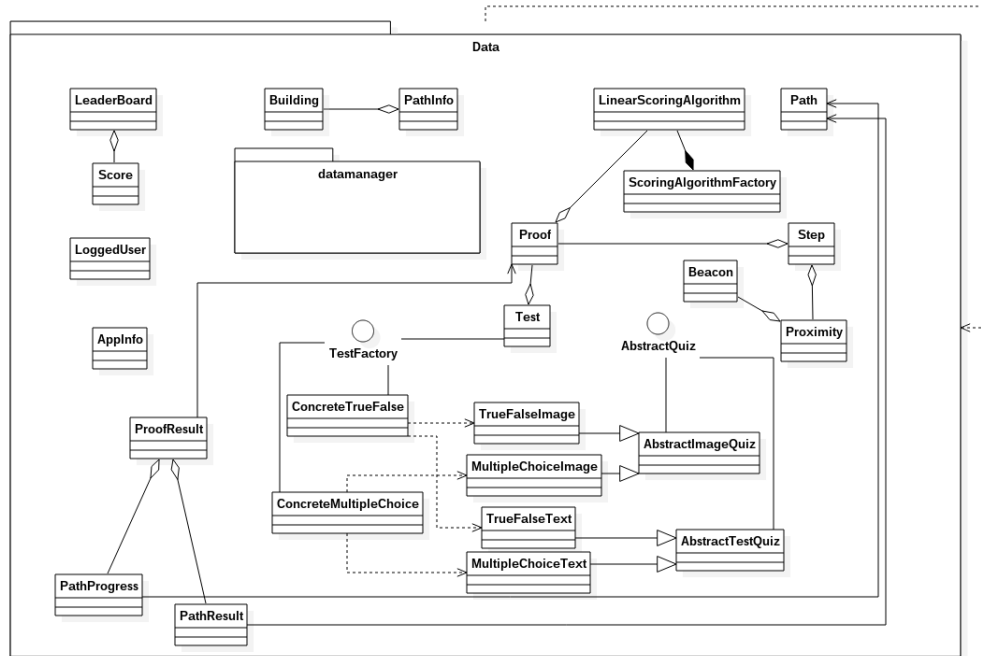


Figura 2: Schema sintetico package client::data

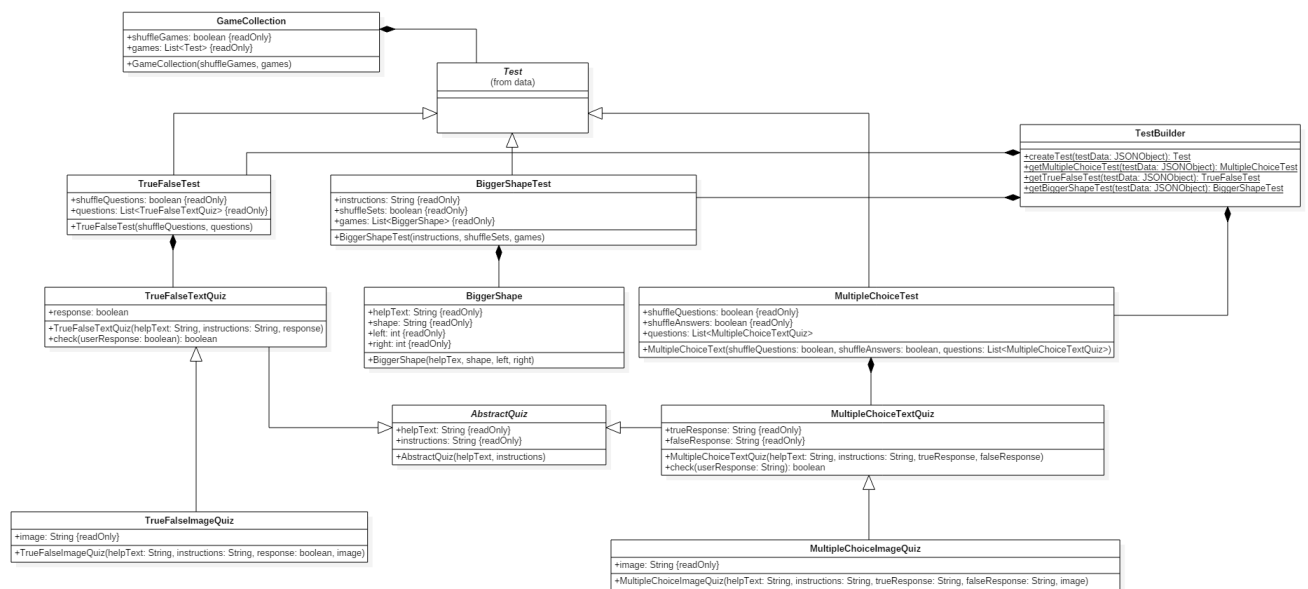


Figura 3: Prima parte schema package client::data

- **Descrizione:** package per la gestione in locale dei dati
- **Padre:** client
- **Package contenuti:**

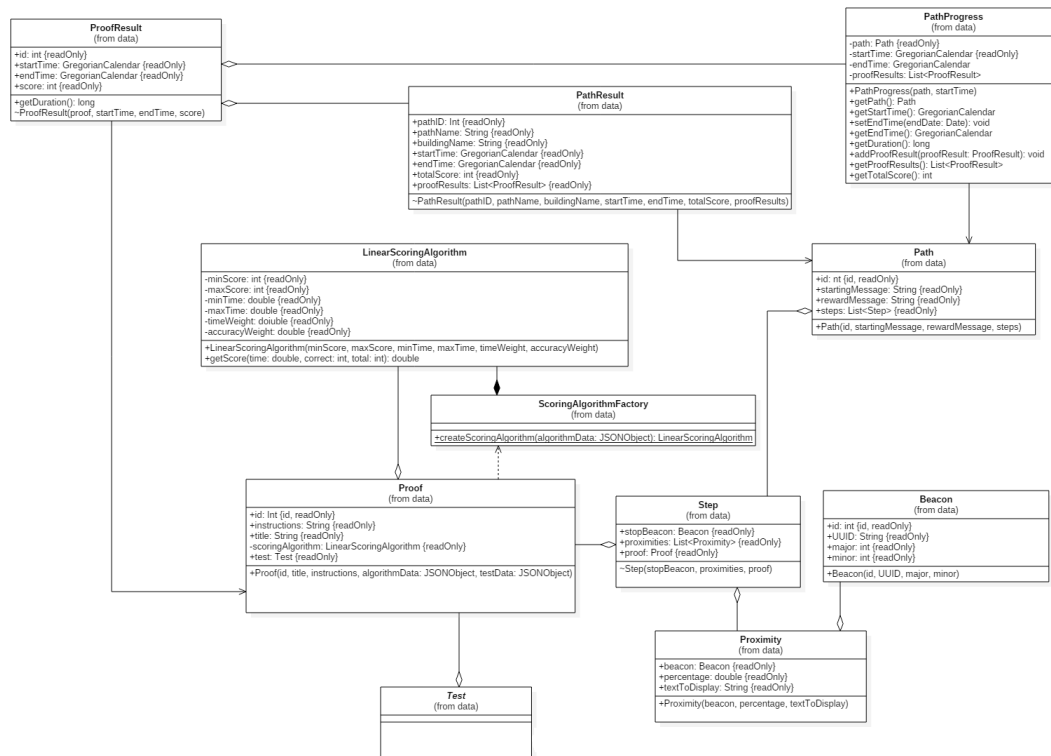


Figura 4: Seconda parte schema package client::data

- CLIPS::client::data::datamanager: componente che gestisce i dati in locale
- CLIPS::client::data::urlrequest: componente che si occupa di effettuare le richieste al server

3.3.2 Classi

3.3.2.1 CLIPS::client::data::textitAbstractQuiz

- **Descrizione:** classe astratta per la definizione di un quiz da proporre all'utente
- **Utilizzo:** classe astratta da cui far derivare tutte le classi che rappresentano un quiz
- **Attributi:**

helpText: string stringa contenente il messaggio d'aiuto per la prova

instructions: string stringa contenente le istruzioni della prova

3.3.2.2 CLIPS::client::data::textitTest

- **Descrizione:** classe astratta che rappresenta la definizione di un test da proporre all'utente
- **Utilizzo:** classe astratta per rappresentare un test, ovvero un gruppo di giochi generalmente dello stesso tipo

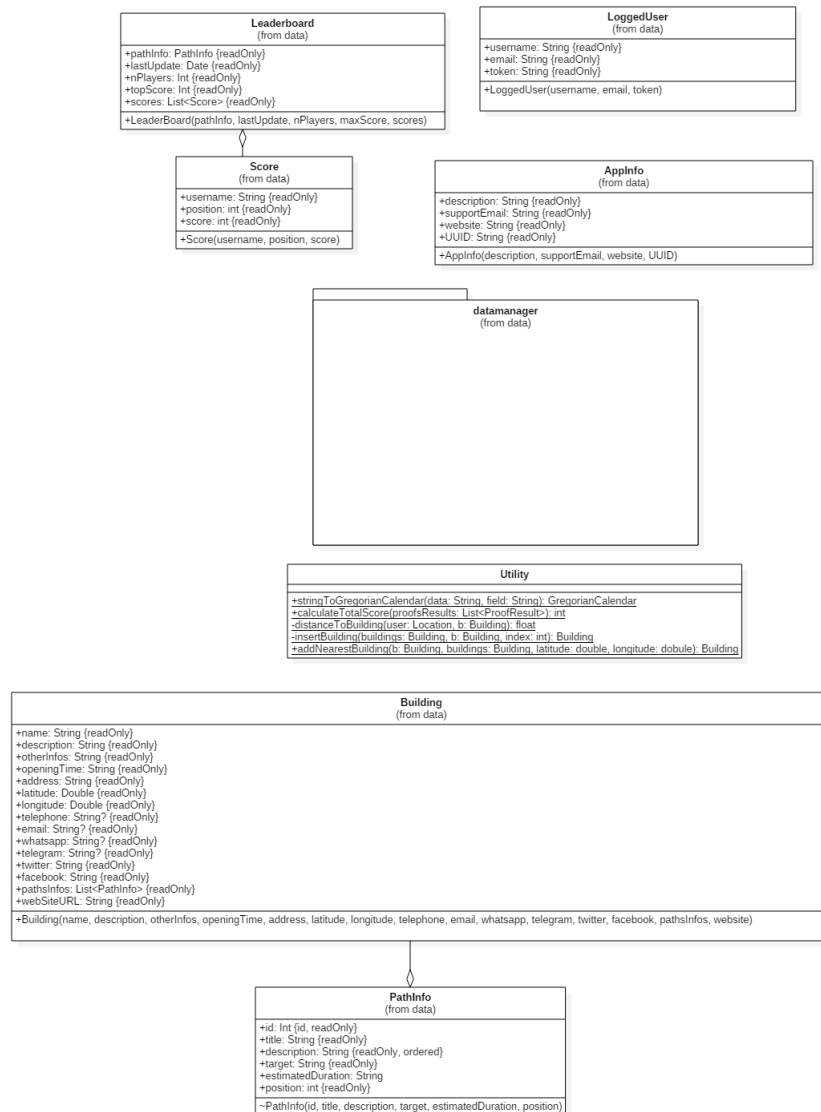


Figura 5: Terza parte schema package client::data

3.3.2.3 CLIPS::client::data::AppInfo

- **Descrizione:** classe per la rappresentazione delle informazioni generali dell'applicazione
- **Utilizzo:** permette di salvare le informazioni generali relative all'applicazione
- **Attributi:**
 - description:** string stringa contenente la descrizione dell'applicazione
 - supportEmail:** string stringa contenente l'email per l'assistenza
 - UUID:** string stringa contenente l'UUID utilizzato dai beacon dell'applicazione
 - website:** string stringa contenente il sito dell'applicazione

3.3.2.4 CLIPS::client::data::Beacon

- **Descrizione:** classe che rappresenta i dati di un beacon in locale
- **Utilizzo:** permette di salvare in locale le informazioni di un beacon

- **Attributi:**

`id: int` rappresenta il codice identificativo
`major: int` rappresenta il parametro major di un beacon
`minor: int` rappresenta il parametro minor di un beacon
`UUID: string` rappresenta il codice UUID di un beacon

- **Metodi:**

`Beacon(id:int,UUID:string,major:int,minor:int)(id, major, minor, UUID)`
`: void`
costruttore della classe beacon

Argomenti:

* `id : int` codice identificativo
* `major : int` parametro major del beacon
* `minor : int` parametro minor del beacon
* `UUID : string` codice UUID

3.3.2.5 CLIPS::client::data::BiggerShape

- **Descrizione:** classe che rappresenta un gioco in cui bisogna selezionare la figura più grande tra le due che appaiono
- **Utilizzo:** classe per rappresentare un gioco in cui l'utente deve selezionare la figura più grande tra le due che compaiono

- **Attributi:**

`helpText: string` stringa contenente il messaggio d'aiuto per la prova
`left: int` intero contenente l'indice della figura rappresentata a sinistra
`right: int` intero contenente l'indice della figura rappresentata a destra
`shape: string` stringa contenente il tipo di figura rappresentata

3.3.2.6 CLIPS::client::data::BiggerShapeTest

- **Descrizione:** classe che rappresenta un test di BiggerShape
- **Utilizzo:** classe per rappresentare un test di BiggerShape

- **Attributi:**

`games: List<BiggerShape>` lista contenente i quesiti proposti

`instructions:` `string` stringa contenente le istruzioni della prova
`shuffleSets:` `boolean` booleano per indicare se la lista dei quesiti va mescolata o meno

- **Relazioni con altre classi:**

`CLIPS::client::data::BiggerShape:` classe che rappresenta un gioco in cui bisogna selezionare la figura più grande tra le due che appaiono

3.3.2.7 `CLIPS::client::data::Building`

- **Descrizione:** classe che rappresenta i dati di un edificio in cui è possibile utilizzare l'applicazione

- **Utilizzo:** permette di memorizzare i dati di un edificio in locale

- **Attributi:**

`address:` `string` indica l'indirizzo dell'edificio
`description:` `string` rappresenta una breve descrizione dell'edificio
`email:` `string` rappresenta l'indirizzo email dell'edificio
`facebook:` `string` indica l'indirizzo facebook dell'edificio
`id:` `int` identifica l'edificio univocamente
`latitude:` `double` rappresenta la latitudine dell'edificio
`longitude:` `double` rappresenta la longitudine dell'edificio
`name:` `string` indica il nome dell'edificio
`openingTime:` `string` rappresenta gli orari di apertura dell'edificio
`otherInfos:` `string` contiene informazioni aggiuntive sull'edificio
`pathsInfos:` `string` fornisce informazioni sui percorsi dell'edificio
`telegram:` `string` rappresenta il contatto telegram dell'edificio
`telephone:` `string` indica il numero di telefono dell'edificio
`twitter:` `string` rappresenta l'indirizzo twitter dell'edificio
`webSiteURL:` `string` indica l'indirizzo web dell'edificio
`whatsapp:` `string` indica il contatto whatsapp dell'edificio

- **Relazioni con altre classi:**

`CLIPS::client::data::PathInfo:` classe che si occupa di salvare in locale le informazioni generali di un percorso

3.3.2.8 `CLIPS::client::data::GameCollection`

- **Descrizione:** classe che rappresenta un insieme di test

- **Utilizzo:** classe per rappresentare un test in cui sono contenuti più tipi di giochi, di fatto è un test che contiene altri test

- **Attributi:**

games: List<Test> lista contenente i test da proporre

shuffleGames: boolean booleano per indicare se la lista dei Test va mescolata o meno

- **Relazioni con altre classi:**

CLIPS::client::data::TextitTest: classe astratta che rappresenta un test

3.3.2.9 CLIPS::client::data::LeaderBoard

- **Descrizione:** classe che rappresenta la classifica di un percorso di un determinato edificio

- **Utilizzo:** consente di salvare i dati della classifica in locale

- **Attributi:**

lastUpdate: DataManager data contenente la data dell'ultimo aggiornamento

maxScore: int intero contenente lo Score massimo raggiunto nella Leaderboard

nPlayers: int intero contenente il numero di giocatori nella Leaderboard

pathInfo: PathInfo oggetto PathInfo di cui registrare i record

scores: List<Score> lista contenente gli Score da mostrare nella Leaderboard

- **Relazioni con altre classi:**

CLIPS::client::data::PathInfo: classe che si occupa di salvare in locale le informazioni generali di un percorso

CLIPS::client::data::Score: classe che rappresenta il risultato nella classifica di uno specifico percorso svolto dall'utente

3.3.2.10 CLIPS::client::data::LinearScoringAlgorithm

- **Descrizione:** classe per calcolare il punteggio di una prova

- **Utilizzo:** si occupa di calcolare il punteggio utilizzando un algoritmo lineare

- **Attributi:**

accuracyWeight: double indica l'incidenza dell'accuratezza rispetto alla risposta

maxScore: int indica il punteggio massimo

maxTime: double indica il tempo massimo

minScore: int indica il punteggio minimo

minTime: double indica il tempo minimo

timeWeight: double indica l'incidenza del tempo nel calcolo del punteggio

- **Metodi:**

getScore(correct, total) : int
fornisce il punteggio ottenuto nella prova

Argomenti:

- * **correct** : **int** indica il numero di risposte corrette
- * **total** : **int** *da aggiungere*

3.3.2.11 CLIPS::client::data::LoggedInUser

- **Descrizione:** classe che si occupa di memorizzare in locale i dati dell'utente che ha eseguito l'accesso
- **Utilizzo:** permette il salvataggio in locale dei dati di un utente loggato
- **Attributi:**
 - email:** **string** stringa contenente l'email dell'utente che ha fatto il login
 - token:** **string** stringa contenente il token di autenticazione dell'utente che ha fatto il login
 - username:** **string** stringa contenente lo username dell'utente che ha fatto il login

3.3.2.12 CLIPS::client::data::MultipleChoiceImageQuiz

- **Descrizione:** classe che rappresenta una domanda a risposta multipla con immagine
- **Utilizzo:** classe per rappresentare una domanda a risposta multipla in cui viene usata un'immagine per la formulazione del quesito
- **Attributi:**
 - image:** **string** stringa contenente il nome dell'immagine

3.3.2.13 CLIPS::client::data::MultipleChoiceTest

- **Descrizione:** classe che rappresenta un test di domande a risposta multiple
- **Utilizzo:** classe per rappresentare un test di domande a risposta multipla, con immagine o senza
- **Attributi:**
 - questions:** **List<MultipleChoiceTextQuiz>** lista contenente i quesiti proposti
 - shuffleAnswers:** **boolean** booleano per indicare se le risposte vanno mescolate o meno
 - shuffleQuestions:** **boolean** booleano per indicare se le domande vanno mescolate o meno
- **Relazioni con altre classi:**
 - CLIPS::client::data::MultipleChoiceTextQuiz:** classe che rappresenta una domanda a risposta multipla

3.3.2.14 CLIPS::client::data::MultipleChoiceTextQuiz

- **Descrizione:** classe che rappresenta una domanda a risposta multipla
- **Utilizzo:** classe per rappresentare una domanda a risposta multipla il cui contenuto è solo testo
- **Attributi:**
 - `falseResponse:` `string` array contenente le risposte errate
 - `trueResponse:` `string` stringa contenente la risposta corretta
- **Metodi:**
 - `check(userResponse) :` `boolean`
metodo per controllare se la risposta data dall'utente è quella corretta o meno
 - Argomenti:*
 - * `userResponse :` `string` stringa contenente la risposta data dall'utente

3.3.2.15 CLIPS::client::data::Path

- **Descrizione:** classe che si occupa di salvare in locale i dati riguardanti un percorso di un determinato edificio
- **Utilizzo:** permette di salvare i dati di un percorso in locale
- **Attributi:**
 - `id:` `int` intero contenente l'id dell'oggetto
 - `rewardMessage:` `string` stringa contenente il messaggio che appare quando si completa il percorso
 - `startingMessage:` `string` stringa contenente il messaggio che appare quando si comincia il percorso
 - `steps:` `List<Step>` lista contenente gli step del percorso
- **Relazioni con altre classi:**
 - `CLIPS::client::data::Step:` classe che rappresenta in locale lo spostamento da una prova a quella successiva

3.3.2.16 CLIPS::client::data::PathInfo

- **Descrizione:** classe che si occupa di salvare in locale le informazioni generali di un percorso
- **Utilizzo:** consente di salvare in locale le informazioni generali di un percorso
- **Attributi:**
 - `description:` `string` stringa contenente la descrizione del percorso
 - `estimatedDuration:` `string` stringa contenente la durata stimata del percorso

id: `int` intero contenente l'id dell'oggetto
target: `string` stringa per indicare a chi è consigliato il percorso
title: `string` stringa contenente il titolo del percorso

3.3.2.17 CLIPS::client::data::PathProgress

- **Descrizione:** classe per il salvataggio in locale del progresso di un percorso svolto dall'utente
- **Utilizzo:** permette di salvare in locale i risultati delle prove durante lo svolgimento del percorso

- **Attributi:**

endTime: `GregorianCalendar` oggetto contenente la data e l'orario in cui l'utente ha completato il percorso
path: `Path` oggetto `Path` di cui tenere traccia del progresso dell'utente
proofResults: `List<Building>` lista contenente i risultati delle prove completate dall'utente
startTime: `GregorianCalendar` oggetto contenente la data e l'orario in cui l'utente ha cominciato il percorso

- **Metodi:**

addProofResult(proofResult) : `void`
 metodo per aggiungere il risultato di una Proof completata dall'utente alla lista `proofResults`

Argomenti:

* **proofResult :** `ProofResult` oggetto contenente i risultati della prova completata dall'utente

getDuration() : `long`
 metodo per ottenere la durata del percorso svolto dall'utente

getEndTime() : `GregorianCalendar`
 metodo per ottenere l'oggetto contenente la data e l'orario in cui si è completato il percorso

getPath() : `Path`
 metodo per ottenere l'oggetto `Path` a cui si riferisce l'oggetto `PathProgress`

getProofResults() : `List<Building>`
 metodo per ottenere la lista dei risultati delle Proof completate dall'utente

getStartTime() : `GregorianCalendar`
 metodo per ottenere l'oggetto contenente la data e l'orario in cui si è cominciato il percorso

getTotalScore() : `int`
 metodo per calcolare lo Score totale ottenuto dall'utente nel percorso svolto

setEndTime(endTime) : `void`
 metodo per impostare data e orario di quando l'utente ha completato il percorso

Argomenti:

* **endTime** : `GregorianCalendar` oggetto contenente la data e l'orario in cui l'utente ha completato il percorso

- **Relazioni con altre classi:**

`CLIPS::client::data::Path`: classe che si occupa di salvare in locale i dati riguardanti un percorso di un determinato edificio

`CLIPS::client::data::ProofResult`: classe che rappresenta il risultato di una prova svolta dall'utente

3.3.2.18 `CLIPS::client::data::PathResult`

- **Descrizione:** classe che rappresenta i risultati di un percorso svolto dall'utente

- **Utilizzo:** permette di salvare i dati di un percorso in locale

- **Attributi:**

buildingName: `string` stringa contenente il nome dell'edificio in cui si trova il percorso

endTime: `GregorianCalendar` oggetto contenente la data e l'orario in cui l'utente ha completato il percorso

pathID: `int` intero contenente l'id del Patha cui si riferisce l'oggetto

pathName: `string` stringa contenente il nome del percorso

proofResults: `List<ProofResult>` lista contenente i risultati delle prove del percorso completate dall'utente

startTime: `GregorianCalendar` oggetto contenente la data e l'orario in cui l'utente ha cominciato il percorso

totalScore: `int` intero contenente lo Score totale del percorso ottenuto dall'utente

- **Relazioni con altre classi:**

`CLIPS::client::data::ProofResult`: classe che rappresenta il risultato di una prova svolta dall'utente

3.3.2.19 `CLIPS::client::data::Proof`

- **Descrizione:** classe che rappresenta una prova da proporre all'utente che sta svolgendo un percorso

- **Utilizzo:** salva in locale i dati della prova da far visualizzare all'utente

- **Attributi:**

id: `int` intero contenente l'id dell'oggetto

instructions: `string` stringa contenente le istruzioni della prova

scoringAlgorithm: `LinearScoringAlgorithm` oggetto contenente l'algoritmo per calcolare il punteggio della prova

test: `Test` oggetto contenente il quesito da proporre all'utente

`title:` `string` stringa contenente il titolo della prova

- **Relazioni con altre classi:**

`CLIPS::client::data::/textitTest:` classe astratta che rappresenta un test

`CLIPS::client::data::LinearScoringAlgorithm:` classe per calcolare il punteggio di una prova

`CLIPS::client::data::ScoringAlgorithmFactory:` classe base per la creazione degli algoritmi per il calcolo del punteggio di una determinata prova

3.3.2.20 `CLIPS::client::data::ProofResult`

- **Descrizione:** classe che rappresenta il risultato di una prova svolta dall'utente

- **Utilizzo:** permette di salvare in locale il risultato di una prova

- **Attributi:**

`endTime:` `GregorianCalendar` oggetto contenente la data e l'orario in cui l'utente ha completato il percorso

`id:` `int` intero contenente l'id dell'oggetto

`score:` `int` intero contenente il punteggio ottenuto dall'utente

`startTime:` `GregorianCalendar` oggetto contenente la data e l'orario in cui l'utente ha cominciato il percorso

- **Metodi:**

`getDuration() :` `long`

metodo per ottenere la durata della prova svolta dall'utente

- **Relazioni con altre classi:**

`CLIPS::client::data::PathProgress:` classe per il salvataggio in locale del progresso di un percorso svolto dall'utente

`CLIPS::client::data::PathResult:` classe che rappresenta i risultati di un percorso svolto dall'utente

`CLIPS::client::data::Proof:` classe che rappresenta una prova del percorso

3.3.2.21 `CLIPS::client::data::Proximity`

- **Descrizione:** classe che rappresenta in locale un beacon dedicato alla segnalazione della distanza dalla prossima prova

- **Utilizzo:** consente di salvare un beacon e la sua distanza dal beacon della prossima prova

- **Attributi:**

`beacon:` `Beacon` oggetto `Beacon` a cui fa riferimento l'oggetto

`percentage:` `double` rappresenta la distanza dell'utente dal Beacon in percentuale

`textToDisplay`: `string` stringa contenente il testo da mostrare all'utente quando si avvicina al Beacon

- **Relazioni con altre classi:**

`CLIPS::client::data::Beacon`: classe che rappresenta un beacon in locale

3.3.2.22 `CLIPS::client::data::Score`

- **Descrizione:** classe che rappresenta il risultato nella classifica dell'utente

- **Utilizzo:** permette di memorizzare il punteggio e la posizione in classifica dell'utente

- **Attributi:**

`position`: `int` intero contenente la posizione dello Score nella schermata

`score`: `int` intero contenente il punteggio ottenuto dall'utente

`username`: `string` stringa contenente lo username dell'utente che ha ottenuto il punteggio

3.3.2.23 `CLIPS::client::data::ScoringAlgorithmFactory`

- **Descrizione:** classe base per la creazione degli algoritmi di calcolo del punteggio di una determinata prova

- **Utilizzo:** fornisce il metodo per la creazione di algoritmi per il calcolo dei punteggi

- **Metodi:**

`createScoringAlgorithm(algorithmData) : void`
 si occupa di creare l'algoritmo per calcolare il punteggio

Argomenti:

* `algorithmData : LinearScoringAlgorithm` algoritmo per il punteggio

- **Relazioni con altre classi:**

`CLIPS::client::data::LinearScoringAlgorithm`: classe per calcolare il punteggio di una prova

3.3.2.24 `CLIPS::client::data::Step`

- **Descrizione:** classe che rappresenta in locale una frazione del percorso da svolgere

- **Utilizzo:** permette di salvare in locale le informazioni che rappresentano la ricerca della nuova prova

- **Attributi:**

`proof`: `Proof` oggetto contenente la prova da proporre all'utente

`proximities`: `List<Proximity>` lista contenente i Beacon vicini

`stopBeacon`: `Beacon` oggetto Beacon a cui fa riferimento l'oggetto

- **Relazioni con altre classi:**

CLIPS::client::data::Beacon: classe che rappresenta un beacon in locale

CLIPS::client::data::Proof: classe che rappresenta una prova del percorso

CLIPS::client::data::Proximity: classe che rappresenta in locale un beacon dedicato alla segnalazione della distanza dalla prossima prova

3.3.2.25 CLIPS::client::data::TestBuilder

- **Descrizione:** classe per la creazione degli oggetti Test partendo dagli oggetti JSON ricevuti dal server

- **Utilizzo:** permette di costruire gli oggetti Test partendo dagli oggetti JSON ricevuti dal server

- **Metodi:**

`createTest(testData) : Test`

metodo per la creazione di un oggetto Test a partire da un oggetto JSONObject

Argomenti:

- * `testData : JSONObject` oggetto contenente i dati del test in formato JSON ottenuti dal server

`getBiggerShapeTest(testData) : BiggerShapeTest`

metodo per la creazione di un oggetto BiggerShapeTest a partire da un oggetto JSONObject

Argomenti:

- * `testData : JSONObject` oggetto contenente i dati del test in formato JSON ottenuti dal server

`getMultipleChoiceTest(testData) : MultipleChoiceTest`

metodo per la creazione di un oggetto MultipleChoiceTest a partire da un oggetto JSONObject

Argomenti:

- * `testData : JSONObject` oggetto contenente i dati del test in formato JSON ottenuti dal server

`getTrueFalseTest(testData) : TrueFalseTest`

metodo per la creazione di un oggetto TrueFalseTest a partire da un oggetto JSONObject

Argomenti:

- * `testData : JSONObject` oggetto contenente i dati del test in formato JSON ottenuti dal server

- **Relazioni con altre classi:**

CLIPS::client::data::BiggerShape: classe che rappresenta un gioco in cui bisogna selezionare la figura più grande tra le due che appaiono

CLIPS::client::data::BiggerShapeTest: classe che rappresenta un test di BiggerShape

CLIPS::client::data::GameCollection: classe che rappresenta un insieme di test

CLIPS::client::data::GameCollection: classe che rappresenta un insieme di test

CLIPS::client::data::MultipleChoiceImageQuiz: classe che rappresenta una domanda a risposta multipla con immagine

CLIPS::client::data::MultipleChoiceImageQuiz: classe che rappresenta una domanda a risposta multipla con immagine

CLIPS::client::data::MultipleChoiceTest: classe che rappresenta un test di domande a risposta multiple

CLIPS::client::data::MultipleChoiceTest: classe che rappresenta un test di domande a risposta multiple

CLIPS::client::data::MultipleChoiceTextQuiz: classe che rappresenta una domanda a risposta multipla

CLIPS::client::data::MultipleChoiceTextQuiz: classe che rappresenta una domanda a risposta multipla

CLIPS::client::data::TrueFalseImageQuiz: classe che rappresenta una domanda vero o falso con immagine

CLIPS::client::data::TrueFalseImageQuiz: classe che rappresenta una domanda vero o falso con immagine

CLIPS::client::data::TrueFalseTest: classe che rappresenta un test di domande vero o falso

CLIPS::client::data::TrueFalseTest: classe che rappresenta un test di domande vero o falso

CLIPS::client::data::TrueFalseTextQuiz: classe controller che si occupa di interagire con TrueFalseView

CLIPS::client::data::TrueFalseTextQuiz: classe controller che si occupa di interagire con TrueFalseView

3.3.2.26 CLIPS::client::data::TrueFalseImageQuiz

- **Descrizione:** classe che rappresenta una domanda vero o falso con immagine
- **Utilizzo:** classe per rappresentare una domanda vero o falso in cui viene usata un'immagine per la formulazione del quesito
- **Attributi:**
 - image: string stringa contenente il nome dell'immagine

3.3.2.27 CLIPS::client::data::TrueFalseTest

- **Descrizione:** classe che rappresenta un test di domande vero o falso
- **Utilizzo:** classe per rappresentare un test di domande vero o falso, con immagine o senza
- **Attributi:**
 - questions: List<BiggerShape> lista contenente i quesiti da proporre all'utente
 - shuffleQuestions: boolean booleano per indicare se la lista delle domande va mescolata o meno

- **Relazioni con altre classi:**

CLIPS::client::data::TrueFalseTextQuiz: classe controller che si occupa di interagire con TrueFalseView

3.3.2.28 CLIPS::client::data::TrueFalseTextQuiz

- **Descrizione:** classe controller che si occupa di interagire con TrueFalseView

- **Utilizzo:** si occupa di gestire le interazioni dell'utente con TrueFalseView

- **Attributi:**

response: boolean booleano per indicare la risposta corretta

- **Metodi:**

check(userResponse) : boolean

metodo per controllare se la risposta data dall'utente è corretta

Argomenti:

* userResponse : boolean booleano contenente la risposta fornita dall'utente

3.3.2.29 CLIPS::client::data::Utility

- **Descrizione:** classe contenente vari metodi di utilità per altre classi

- **Utilizzo:** classe contenente vari metodi di utilità per altre classi

- **Metodi:**

addNearestBuilding(b, buildings, latitude, longitude) : Building[]

metodo per aggiungere un oggetto Building all'array contenente gli oggetti Building più vicini all'utente

Argomenti:

* b : Building oggetto Building da aggiungere all'array in caso sia sufficientemente vicino all'utente

* buildings : Building[] array contenente gli oggetti Building più vicini all'utente

* latitude : double latitudine a cui si trova l'utente

* longitude : double longitudine a cui si trova l'utente

calculateTotalScore(proofsResults) : int

metodo per calcolare lo score totale di un Path

Argomenti:

* proofsResults : List<ProofResult> lista contenente i risultati di ogni prova e da cui ricavare lo score totale

distanceToBuilding(b, user) : float

metodo per calcolare la distanza tra un edificio e l'utente

Argomenti:

* b : Building oggetto contenente le informazioni dell'edificio, compresa la posizione

* **user** : `Location` oggetto contenente la posizione dell'utente

insertBuilding(b, buildings, index) : Building[]
 metodo per inserire un oggetto `Building` in un array nella posizione `index`, eventualmente spostando nella posizione successiva l'elemento già presente

Argomenti:

* **b** : `Building` oggetto `Building` da inserire

* **buildings** : `Building[]` array in cui inserire l'oggetto `Building`

* **index** : `int` posizione in cui inserire l'oggetto `Building`

stringToGregorianCalendar(data, field) : GregorianCalendar
 metodo per convertire una stringa in un oggetto `GregorianCalendar`

Argomenti:

* **data** : `string` stringa formattata come un oggetto JSON contenente la data

* **field** : `string` stringa contenente il nome del campo da estrarre dalla stringa `data`

- **Relazioni con altre classi:**

`CLIPS::client::data::Building`: classe che rappresenta un edificio

`CLIPS::client::data::ProofResult`: classe che rappresenta il risultato di una prova

3.4 Componente `CLIPS::client::data::datamanager`

3.4.1 Informazioni sul package

- **Descrizione:** componente che gestisce i dati in locale
- **Padre:** `data`

3.4.2 Classi

3.4.2.1 `CLIPS::client::data::datamanager::textitSaveDataManager`

- **Descrizione:** classe astratta per il salvataggio di dati nel server
- **Utilizzo:** permette di effettuare il salvataggio dei dati nel server
- **Attributi:**
 - cx:** `Context` è una classe delle librerie base di Android che costituisce il cuore dell'applicazione, e quindi è necessaria per poter effettuare alcune operazioni, tra cui le richieste al server
 - listener:** `AbstractDataManagerListener<Boolean>` il listener che verrà usato per informare l'applicazione se il salvataggio dei dati ha avuto successo o meno
- **Metodi:**
 - execute() : void**
 metodo astratto che effettua la richiesta al server necessaria, definita dall'implementazione concreta delle sottoclassi

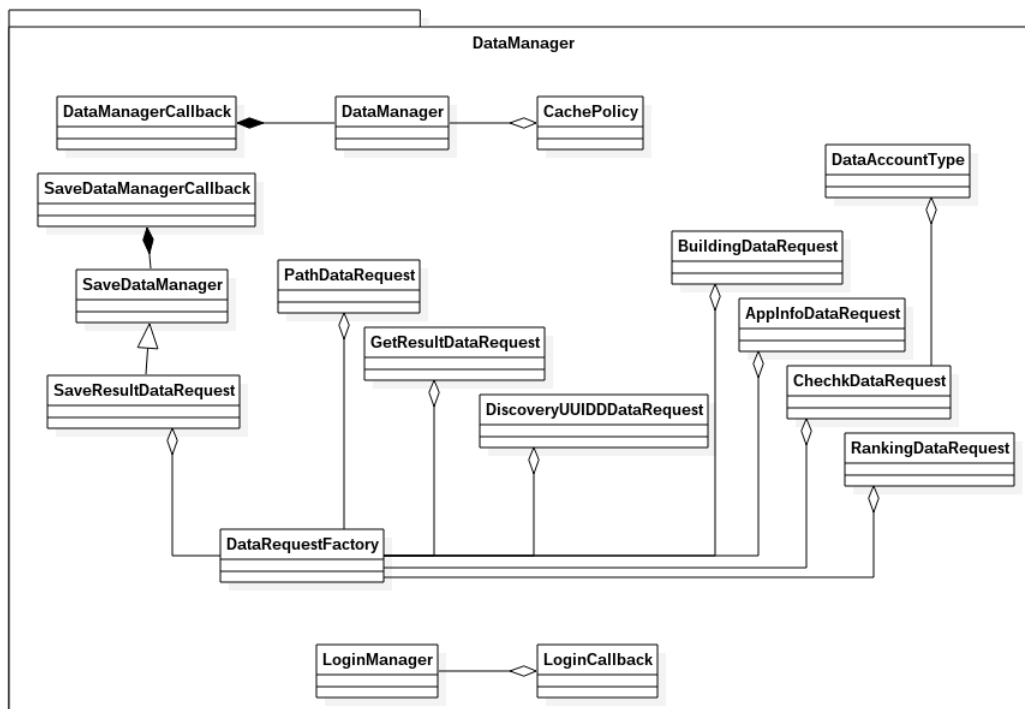


Figura 6: Schema sintetico package client::data::datamanager

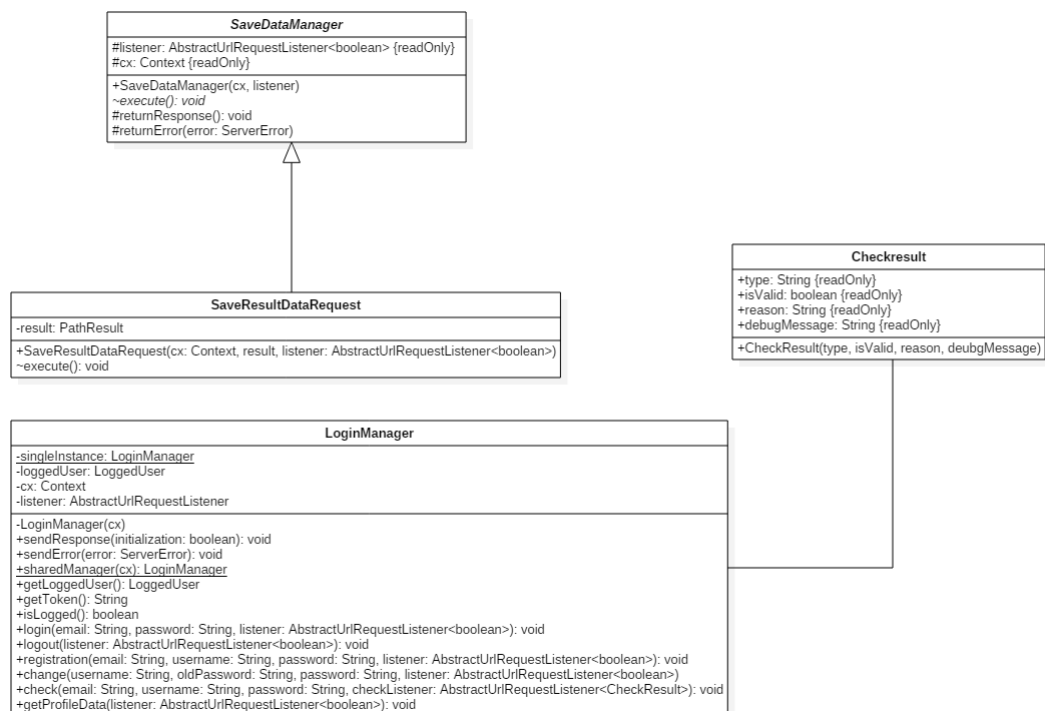


Figura 7: Prima parte schema package client::data::datamanager

```
returnError(error) : void
chiama il metodo onError del listener
```

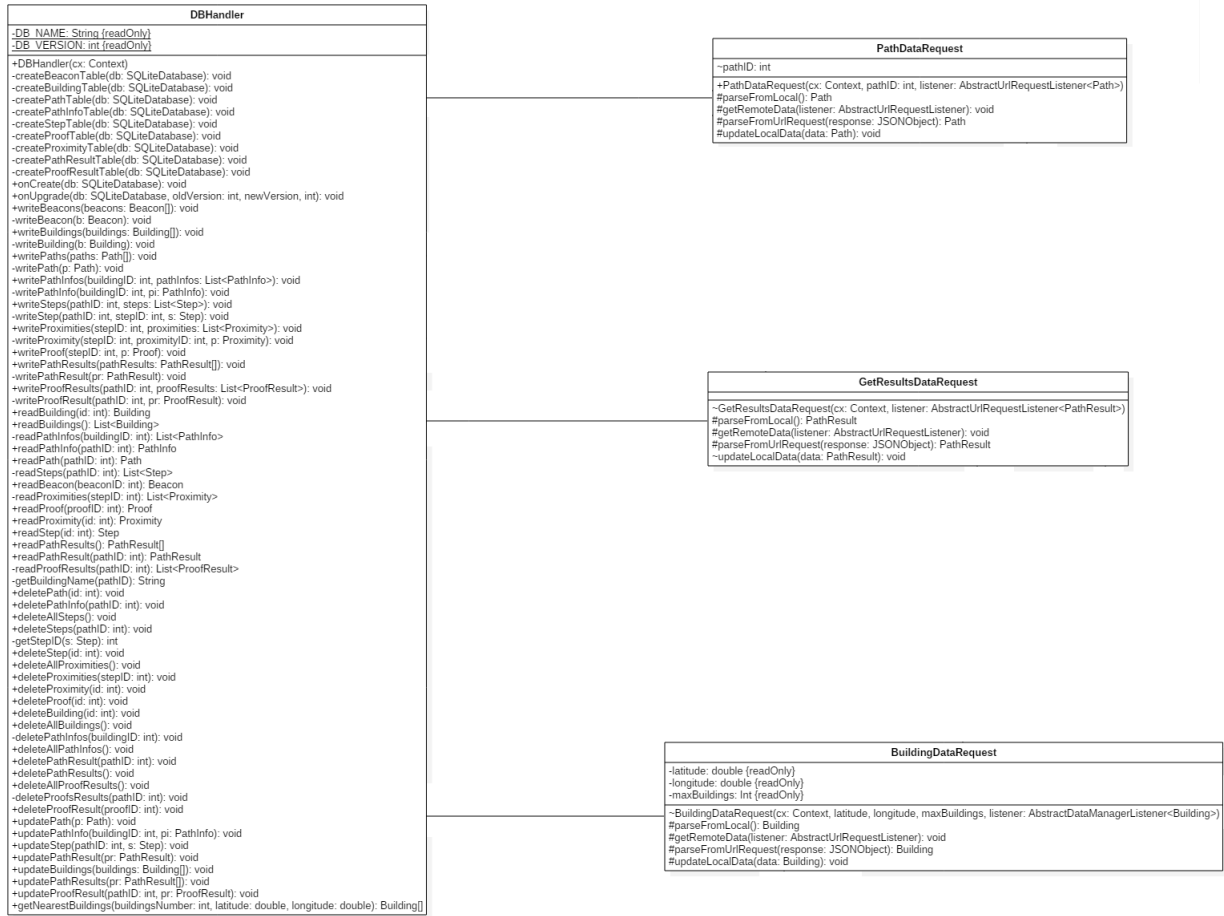


Figura 8: Seconda parte schema package client::data::datamanager

Argomenti:

* **error** : **ServerError** oggetto che contiene le informazioni relative all'errore rilevato dall'applicazione durante il salvataggio dei dati del percorso svolto

returnResponse() : **void**

chiama il metodo onResponse del listener e ritorna sempre "true", è un valore che serve solo per mantenere la consistenza della struttura del listener perché già chiamare il metodo indica che l'operazione è riuscita

• Relazioni con altre classi:

CLIPS::client::data::datamanager::AbstractDataManagerListener: interfaccia che definisce la struttura del listener astratto usato nel package datamanager

CLIPS::client::data::urlrequest::ServerError: classe che rappresenta l'errore che il client riceve quando la richiesta al server non ha successo

3.4.2.2 CLIPS::client::data::datamanager::AbstractDataManagerListener

- **Descrizione:** interfaccia che definisce la struttura del listener astratto usato nel package datamanager
- **Utilizzo:** permette alle classi che interagiscono con il datamanager di definire un proprio listener concreto da poter usare per ottenere le informazioni richieste

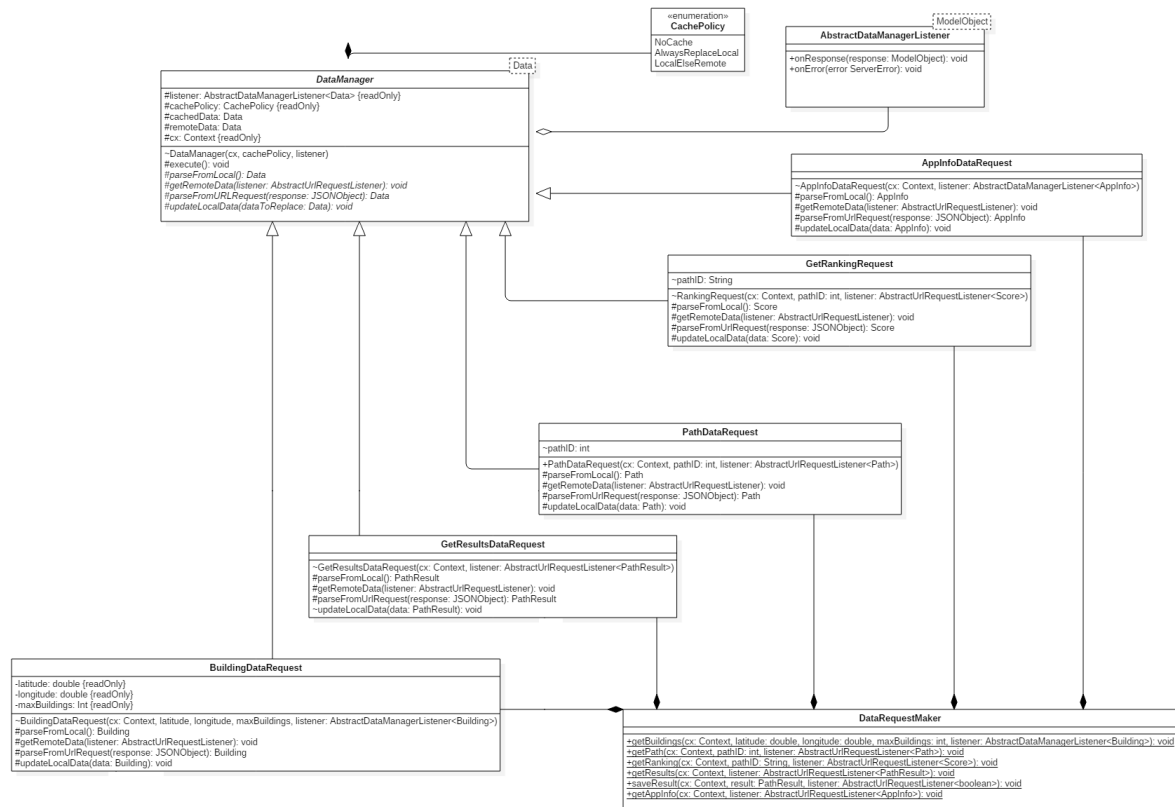


Figura 9: Terza parte schema package client::data::datamanager

- **Metodi:**

onError(error) : void

metodo che viene invocato quando si ottiene un errore mentre si cerca di ricavare il dato richiesto

Argomenti:

* **error** : **ServerError** l'errore ottenuto durante la ricerca del dato richiesto

onResponse(response) : void

metodo che viene invocato quando si ottiene con successo il dato richiesto

Argomenti:

* **response** : **ModelObject** è l'oggetto ricavato dal server o dal database locale

- **Relazioni con altre classi:**

CLIPS::client::data::urlrequest::ServerError: classe che rappresenta l'errore che il client riceve quando la richiesta al server non ha successo

3.4.2.3 CLIPS::client::data::datamanager::AppInfoDataRequest

- **Descrizione:** classe concreta che gestisce le richieste dei dati sulle info dell'app
- **Utilizzo:** permette di ottenere i dati riguardanti le informazioni generali dell'app
- **Relazioni con altre classi:**

CLIPS::client::data::datamanager::AbstractDataManagerListener: interfaccia che definisce la struttura del listener astratto usato nel package datamanager

CLIPS::client::data::urlrequest::AbstractUrlRequestListener: interfaccia che definisce la struttura del listener astratto usato nel package urlrequest

CLIPS::client::data::urlrequest::AbstractUrlRequestListener: interfaccia che definisce la struttura del listener astratto usato nel package urlrequest

CLIPS::client::data::AppInfo: classe per la rappresentazione delle informazioni dell'applicazione

CLIPS::client::data::urlrequest::RequestMaker: classe per la costruzione e l'invio di richieste al server, funziona come un'interfaccia dell'urlrequest per il resto dell'applicazione e quindi regola l'interazione con il package

CLIPS::client::data::urlrequest::ServerError: classe che rappresenta l'errore che il client riceve quando la richiesta al server non ha successo

3.4.2.4 CLIPS::client::data::datamanager::BuildingsDataRequest

- **Descrizione:** classe concreta che gestisce le richieste al server per ottenere i dati riguardanti gli edifici abilitati ad utilizzare l'applicazione

- **Utilizzo:** permette di ottenere i dati sugli edifici

- **Attributi:**

latitude: double la latitudine dell'utente

longitude: double la longitudine dell'utente

maxNumber: int il numero massimo di edifici richiesti o la distanza massimi entro cui devono essere presenti gli edifici restituiti, a seconda del valore di "searchByDistance"

searchByDistance: boolean indica se la richiesta al server deve ritornare gli edifici più vicini secondo un numero massimo o una distanza massima entro cui essi sono presenti

- **Relazioni con altre classi:**

CLIPS::client::data::datamanager::AbstractDataManagerListener: interfaccia che definisce la struttura del listener astratto usato nel package datamanager

CLIPS::client::data::urlrequest::AbstractUrlRequestListener: interfaccia che definisce la struttura del listener astratto usato nel package urlrequest

CLIPS::client::data::Building: classe che rappresenta un edificio

CLIPS::server::urlrequesthandler::DBHandler: classe che si occupa di gestire il DB e di ritornare un oggetto configurato della libreria knex

CLIPS::client::data::PathInfo: classe che si occupa di salvare in locale le informazioni generali di un percorso

CLIPS::client::data::urlrequest::RequestMaker: classe per la costruzione e l'invio di richieste al server, funziona come un'interfaccia dell'urlrequest per il resto dell'applicazione e quindi regola l'interazione con il package

CLIPS::client::data::urlrequest::ServerError: classe che rappresenta l'errore che il client riceve quando la richiesta al server non ha successo

3.4.2.5 CLIPS::client::data::datamanager::CheckResult

- **Descrizione:** classe che contiene il risultato del check per il dato specificato
- **Utilizzo:** contiene tutte le informazioni sul dato controllato dal server, ovvero se è valido e il messaggio d'errore quando ne viene rilevato uno
- **Attributi:**
 - `debugMessage:` `string` indica il motivo per cui il dato non è valido attraverso una spiegazione più tecnica, è destinato agli sviluppatori per aiutarli a trovare i bug nell'applicazione
 - `isValid:` `boolean` indica se il dato controllato ha un formato valido o no
 - `reason:` `string` indica il motivo per cui il dato non è valido
 - `type:` `string` il tipo di dato controllato, i possibili valori attesi sono "email", "password" e "username"

3.4.2.6 CLIPS::client::data::datamanager::DataManager

- **Descrizione:** classe astratta per la gestione della memorizzazione e del prelievo dei dati necessari per il funzionamento dell'applicazione
- **Utilizzo:** permette di prelevare i dati dal server o in locale ed eventualmente di salvarli nel database locale
- **Attributi:**
 - `cachedData:` `Data` oggetto generico in cui vengono memorizzati i dati letti dal database locale
 - `cachePolicy:` `CachePolicy` indica il comportamento dell'applicazione per la memorizzazione dei dati richiesti: "NoCache" indica che i dati vengono richiesti al server e mai salvati in locale; "AlwaysReplaceLocal" significa che i dati vengono richiesti al server, se vengono ricevuti allora i dati del database locale corrispondenti vengono sostituiti da quelli appena ottenuti, altrimenti vengono usati i dati salvati in locale; "LocalElseRemote" indica che vengono usati i dati del database locale, se non ci sono vengono richiesti al server
 - `cx:` `Context` è una classe delle librerie base di Android che costituisce il cuore dell'applicazione, e quindi è necessaria per poter effettuare alcune operazioni, tra cui le richieste al server
 - `listener:` `AbstractDataManagerListener<Building[]>` il listener che verrà usato per informare l'applicazione se l'operazione prevista ha avuto successo o meno
 - `remoteData:` `Data` oggetto generico in cui vengono memorizzati i dati ricevuti dal server
- **Metodi:**
 - `execute()` : `void`
metodo in cui vengono definite le operazioni da usare per i vari tipi di `cachePolicy`, per definire tali operazioni vengono usati i metodi astratti di `DataManager`
 - `getRemoteData(listener)` : `void`
metodo astratto che effettua la richiesta al server per ottenere i dati richiesti

Argomenti:

* **listener** : **AbstractUrlRequestListener** il listener che verrà usato per informare l'applicazione se la richiesta al server ha avuto successo o meno

parseFromLocal() : **Data**

metodo astratto dove vengono definite le operazioni per leggere i dati richiesti dal database locale

parseFromUrlRequest(response) : **Data**

metodo astratto per estrarre i dati richiesti dal JSON ottenuto con la richiesta al server

Argomenti:

* **response** : **JSONObject** l'oggetto JSON da cui estrarre i dati richiesti

updateLocalData(dataToReplace) : **void**

metodo astratto che cancella i dati del database locale per sostituirli con quelli ricevuti dal server

Argomenti:

* **dataToReplace** : **Data** i dati ottenuti dal server che devono rimpiazzare quelli salvati nel database locale

- **Relazioni con altre classi:**

CLIPS::client::data::datamanager::AbstractDataManagerListener: interfaccia che definisce la struttura del listener astratto usato nel package datamanager

CLIPS::client::data::urlrequest::AbstractUrlRequestListener: interfaccia che definisce la struttura del listener astratto usato nel package urlrequest

CLIPS::client::data::urlrequest::ServerError: classe che rappresenta l'errore che il client riceve quando la richiesta al server non ha successo

3.4.2.7 CLIPS::client::data::datamanager::DataRequestMaker

- **Descrizione:** classe che effettua le operazioni di ricerca e salvataggio dei dati

- **Utilizzo:** permette di effettuare operazioni di salvataggio dei dati nel server e di ricerca dei dati, richiedendoli al server o cercandoli in locale

- **Metodi:**

getAppInfo(cx, listener) : **void**

crea l'oggetto per ottenere le informazioni sull'applicazione dal server

Argomenti:

* **cx** : **Context** è una classe delle librerie base di Android che costituisce il cuore dell'applicazione, e quindi è necessaria per poter effettuare alcune operazioni, tra cui le richieste al server

* **listener** : **AbstractDataManagerListener<AppInfo>** il listener che verrà usato per informare l'applicazione se la ricerca dei dati ha avuto successo o meno

getBuildings(cx, latitude, listener, longitude, maxNumber, searchByDistance) : **void**

crea l'oggetto per ottenere la lista degli edifici da visualizzare nell'applicazione, prima cercandoli nel server e poi, in caso ci siano problemi, nel database locale

Argomenti:

- * **cx** : **Context** è una classe delle librerie base di Android che costituisce il cuore dell'applicazione, e quindi è necessaria per poter effettuare alcune operazioni, tra cui le richieste al server
- * **latitude** : **double** la latitudine attuale dell'utente
- * **listener** : **AbstractDataManagerListener<Building[]>** il listener che verrà usato per informare l'applicazione se la ricerca dei dati ha avuto successo o meno
- * **longitude** : **double** la longitudine attuale dell'utente
- * **maxNumber** : **int** valore che indica il numero massimo di edifici restituiti dal server o la loro distanza massima, a seconda del valore di "searchByDistance"
- * **searchByDistance** : **boolean** valore che indica se la chiamata deve restituire gli edifici in base ad un numero massimo o ad una distanza massima

getPath(cx, listener, pathID) : void

crea l'oggetto per ottenere i dati del percorso selezionato, prima cercandoli nel server e poi, in caso ci siano problemi, nel database locale

Argomenti:

- * **cx** : **Context** è una classe delle librerie base di Android che costituisce il cuore dell'applicazione, e quindi è necessaria per poter effettuare alcune operazioni, tra cui le richieste al server
- * **listener** : **AbstractDataManagerListener<Path>** il listener che verrà usato per informare l'applicazione se la ricerca dei dati ha avuto successo o meno
- * **pathID** : **int** l'ID del percorso selezionato

getRanking(cx, listener, pathID) : void

crea l'oggetto per ottenere la classifica del percorso selezionato dal server

Argomenti:

- * **cx** : **Context** è una classe delle librerie base di Android che costituisce il cuore dell'applicazione, e quindi è necessaria per poter effettuare alcune operazioni, tra cui le richieste al server
- * **listener** : **AbstractDataManagerListener<PlayerRanking[]>** il listener che verrà usato per informare l'applicazione se la ricerca dei dati ha avuto successo o meno
- * **pathID** : **int** l'ID del percorso selezionato

getResults(cx, listener) : void

crea l'oggetto per ottenere la lista di tutti i risultati dell'utente, prima cercandoli nel server e poi, in caso ci siano problemi, nel database locale

Argomenti:

- * **cx** : **Context** crea l'oggetto per ottenere la lista degli edifici da visualizzare nell'applicazione, prima cercandoli nel server e poi, in caso ci siano problemi, nel database locale
- * **listener** : **AbstractDataManagerListener<PathResult[]>** il listener che verrà usato per informare l'applicazione se la ricerca dei dati ha avuto successo o meno

saveResult(cx, listener, pathResult) : void

crea l'oggetto per salvare nel server il risultato appena ottenuto

Argomenti:

- * **cx** : **Context** è una classe delle librerie base di Android che costituisce il cuore dell'applicazione, e quindi è necessaria per poter effettuare alcune operazioni, tra cui le richieste al server
- * **listener** : **AbstractDataManagerListener<Boolean>** il listener che verrà usato per informare l'applicazione se la ricerca dei dati ha avuto successo o meno
- * **pathResult** : **PathResult** contiene il risultato del percorso appena svolto dall'utente

- **Relazioni con altre classi:**

CLIPS::client::data::datamanager::AbstractDataManagerListener: interfaccia che definisce la struttura del listener astratto usato nel package datamanager

CLIPS::client::data::AppInfo: classe per la rappresentazione delle informazioni dell'applicazione

CLIPS::client::data::datamanager::AppInfoDataRequest: classe concreta che gestisce le richieste dei dati sulle info dell'app

CLIPS::client::data::Building: classe che rappresenta un edificio

CLIPS::client::data::datamanager::BuildingsDataRequest: classe concreta che gestisce le richieste dei dati sugli edifici

CLIPS::client::data::datamanager::GetRankingDataRequest: classe concreta che gestisce le richieste dei dati della classifica

CLIPS::client::data::datamanager::GetResultsDataRequest: classe concreta che gestisce le richieste dei dati dei risultati

CLIPS::client::data::Path: classe che si occupa di salvare in locale i dati riguardanti un percorso

CLIPS::client::data::datamanager::PathDataRequest: classe concreta che gestisce le richieste dei dati sui percorsi

CLIPS::client::data::PathResult: classe che rappresenta i risultati di un percorso

CLIPS::client::data::datamanager::SaveResultDataRequest: classe concreta che effettua le richieste per salvare il risultato del percorso appena concluso

CLIPS::client::data::Score: classe che rappresenta il risultato nella classifica dell'utente

3.4.2.8 CLIPS::client::data::datamanager::DBHandler

- **Descrizione:** classe che si occupa della gestione del database locale
- **Utilizzo:** utilizzando questa classe è possibile interagire con il database locale per ottenere e salvare i dati desiderati

- **Attributi:**

DB_NAME : *stringvariabilecostantestaticacontenenteilnomedeldatabase locale*

- **Metodi:**

createBeaconTable(db) : void

metodo per la creazione della tabella Beacon all'interno del database locale

Argomenti:

* db : SQLiteDatabase oggetto necessario per l'interazione con il database locale

createBuildingTable(db) : void

metodo per la creazione della tabella Building all'interno del database locale

Argomenti:

* db : SQLiteDatabase oggetto necessario per l'interazione con il database locale

createPathInfoTable(db) : void

metodo per la creazione della tabella PathInfo all'interno del database locale

Argomenti:

* db : SQLiteDatabase oggetto necessario per l'interazione con il database locale

createPathResultTable(db) : void

metodo per la creazione della tabella PathResult all'interno del database locale

Argomenti:

* db : SQLiteDatabase oggetto necessario per l'interazione con il database locale

createPathTable(db) : void

metodo per la creazione della tabella Path all'interno del database locale

Argomenti:

* db : SQLiteDatabase oggetto necessario per l'interazione con il database locale

createProofResultTable(db) : void

metodo per la creazione della tabella ProofResult all'interno del database locale

Argomenti:

* db : SQLiteDatabase oggetto necessario per l'interazione con il database locale

createProofTable() : void

metodo per la creazione della tabella Proof all'interno del database locale

createProximityTable(db) : void

metodo per la creazione della tabella Proximity all'interno del database locale

Argomenti:

* db : SQLiteDatabase oggetto necessario per l'interazione con il database locale

createStepTable(db) : void

metodo per la creazione della tabella Step all'interno del database locale

Argomenti:

* db : SQLiteDatabase oggetto necessario per l'interazione con il database locale

deleteAllBuildings() : void

metodo per l'eliminazione di tutti gli oggetti Building dal database locale

deleteAllPathInfos() : void

metodo per l'eliminazione di tutti gli oggetti PathInfo dal database locale

deleteAllProofResults() : void

metodo per l'eliminazione di tutti gli oggetti ProofResult dal database locale

deleteAllSteps() : void

metodo per l'eliminazione di tutti gli oggetti Step salvati nel database locale

deleteBuilding(id) : void

metodo per l'eliminazione di un singolo oggetto Building dal database locale

Argomenti:

* **id : int** id dell'oggetto Building da eliminare

deletePath(id) : void

metodo per l'eliminazione di un oggetto Path dal database locale

Argomenti:

* **id : int** id dell'oggetto Path da eliminare

deletePathInfo(pathID) : void

metodo per l'eliminazione dell'oggetto PathInfo relativo ad un oggetto Path dal database locale

Argomenti:

* **pathID : int** id dell'oggetto Path da cui ricavare l'oggetto PathInfo da eliminare

deletePathInfos(buildingID) : void

metodo per l'eliminazione di tutti gli oggetti PathInfo relativi ad un oggetto Building dal database locale

Argomenti:

* **buildingID : int** id dell'oggetto Building da cui ricavare gli oggetti PathInfo da eliminare

deletePathResult(pathID) : void

metodo per l'eliminazione dell'oggetto PathResult relativo ad un oggetto Path

Argomenti:

* **pathID : int** id dell'oggetto Path da cui ricavare l'oggetto PathResult da eliminare

deletePathResults() : void

metodo per l'eliminazione di tutti gli oggetti PathResult dal database locale

deleteProof(id) : void

metodo per l'eliminazione di un singolo oggetto Proof

Argomenti:

* **id : int** id dell'oggetto Proof da eliminare

deleteProofResult(proofID) : void

metodo per l'eliminazione di un oggetto ProofResult relativo ad un oggetto Proof

Argomenti:

* **proofID : int** id dell'oggetto Proof da cui ricavare l'oggetto ProofResult da eliminare

deleteProofResults(pathID) : void

metodo per l'eliminazione degli oggetti ProofResult relativi ad un oggetto PathResult

Argomenti:

* **pathID** : **int** id dell'oggetto PathResult da cui ricavare gli oggetti ProofResult da eliminare

deleteProximities(stepID) : **void**

metodo per l'eliminazione di tutti gli oggetti Proximity dal database locale

Argomenti:

* **stepID** : **int** id dell'oggetto Step da cui ricavare gli oggetti Proximity da eliminare

deleteProximity(id) : **void**

metodo per l'eliminazione di un singolo oggetto Proximity

Argomenti:

* **id** : **int** id dell'oggetto Proximity da eliminare

deleteStep(id) : **void**

metodo per l'eliminazione di un oggetto Step dal database locale

Argomenti:

* **id** : **int** id dell'oggetto Step da eliminare

deleteSteps(pathID) : **void**

metodo per l'eliminazione di tutti gli oggetti Step relativi ad un oggetto Path dal database locale

Argomenti:

* **pathID** : **int** id dell'oggetto Path da cui ricavare gli oggetti Step da eliminare

getBuildingName(pathID) : **string**

metodo per ricavare il nome dell'edificio a cui è associato un oggetto Path conosciuto

Argomenti:

* **pathID** : **int** id dell'oggetto Path da cui ricavare il nome dell'edificio a cui l'oggetto è associato

getNearestBuildings(buildingNumber, latitude, longitude) : **Building[]**

metodo per ottenere un array contenente i buildingNumber oggetti Building più vicini rispetto alle variabili latitude e longitude fornite

Argomenti:

* **buildingNumber** : **int** numero degli oggetti Building più vicini all'utente rispetto alle sue coordinate

* **latitude** : **int** valore della latitudine a cui si trova l'utente

* **longitude** : **int** valore della longitudine a cui si trova l'utente

getStepID(s) : **int**

metodo per ricavare l'id di un oggetto Step conosciuto e salvato nel database locale

Argomenti:

* **s** : **Step** oggetto Step da cercare nel database locale per ricavarne l'id

onCreate() : **void**

metodo chiamato quando viene creato il database locale che a sua volta chiama tutti i metodi necessari per creare le tabelle del database

readBeacon(beaconID) : Beacon

metodo per la lettura di un singolo oggetto Beacon dal database locale

Argomenti:

* **beaconID : int** id dell'oggetto Beacon di cui si vogliono ottenere i dati

readBuilding(id) : Building

metodo per la lettura di un singolo oggetto Building dal database locale

Argomenti:

* **id : int** id dell'oggetto Building di cui si vogliono ottenere i dati

readBuildings() : List<Building>

metodo per la lettura di tutti gli oggetti Building presenti nel database locale

readPath(pathID) : Path

metodo per la lettura di un oggetto Path dal database locale

Argomenti:

* **pathID : int** id dell'oggetto Path di cui si vogliono ottenere i dati

readPathInfo(pathID) : PathInfo

metodo per la lettura dell'oggetto PathInfo relativo ad un oggetto Path

Argomenti:

* **pathID : int** id dell'oggetto Path da cui ricavare l'oggetto PathInfo

readPathInfos(buildingID) : List<PathInfo>

metodo per la lettura di tutti gli oggetti PathInfo relativi ad un oggetto Building presenti nel database locale

Argomenti:

* **buildingID : int** id dell'oggetto Building da cui ricavare la lista degli oggetti PathInfo

readPathResult(pathID) : PathResult

metodo per la lettura dell'oggetto PathResult relativo ad un oggetto Path

Argomenti:

* **pathID : int** id dell'oggetto Path da cui ricavare l'oggetto PathResult

readPathResults() : PathResult[]

metodo per la lettura di tutti gli oggetti PathResult dell'utente presenti nel database locale

readProof(proofID) : Proof

metodo per la lettura di un singolo oggetto Proof dal database locale

Argomenti:

* **proofID : int** id dell'oggetto Proof di cui si vogliono ottenere i dati

readProofResults(pathID) : List<ProofResult>

metodo per la lettura di tutti gli oggetti ProofResult relativi ad un oggetto Path presenti nel database locale

Argomenti:

* **pathID : int** id dell'oggetto Path da cui ricavare la lista degli oggetti ProofResult

readProximities(stepID) : List<Proximity>

metodo per la lettura di tutti gli oggetti Proximity relativi ad un oggetto Step presenti nel database locale

Argomenti:

* **stepID : int** id dell'oggetto Step da cui ricavare la lista di oggetti Step

readProximity(id) : Proximity

metodo per la lettura di un singolo oggetto Proximity dal database locale

Argomenti:

* **id : int** id dell'oggetto Proximity di cui si vogliono ottenere i dati

readStep(id) : Step

metodo per la lettura di un singolo oggetto Step dal database locale

Argomenti:

* **id : int** id dell'oggetto Step di cui si vogliono ottenere i dati

readSteps(pathID) : List<Step>

metodo per la lettura di tutti gli oggetti Step relativi ad un oggetto Path presenti nel database locale

Argomenti:

* **pathID : int** id dell'oggetto Path da cui ricavare la lista degli oggetti Step

updateBuildings(buildings) : void

metodo per aggiornare gli oggetti Building salvati nel database locale eliminando tutti gli oggetti Building presenti ed inserendone di nuovi

Argomenti:

* **buildings : Building[]** array contenente gli oggetti Building da inserire

updatePath(p) : void

metodo per l'aggiornamento di un oggetto Path nel database locale

Argomenti:

* **p : Path** oggetto da aggiornare nel database locale

updatePathInfo(buildingID, pi) : void

metodo per l'aggiornamento di un oggetto PathInfo nel database locale

Argomenti:

* **buildingID : int** id dell'oggetto Building a cui l'oggetto PathInfo da aggiornare è associato

* **pi : PathInfo** oggetto PathInfo da aggiornare

updatePathResult(pr) : void

metodo per l'aggiornamento di un oggetto PathResult nel database locale

Argomenti:

* **pr : PathResult** oggetto PathResult da aggiornare

updatePathResults(pathResults) : void

metodo per aggiornare gli oggetti PathResult salvati nel database locale eliminando tutti gli oggetti PathResult presenti ed inserendone di nuovi

Argomenti:

* **pathResults** : **PathResult[]** array contenente gli oggetti **PathResult** da inserire

updateProofResult(pathID, pr) : void

metodo per l'aggiornamento di un oggetto **ProofResult** nel database locale

Argomenti:

* **pathID** : **int** id dell'oggetto **PathResult** a cui l'oggetto **ProofResult** è associato

* **pr** : **ProofResult** oggetto **ProofResult** da aggiornare

updateStep(pathID, s) : void

metodo per l'aggiornamento di un oggetto **Step** nel database locale

Argomenti:

* **pathID** : **int** id dell'oggetto **Path** a cui l'oggetto **Step** da aggiornare è associato

* **s** : **Step** oggetto **Step** da aggiornare

writeBeacon(b) : void

metodo per l'inserimento di un singolo oggetto **Beacon** all'interno del database locale

Argomenti:

* **b** : **Beacon** oggetto **Beacon** da salvare nel database locale

writeBeacons(beacons) : void

metodo per l'inserimento multiplo di oggetti **Beacon** nel database locale

Argomenti:

* **beacons** : **Beacon[]** array contenente gli oggetti di tipo **Beacon** da inserire nel database locale

writeBuilding(b, id) : void

metodo per l'inserimento di un singolo oggetto **Building** all'interno del database locale

Argomenti:

* **b** : **Building** oggetto **Beacon** da salvare nel database locale

* **id** : **int** id dell'oggetto da inserire nel database

writeBuildings(buildings) : void

metodo per l'inserimento multiplo di oggetti **Building** nel database locale

Argomenti:

* **buildings** : **Building[]** array contenente gli oggetti di tipo **Building** da inserire nel database locale

writePath(p) : void

metodo per l'inserimento di un singolo oggetto **Path** all'interno del database locale

Argomenti:

* **p** : **Path[]** oggetto **Path** da salvare nel database locale

writePathInfo(buildingID, pi) : void

metodo per l'inserimento di un singolo oggetto **PathInfo** all'interno del database locale

Argomenti:

* **buildingID** : **int** id dell'oggetto **Building** a cui fa riferimento

* **pi** : **PathInfo** oggetto Beacon da salvare nel database locale

writePathInfos(buildingID, pathInfos) : void

metodo per l'inserimento multiplo di oggetti PathInfo nel database locale

Argomenti:

* **buildingID** : **int** id dell'oggetto Building a cui fa riferimento

* **pathInfos** : **List<PathInfo>** lista contenente gli oggetti di tipo PathInfo da inserire nel database locale

writePathResult(pr) : void

metodo per l'inserimento di un singolo oggetto PathResult all'interno del database locale

Argomenti:

* **pr** : **PathResult** oggetto PathResult da salvare nel database locale

writePathResults(pathresults) : void

metodo per l'inserimento multiplo di oggetti PathResult nel database locale

Argomenti:

* **pathresults** : **PathResult[]** array contenente gli oggetti di tipo PathResult da inserire nel database locale

writePaths(paths) : void

metodo per l'inserimento multiplo di oggetti Path nel database locale

Argomenti:

* **paths** : **Path[]** array contenente gli oggetti di tipo Beacon da inserire nel database locale

writeProof(p, stepID) : void

metodo per l'inserimento di un singolo oggetto Proof all'interno del database locale

Argomenti:

* **p** : **Proof** oggetto Proof da salvare nel database locale

* **stepID** : **int** id dell'oggetto Step a cui fa riferimento l'oggetto

writeProofResult(pathID, pr) : void

metodo per l'inserimento di un singolo oggetto ProofResult all'interno del database locale

Argomenti:

* **pathID** : **int** id dell'oggetto Path a cui fa riferimento l'oggetto

* **pr** : **ProofResult** oggetto ProofResult da salvare nel database locale

writeProofResults(pathID, proofResults) : void

metodo per l'inserimento multiplo di oggetti ProofResult nel database locale

Argomenti:

* **pathID** : **int** id dell'oggetto Path a cui fanno riferimento gli oggetti nella lista

* **proofResults** : **List<ProofResult>** lista contenente gli oggetti di tipo ProofResult da inserire nel database locale

writeProximities(proximities, stepID) : void

metodo per l'inserimento multiplo di oggetti Proximity nel database locale

Argomenti:

* **proximities** : `List<Proximity>` lista contenente gli oggetti di tipo `Proximity` da inserire nel database locale

* **stepID** : `int` id dell'oggetto `Step` a cui fanno riferimento gli oggetti nella lista

writeProximity(p, proximityID, stepID) : void

metodo per l'inserimento di un singolo oggetto `Proximity` all'interno del database locale

Argomenti:

* **p** : `Proximity` oggetto `Proximity` da salvare nel database locale

* **proximityID** : `int` id dell'oggetto da inserire nel database

* **stepID** : `int` id dell'oggetto `Step` a cui fa riferimento l'oggetto

writeStep(pathID, s, stepID) : void

metodo per l'inserimento di un singolo oggetto `Step` all'interno del database locale

Argomenti:

* **pathID** : `int` id dell'oggetto `Path` a cui fa riferimento l'oggetto `Step`

* **s** : `Step` oggetto `Step` da salvare nel database locale

* **stepID** : `int` id dell'oggetto da inserire nel database

writeSteps(pathID, steps) : void

metodo per l'inserimento multiplo di oggetti `Step` nel database locale

Argomenti:

* **pathID** : `int` id del `Path` a cui fanno riferimento gli oggetti `Step` nella lista

* **steps** : `List<Step>` lista contenente gli oggetti di tipo `Step` da inserire nel database locale

3.4.2.9 CLIPS::client::data::datamanager::GetRankingDataRequest

- **Descrizione:** classe concreta che gestisce le richieste dei dati della classifica

- **Utilizzo:** permette di ottenere i dati della classifica

- **Attributi:**

pathID: `int` l'ID del percorso di cui vengono richiesti i dati da ricavare

- **Relazioni con altre classi:**

CLIPS::client::data::datamanager::AbstractDataManagerListener: interfaccia che definisce la struttura del listener astratto usato nel package `datamanager`

CLIPS::client::data::urlrequest::AbstractUrlRequestListener: interfaccia che definisce la struttura del listener astratto usato nel package `urlrequest`

CLIPS::client::data::urlrequest::RequestMaker: classe per la costruzione e l'invio di richieste al server, funziona come un'interfaccia dell'`urlrequest` per il resto dell'applicazione e quindi regola l'interazione con il package

CLIPS::client::data::Score: classe che rappresenta il risultato nella classifica dell'utente

CLIPS::client::data::urlrequest::ServerError: classe che rappresenta l'errore che il client riceve quando la richiesta al server non ha successo

3.4.2.10 CLIPS::client::data::datamanager::GetResultsDataRequest

- **Descrizione:** classe concreta che gestisce le richieste dei dati dei risultati delle prove
- **Utilizzo:** permette di ottenere i dati dei risultati

- **Relazioni con altre classi:**

CLIPS::client::data::datamanager::AbstractDataManagerListener: interfaccia che definisce la struttura del listener astratto usato nel package datamanager

CLIPS::client::data::urlrequest::AbstractUrlRequestListener: interfaccia che definisce la struttura del listener astratto usato nel package urlrequest

CLIPS::server::urlrequesthandler::DBHandler: classe che si occupa di gestire il DB e di ritornare un oggetto configurato della libreria knex

CLIPS::client::data::PathResult: classe che rappresenta i risultati di un percorso

CLIPS::client::data::ProofResult: classe che rappresenta il risultato di una prova

CLIPS::client::data::urlrequest::RequestMaker: classe per la costruzione e l'invio di richieste al server, funziona come un'interfaccia dell'urlrequest per il resto dell'applicazione e regola l'interazione con il package

CLIPS::client::data::urlrequest::ServerError: classe che rappresenta l'errore che il client riceve quando la richiesta al server non ha successo

3.4.2.11 CLIPS::client::data::datamanager::LoginManager

- **Descrizione:** classe che gestisce le operazioni del profilo dell'utente
- **Utilizzo:** permette di effettuare il login, il logout, la registrazione, il check dei dati inseriti, l'aggiornamento dei dati salvati in locale e la modifica dei dati del profilo

- **Attributi:**

cx: Context è una classe delle librerie base di Android che costituisce il cuore dell'applicazione, e quindi è necessaria per poter effettuare alcune operazioni, tra cui le richieste al server

listener: AbstractDataManagerListener il listener che verrà usato per informare l'applicazione se l'operazione prevista ha avuto successo o meno

loggedUser: LoggedUser l'attributo contenente tutti i dati del profilo dell'utente

singleInstance: LoginManager l'attributo che permette di sfruttare il pattern design Singleton

- **Metodi:**

change(listener, oldPassword, password, username) : void
effettua la modifica dei dati dell'utente, controllandone la validità tramite l'apposita richiesta al server

Argomenti:

* **listener :** AbstractDataManagerListener<boolean> il listener che verrà usato per informare l'applicazione se l'operazione prevista ha avuto successo o meno

* **oldPassword** : **string** la vecchia password digitata dall'utente, serve per prevenire la modifica dei dati in caso di furto dell'account

* **password** : **string** la nuova password digitata dall'utente

* **username** : **string** il nuovo username digitato dall'utente

check(email, listener, password, username) : void

effettua la validazione dei dati digitati dall'utente, usando l'apposita richiesta al server

Argomenti:

* **email** : **string** l'indirizzo email digitato dall'utente

* **listener** : **AbstractDataManagerListener<CheckResult[]>** il listener che verrà usato per informare l'applicazione se l'operazione prevista ha avuto successo o meno

* **password** : **string** la password digitata dall'utente

* **username** : **string** l'username digitato dall'utente

getLoggedUser() : LoggedUser

metodo getter per ritornare loggedUser, altrimenti non sarebbe possibile accedervi al di fuori di LoginMnager

getProfileData(listener) : void

effettua l'aggiornamento dei dati del profilo dell'utente salvati in locale, tramite l'apposita chiamata al server

Argomenti:

* **listener** : **AbstractDataManagerListener<boolean>** il listener che verrà usato per informare l'applicazione se l'operazione prevista ha avuto successo o meno

getToken() : string

se esiste un token salvato restituisce il suo valore, altrimenti restituisce una stringa vuota

isLogged() : boolean

restituisce "true" se l'utente è loggato, ovvero se è salvato un token in locale, "false" altrimenti

login(email, listener, password) : void

effettua il login dell'utente, controllando la validità dei dati tramite l'apposita richiesta al server

Argomenti:

* **email** : **string** l'indirizzo email digitato dall'utente per il login

* **listener** : **AbstractDataManagerListener<boolean>** il listener che verrà usato per informare l'applicazione se l'operazione prevista ha avuto successo o meno

* **password** : **string** la password digitata dall'utente per effettuare il login

logout(listener) : void

effettua il logout dell'utente, informando il server tramite l'apposita richiesta

Argomenti:

* **listener** : **AbstractDataManagerListener<appinfo>** il listener che verrà usato per informare l'applicazione se l'operazione prevista ha avuto successo o meno

`registration(email, listener, password, username) : void`
 effettua la registrazione del profilo dell'utente, controllando la validità dei dati inseriti tramite l'apposita richiesta al server

Argomenti:

- * `email : string` l'indirizzo email digitato dall'utente per effettuare la registrazione
- * `listener : AbstractDataManagerListener<boolean>` il listener che verrà usato per informare l'applicazione se l'operazione prevista ha avuto successo o meno
- * `password : string` la password digitata dall'utente per effettuare la registrazione
- * `username : string` lo username digitato dall'utente per effettuare la registrazione

`sharedManager(cx) : LoginManager`

il metodo statico che permette di implementare il design pattern Singleton, dato che LoginManager può essere creato solo tramite questo metodo e quindi permette l'esistenza di un'unica copia

Argomenti:

- * `cx : Context` è una classe delle librerie base di Android che costituisce il cuore dell'applicazione, e quindi è necessaria per poter effettuare alcune operazioni, tra cui le richieste al server

- **Relazioni con altre classi:**

`CLIPS::client::data::datamanager::AbstractDataManagerListener`: interfaccia che definisce la struttura del listener astratto usato nel package datamanager

`CLIPS::client::data::urlrequest::AbstractUrlRequestListener`: interfaccia che definisce la struttura del listener astratto usato nel package urlrequest

`CLIPS::client::data::datamanager::CheckResult`: classe che contiene il risultato del check per il dato specificato

`CLIPS::client::data::LoggedUser`: classe che si occupa di memorizzare in locale i dati dell'utente loggato

`CLIPS::client::data::urlrequest::RequestMaker`: classe per la costruzione e l'invio di richieste al server, funziona come un'interfaccia dell'urlrequest per il resto dell'applicazione e regola l'interazione con il package

`CLIPS::client::data::urlrequest::ServerError`: classe che rappresenta l'errore che il client riceve quando la richiesta al server non ha successo

3.4.2.12 `CLIPS::client::data::datamanager::PathDataRequest`

- **Descrizione:** classe concreta che gestisce le richieste dei dati sui percorsi di un edificio
- **Utilizzo:** permette di ottenere i dati sui percorsi
- **Attributi:**
 - `pathID: int` l'ID del percorso selezionato
- **Relazioni con altre classi:**

`CLIPS::client::data::datamanager::AbstractDataManagerListener`: interfaccia che definisce la struttura del listener astratto usato nel package `datamanager`

`CLIPS::client::data::urlrequest::AbstractUrlRequestListener`: interfaccia che definisce la struttura del listener astratto usato nel package `urlrequest`

`CLIPS::client::data::Beacon`: classe che rappresenta un beacon in locale

`CLIPS::server::urlrequesthandler::DBHandler`: classe che si occupa di gestire il DB e di ritornare un oggetto configurato della libreria `knex`

`CLIPS::client::data::Path`: classe che si occupa di salvare in locale i dati riguardanti un percorso

`CLIPS::client::data::Proof`: classe che rappresenta una prova del percorso

`CLIPS::client::data::Proximity`: classe che rappresenta in locale un beacon indicato alla segnalazione della distanza dalla prossima prova

`CLIPS::client::data::urlrequest::ServerError`: classe che rappresenta l'errore che il client riceve quando la richiesta al server non ha successo

`CLIPS::client::data::urlrequest::ServerError`: classe che rappresenta l'errore che il client riceve quando la richiesta al server non ha successo

`CLIPS::client::data::Step`: classe che rappresenta in locale lo spostamento da una prova a quella successiva

3.4.2.13 `CLIPS::client::data::datamanager::SaveResultDataRequest`

- **Descrizione:** classe concreta che effettua le richieste per salvare il risultato del percorso appena concluso

- **Utilizzo:** permette di salvare in remoto il risultato del percorso appena svolto

- **Attributi:**

`result`: `PathResult` i dati del risultato da salvare nel server

- **Relazioni con altre classi:**

`CLIPS::client::data::datamanager::AbstractDataManagerListener`: interfaccia che definisce la struttura del listener astratto usato nel package `datamanager`

`CLIPS::client::data::urlrequest::AbstractUrlRequestListener`: interfaccia che definisce la struttura del listener astratto usato nel package `urlrequest`

`CLIPS::client::data::PathResult`: classe che rappresenta i risultati di un percorso

`CLIPS::client::data::urlrequest::RequestMaker`: classe per la costruzione e l'invio di richieste al server, funziona come un'interfaccia dell'`urlrequest` per il resto dell'applicazione e quindi regola l'interazione con il package

`CLIPS::client::data::urlrequest::ServerError`: classe che rappresenta l'errore che il client riceve quando la richiesta al server non ha successo

`CLIPS::client::data::urlrequest::ServerError`: classe che rappresenta l'errore che il client riceve quando la richiesta al server non ha successo

3.5 Componente CLIPS::client::data::urlrequest

3.5.1 Informazioni sul package

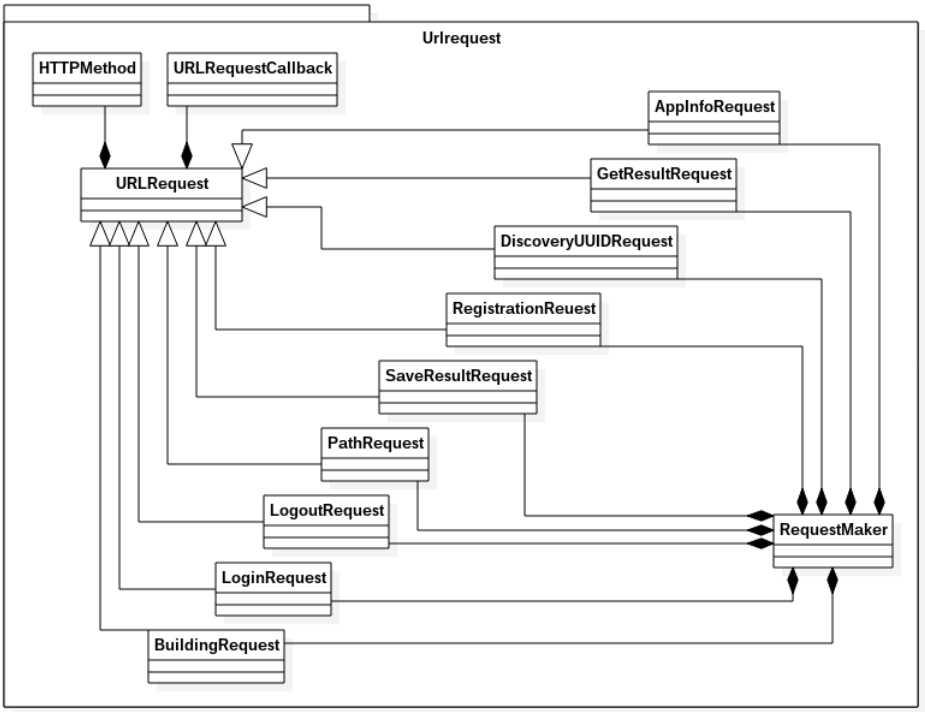


Figura 10: Schema sintetico package client::data::urlrequest

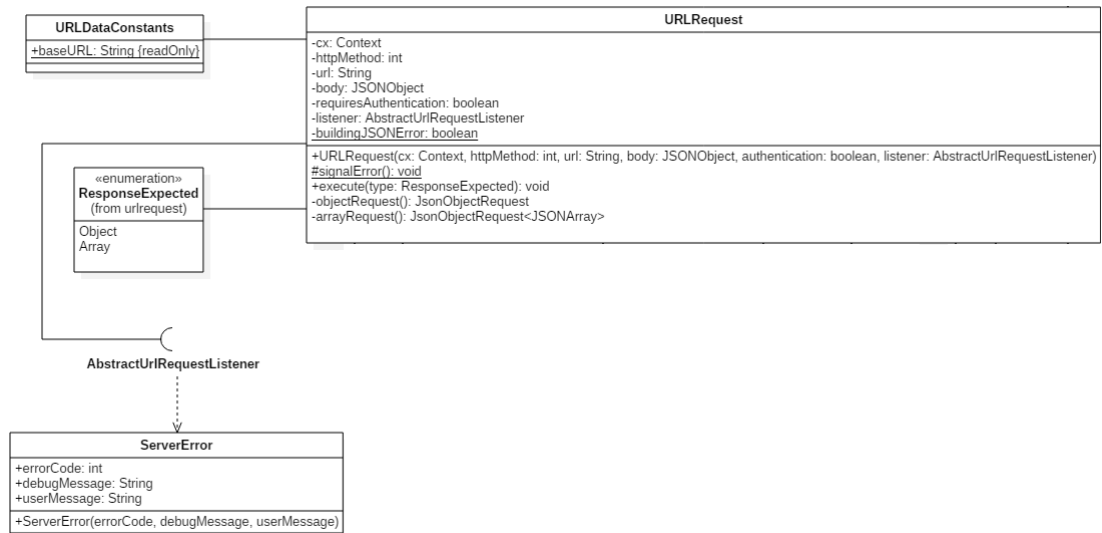


Figura 11: Prima parte schema package client::data::urlrequest

- **Descrizione:** componente che si occupa di effettuare le richieste al server
- **Padre:** data

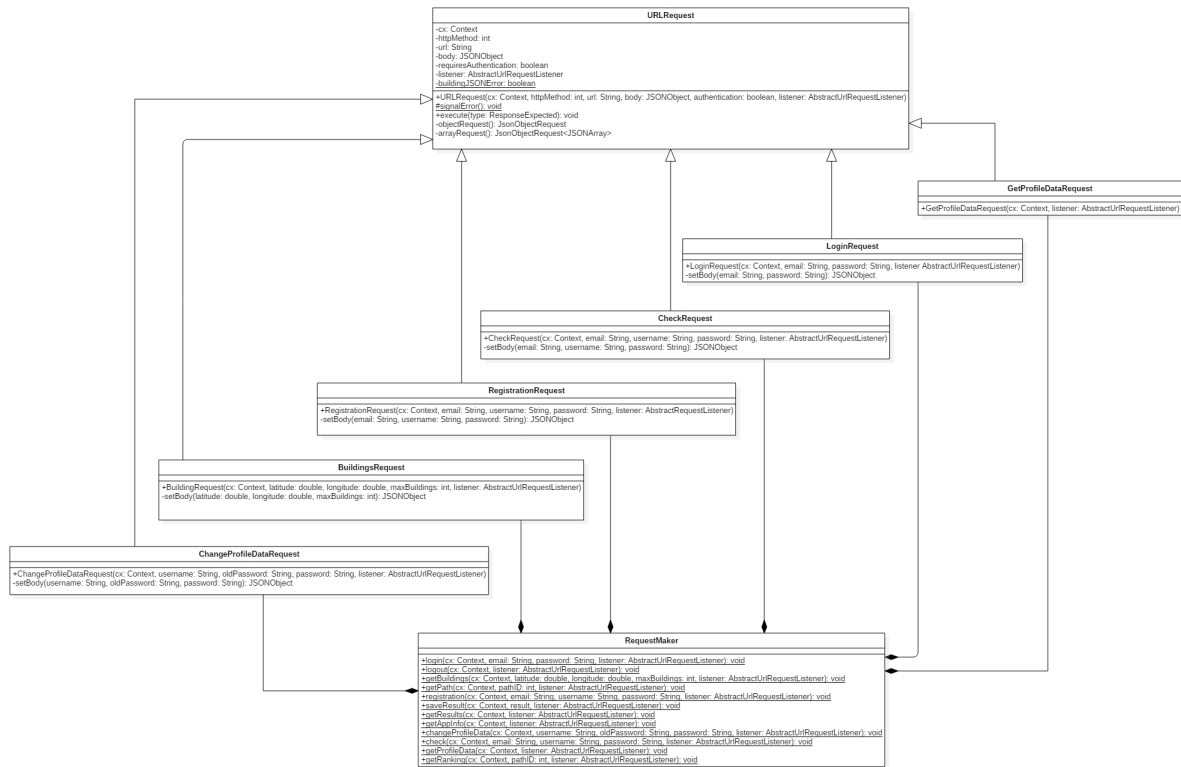


Figura 12: Seconda parte schema package client::data::urlrequest

3.5.2 Classi

3.5.2.1 CLIPS::client::data::urlrequest::AbstractUrlRequestListener

- **Descrizione:** interfaccia che definisce la struttura del listener astratto usato nel package urlrequest
- **Utilizzo:** permette alle classi che interagiscono con l'urlrequest di definire un proprio listener concreto da poter usare per le chiamate al server
- **Metodi:**

onError(error) : void

Metodo chiamato quando viene inviato un errore al listener

Argomenti:

- * **error : ServerError** Contiene i dati relativi all'errore segnalato, quindi il codice e una spiegazione del tipo di errore avuto

onResponse(response) : void

Metodo chiamato quando viene inviata la risposta al listener

Argomenti:

- * **response : JSONObject** Contiene i dati richiesti dal client

- **Relazioni con altre classi:**

CLIPS::client::data::urlrequest::ServerError: classe che rappresenta l'errore che il client riceve quando la richiesta al server non ha successo

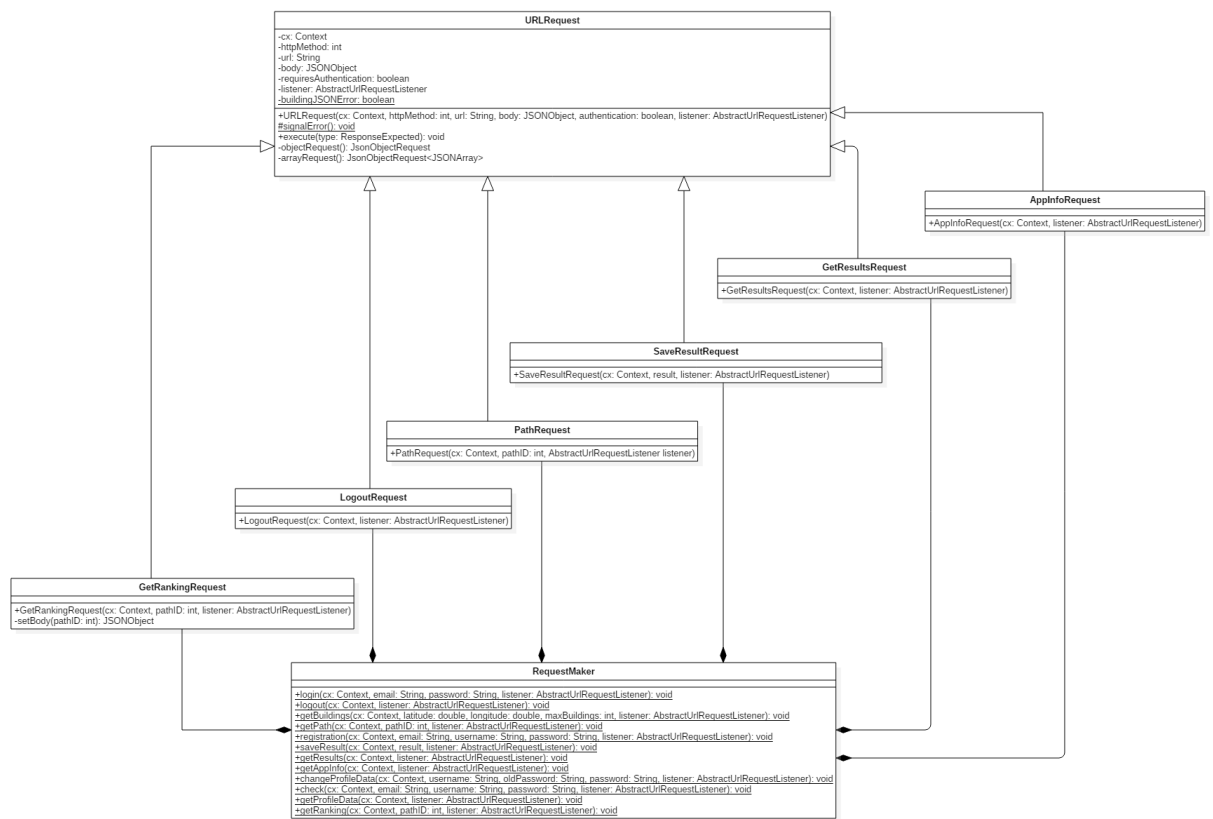


Figura 13: Terza parte schema package client::data::urlrequest

3.5.2.2 CLIPS::client::data::urlrequest::AppInfoRequest

- **Descrizione:** classe per la richiesta al server di dati sulle info dell'app, tra cui lo UUID dei beacon usati per l'applicazione e le informazioni sull'applicazione
- **Utilizzo:** permette di effettuare una richiesta al server di dati sulle info dell'app
- **Relazioni con altre classi:**

CLIPS::client::data::urlrequest::AbstractUrlRequestListener: interfaccia che definisce la struttura del listener astratto usato nel package urlrequest

CLIPS::client::data::urlrequest::URLDataConstants: classe che contiene i dati per effettuare le richieste al server che potrebbero variare nel tempo

3.5.2.3 CLIPS::client::data::urlrequest::BuildingsRequest

- **Descrizione:** classe per la richiesta di dati sugli edifici
- **Utilizzo:** permette di effettuare una richiesta al server di dati sugli edifici
- **Metodi:**

setBody(latitude, longitude, maxNumber, searchByDistance) : JSONObject
costruisce la sezione body dell'oggetto JSON, è un metodo di supporto

Argomenti:

- * **latitude** : **double** indica la latitudine attuale dell'utente, serve per trovare gli edifici più vicini
- * **longitude** : **double** indica la longitudine attuale dell'utente, serve per trovare gli edifici più vicini
- * **maxNumber** : **int** indica il numero massimo di edifici richiesto o la distanza massima entro cui essi sono situati, a seconda del valore di "searchByDistance"
- * **searchByDistance** : **boolean** indica se la ricerca degli edifici pi [U+FFFD] vicini deve essere effettuata in base ad un numero massimo o ad una distanza entro cui devono essere situati

- **Relazioni con altre classi:**

CLIPS::client::data::urlrequest::AbstractUrlRequestListener: interfaccia che definisce la struttura del listener astratto usato nel package urlrequest

CLIPS::client::data::urlrequest::URLDataConstants: classe che contiene i dati per effettuare le richieste al server che potrebbero variare nel tempo

3.5.2.4 CLIPS::client::data::urlrequest::ChangeProfileDataRequest

- **Descrizione:** classe per la richiesta al server di cambiare i dati dell'utente
- **Utilizzo:** permette di inviare la richiesta al server di cambiare i dati dell'utente, in particolare possono essere modificati solo lo username e la password
- **Metodi:**

setBody(oldPassword, password) : JSONObject
costruisce la sezione body dell'oggetto JSON, è un metodo di supporto

Argomenti:

- * **oldPassword** : **string** è la vecchia password dell'utente
- * **password** : **string** è la nuova password dell'utente

- **Relazioni con altre classi:**

CLIPS::client::data::urlrequest::AbstractUrlRequestListener: interfaccia che definisce la struttura del listener astratto usato nel package urlrequest

CLIPS::client::data::urlrequest::URLDataConstants: classe che contiene i dati per effettuare le richieste al server che potrebbero variare nel tempo

3.5.2.5 CLIPS::client::data::urlrequest::CheckRequest

- **Descrizione:** classe per la richiesta al server del controllo dei dati inseriti dall'utente
- **Utilizzo:** permette di inviare la richiesta al server per controllare se l'email, lo username o la password sono corretti
- **Metodi:**

setBody(password, username) : JSONObject
costruisce la sezione body dell'oggetto JSON, è un metodo di supporto

Argomenti:

* `password` : `string` è la password digitata dall'utente

* `username` : `string` è lo username digitato dall'utente

- **Relazioni con altre classi:**

`CLIPS::client::data::urlrequest::AbstractUrlRequestListener`: interfaccia che definisce la struttura del listener astratto usato nel package `urlrequest`

`CLIPS::client::data::urlrequest::URLDataConstants`: classe che contiene i dati per effettuare le richieste al server che potrebbero variare nel tempo

3.5.2.6 `CLIPS::client::data::urlrequest::ForgotPasswordRequest`

- **Descrizione:** Classe che effettua la richiesta di cambiare la password dimenticata dell'utente

- **Utilizzo:** Quando l'utente non ricorda la password può usare questa chiamata per ottenerne una generata casualmente dopo aver inserito l'indirizzo email dell'account da cambiare

- **Metodi:**

`setBody()` : `JSONObject`

Crea l'oggetto JSON da inserire nella chiamata per poter inviare l'email dell'account di cui bisogna cambiare la password con una generata automaticamente

- **Relazioni con altre classi:**

`CLIPS::client::data::urlrequest::URLDataConstants`: classe che contiene i dati per effettuare le richieste al server che potrebbero variare nel tempo

`CLIPS::client::data::urlrequest::URLRequest`: classe che si occupa delle richieste da inviare al server, fornisce un'implementazione delle richieste in base ai dati forniti alle sue classi derivate

3.5.2.7 `CLIPS::client::data::urlrequest::GetProfileDataRequest`

- **Descrizione:** classe per la richiesta al server di aggiornare in locale i dati dell'utente

- **Utilizzo:** permette di inviare la richiesta al server per aggiornare l'email e lo username dell'utente nel caso non fossero uguali a quelli salvati nel server

- **Relazioni con altre classi:**

`CLIPS::client::data::urlrequest::AbstractUrlRequestListener`: interfaccia che definisce la struttura del listener astratto usato nel package `urlrequest`

`CLIPS::client::data::urlrequest::URLDataConstants`: classe che contiene i dati per effettuare le richieste al server che potrebbero variare nel tempo

3.5.2.8 `CLIPS::client::data::urlrequest::GetRankingRequest`

- **Descrizione:** classe per la richiesta al server dei dati sulla classifica di un determinato percorso

- **Utilizzo:** permette di inviare la richiesta al server per ottenere i dati sulla classifica del percorso selezionato

- **Metodi:**

`setBody(pathID) : JSONObject`
costruisce la sezione body dell'oggetto JSON, è un metodo di supporto

Argomenti:

* `pathID : int` l'ID del percorso per cui bisogna chiedere la classifica

- **Relazioni con altre classi:**

`CLIPS::client::data::urlrequest::AbstractUrlRequestListener`: interfaccia che definisce la struttura del listener astratto usato nel package `urlrequest`

`CLIPS::client::data::urlrequest::URLDataConstants`: classe che contiene i dati per effettuare le richieste al server che potrebbero variare nel tempo

3.5.2.9 `CLIPS::client::data::urlrequest::GetResultsRequest`

- **Descrizione:** classe per la richiesta al server dei risultati salvati dall'utente in precedenza

- **Utilizzo:** permette di effettuare la richiesta al server dei risultati salvati dall'utente in precedenza

- **Relazioni con altre classi:**

`CLIPS::client::data::urlrequest::AbstractUrlRequestListener`: interfaccia che definisce la struttura del listener astratto usato nel package `urlrequest`

`CLIPS::client::data::urlrequest::URLDataConstants`: classe che contiene i dati per effettuare le richieste al server che potrebbero variare nel tempo

3.5.2.10 `CLIPS::client::data::urlrequest::LoginRequest`

- **Descrizione:** classe per la richiesta di login al server

- **Utilizzo:** permette di effettuare una richiesta di login al server

- **Metodi:**

`setBody(email, password) : JSONObject`
costruisce la sezione body dell'oggetto JSON, è un metodo di supporto

Argomenti:

* `email : string` l'email inserita dall'utente per effettuare il login

* `password : string` la password inserita dall'utente per effettuare il login

- **Relazioni con altre classi:**

`CLIPS::client::data::urlrequest::AbstractUrlRequestListener`: interfaccia che definisce la struttura del listener astratto usato nel package `urlrequest`

`CLIPS::client::data::urlrequest::URLDataConstants`: classe che contiene i dati per effettuare le richieste al server che potrebbero variare nel tempo

3.5.2.11 CLIPS::client::data::urlrequest::LogoutRequest

- **Descrizione:** classe per la richiesta di logout al server
- **Utilizzo:** permette di effettuare una richiesta di logout al server
- **Relazioni con altre classi:**

CLIPS::client::data::urlrequest::AbstractUrlRequestListener: interfaccia che definisce la struttura del listener astratto usato nel package urlrequest

CLIPS::client::data::urlrequest::URLDataConstants: classe che contiene i dati per effettuare le richieste al server che potrebbero variare nel tempo

3.5.2.12 CLIPS::client::data::urlrequest::PathRequest

- **Descrizione:** classe per la richiesta al server dei dati sul percorso selezionato, in particolare fornisce tutte le informazioni necessarie per poter svolgere il percorso
- **Utilizzo:** permette di richiedere al server i dati sul percorso selezionato
- **Relazioni con altre classi:**

CLIPS::client::data::urlrequest::AbstractUrlRequestListener: interfaccia che definisce la struttura del listener astratto usato nel package urlrequest

CLIPS::client::data::urlrequest::URLDataConstants: classe che contiene i dati per effettuare le richieste al server che potrebbero variare nel tempo

3.5.2.13 CLIPS::client::data::urlrequest::RegistrationRequest

- **Descrizione:** classe per la richiesta di registrazione al server
- **Utilizzo:** permette di effettuare una richiesta di registrazione al server
- **Metodi:**

setBody(email, password, username) : JSONObject
costruisce la sezione body dell'oggetto JSON, è un metodo di supporto

Argomenti:

- * email : string l'email inserita dall'utente per effettuare la registrazione
- * password : string la password inserita dall'utente per effettuare la registrazione
- * username : string lo username inserito dall'utente per effettuare la registrazione

- **Relazioni con altre classi:**

CLIPS::client::data::urlrequest::AbstractUrlRequestListener: interfaccia che definisce la struttura del listener astratto usato nel package urlrequest

CLIPS::client::data::urlrequest::URLDataConstants: classe che contiene i dati per effettuare le richieste al server che potrebbero variare nel tempo

3.5.2.14 CLIPS::client::data::urlrequest::RequestMaker

- **Descrizione:** classe per la costruzione e l'invio di richieste al server, funziona come un'interfaccia dell'urlrequest per il resto dell'applicazione e quindi regola l'interazione con il package
- **Utilizzo:** permette la costruzione e l'invio di richieste al server
- **Metodi:**

`changeProfileData(cx, listener, oldPassword, password, username) : void`
crea l'oggetto per chiedere al server di verificare e cambiare i dati inseriti dall'utente

Argomenti:

- * **cx** : **Context** è una classe delle librerie base di Android che costituisce il cuore dell'applicazione, e quindi è necessaria per poter effettuare alcune operazioni, tra cui le richieste al server
- * **listener** : **AbstractUrlRequestListener** il listener che verrà usato per informare l'applicazione se la richiesta al server ha avuto successo o meno
- * **oldPassword** : **string** la vecchia password, richiesta e digitata dall'utente per evitare cambi di dati a sua insaputa
- * **password** : **string** la nuovo password digitata dall'utente
- * **username** : **string** il nuovo username digitato dall'utente

`check(cx, email, listener, password, username) : void`
crea l'oggetto per chiedere al server di verificare se i dati inseriti dall'utente sono validi

Argomenti:

- * **cx** : **Context** è una classe delle librerie base di Android che costituisce il cuore dell'applicazione, e quindi è necessaria per poter effettuare alcune operazioni, tra cui le richieste al server
- * **email** : **string** l'indirizzo email inserito dall'utente
- * **listener** : **AbstractUrlRequestListener** il listener che verrà usato per informare l'applicazione se la richiesta al server ha avuto successo o meno
- * **password** : **string** la password inserita dall'utente
- * **username** : **string** lo username inserito dall'utente

`getAppInfo(cx, listener) : void`
crea l'oggetto per ottenere dal server le informazioni sull'applicazione

Argomenti:

- * **cx** : **Context** è una classe delle librerie base di Android che costituisce il cuore dell'applicazione, e quindi è necessaria per poter effettuare alcune operazioni, tra cui le richieste al server
- * **listener** : **AbstractUrlRequestListener** il listener che verrà usato per informare l'applicazione se la richiesta al server ha avuto successo o meno

`getBuildings(cx, latitude, listener, longitude, maxBuildings) : void`
crea l'oggetto per ottenere dal server i dati degli edifici abilitati più vicini alla sua posizione

Argomenti:

- * **cx** : **Context** è una classe delle librerie base di Android che costituisce il cuore dell'applicazione, e quindi è necessaria per poter effettuare alcune operazioni, tra cui le richieste al server
- * **latitude** : **double** la latitudine dell'utente, viene usata per trovare gli edifici più vicini a lui
- * **listener** : **AbstractUrlRequestListener** il listener che verrà usato per informare l'applicazione se la richiesta al server ha avuto successo o meno
- * **longitude** : **double** la longitudine dell'utente, viene usata per trovare gli edifici più vicini a lui
- * **maxBuildings** : **int** il numero massimo di edifici richiesti, viene usato per non dover inviare inutilmente i dati di tutti gli edifici

getPath(cx, listener, pathID) : void

crea l'oggetto per ottenere dal server i dati del percorso selezionato

Argomenti:

- * **cx** : **Context** è una classe delle librerie base di Android che costituisce il cuore dell'applicazione, e quindi è necessaria per poter effettuare alcune operazioni, tra cui le richieste al server
- * **listener** : **AbstractUrlRequestListener** il listener che verrà usato per informare l'applicazione se la richiesta al server ha avuto successo o meno
- * **pathID** : **int** l'ID del percorso, verrà usato dal server per capire quale percorso è stato selezionato

getProfileData(cx, listener) : void

crea l'oggetto per ottenere dal server i dati attuali del profilo dell'utente

Argomenti:

- * **cx** : **Context** è una classe delle librerie base di Android che costituisce il cuore dell'applicazione, e quindi è necessaria per poter effettuare alcune operazioni, tra cui le richieste al server
- * **listener** : **AbstractUrlRequestListener** il listener che verrà usato per informare l'applicazione se la richiesta al server ha avuto successo o meno

getRanking(cx, listener, pathID) : void

crea la classe per ottenere dal server i dati sulla classifica del percorso selezionato

Argomenti:

- * **cx** : **Context** è una classe delle librerie base di Android che costituisce il cuore dell'applicazione, e quindi è necessaria per poter effettuare alcune operazioni, tra cui le richieste al server
- * **listener** : **AbstractUrlRequestListener** il listener che verrà usato per informare l'applicazione se la richiesta al server ha avuto successo o meno
- * **pathID** : **int** l'id del percorso selezionato dall'utente per ottenere la relativa classifica

getResults(cx, listener) : void

crea l'oggetto per ottenere dal server i risultati conseguiti dall'utente

Argomenti:

- * **cx** : **Context** [U+FFFD] una classe delle librerie base di Android che costituisce il cuore dell'applicazione, e quindi [U+FFFD] necessaria per poter effettuare alcune operazioni, tra cui le richieste al server

* **listener** : **AbstractUrlRequestListener** il listener che verrà [U+FFFD] [U+FFFD] usato per informare l'applicazione se la richiesta al server ha avuto successo o meno

login(cx, email, listener, password) : void

crea l'oggetto per chiedere al server di verificare i dati inseriti dall'utente per il login

Argomenti:

* **cx** : **Context** è una classe delle librerie base di Android che costituisce il cuore dell'applicazione, e quindi è necessaria per poter effettuare alcune operazioni, tra cui le richieste al server

* **email** : **string** l'indirizzo email inserito dall'utente per effettuare il login

* **listener** : **AbstractUrlRequestListener** il listener che verrà usato per informare l'applicazione se la richiesta al server ha avuto successo o meno

* **password** : **string** la password inserita dall'utente per effettuare il login

logout(cx, listener) : void

crea l'oggetto per notificare al server che l'utente ha effettuato il logout

Argomenti:

* **cx** : **Context** è una classe delle librerie base di Android che costituisce il cuore dell'applicazione, e quindi è necessaria per poter effettuare alcune operazioni, tra cui le richieste al server

* **listener** : **AbstractUrlRequestListener** il listener che verrà usato per informare l'applicazione se la richiesta al server ha avuto successo o meno

registration(cx, email, listener, password, username) : void

crea l'oggetto per chiedere al server di verificare e registrare il profilo inserito dall'utente

Argomenti:

* **cx** : **Context** è una classe delle librerie base di Android che costituisce il cuore dell'applicazione, e quindi è necessaria per poter effettuare alcune operazioni, tra cui le richieste al server

* **email** : **string** l'indirizzo email che l'utente ha inserito per registrarsi

* **listener** : **AbstractUrlRequestListener** il listener che verrà usato per informare l'applicazione se la richiesta al server ha avuto successo o meno

* **password** : **string** la password che l'utente ha inserito per registrarsi

* **username** : **string** lo username che l'utente ha inserito per registrarsi

saveResult(cx, listener, result) : void

crea l'oggetto per salvare nel server il risultato appena ottenuto

Argomenti:

* **cx** : **Context** è una classe delle librerie base di Android che costituisce il cuore dell'applicazione, e quindi è necessaria per poter effettuare alcune operazioni, tra cui le richieste al server

* **listener** : **AbstractUrlRequestListener** il listener che verrà usato per informare l'applicazione se la richiesta al server ha avuto successo o meno

* **result** : **PathResult** l'oggetto che contiene i risultati da salvare nel server

• **Relazioni con altre classi:**

`CLIPS::client::data::urlrequest::AbstractUrlRequestListener`: interfaccia che definisce la struttura del listener astratto usato nel package `urlrequest`

`CLIPS::client::data::urlrequest::AppInfoRequest`: classe per la richiesta al server di dati sulle info dell'app, tra cui lo UUID dei beacon usati per l'applicazione e le informazioni sull'applicazione

`CLIPS::client::data::urlrequest::BuildingsRequest`: classe per la richiesta di dati sugli edifici

`CLIPS::client::data::urlrequest::ChangeProfileDataRequest`: classe per la richiesta al server di cambiare i dati dell'utente

`CLIPS::client::data::urlrequest::CheckRequest`: classe per la richiesta al server del controllo dei dati inseriti dall'utente

`CLIPS::client::data::urlrequest::GetProfileDataRequest`: classe per la richiesta al server di aggiornare in locale i dati dell'utente

`CLIPS::client::data::urlrequest::GetRankingRequest`: classe per la richiesta al server dei dati sulla classifica di un determinato percorso

`CLIPS::client::data::urlrequest::GetResultsRequest`: classe per la richiesta al server dei risultati salvati dall'utente in precedenza

`CLIPS::client::data::urlrequest::LoginRequest`: classe per la richiesta di login al server

`CLIPS::client::data::urlrequest::LogoutRequest`: classe per la richiesta di logout al server

`CLIPS::client::data::urlrequest::PathRequest`: classe per la richiesta al server dei dati sul percorso selezionato, in particolare fornisce tutte le informazioni necessarie per poter svolgere il percorso

`CLIPS::client::data::PathResult`: classe che rappresenta i risultati di un percorso

`CLIPS::client::data::urlrequest::RegistrationRequest`: classe per la richiesta di registrazione al server

`CLIPS::client::data::urlrequest::SaveResultRequest`: classe per la richiesta al server di salvataggio del risultato del percorso appena finito

3.5.2.15 `CLIPS::client::data::urlrequest::SaveResultRequest`

- **Descrizione:** classe per la richiesta al server di salvataggio del risultato del percorso appena finito
- **Utilizzo:** permette di effettuare una richiesta al server di salvataggio del risultato del percorso appena finito
- **Metodi:**
 - `setBody(result) : JSONObject`
costruisce la sezione body dell'oggetto JSON, è un metodo di supporto
 - Argomenti:*
 - * `result : PathResult` il risultato da salvare nel server
- **Relazioni con altre classi:**

`CLIPS::client::data::urlrequest::AbstractUrlRequestListener`: interfaccia che definisce la struttura del listener astratto usato nel package `urlrequest`

`CLIPS::client::data::PathResult`: classe che rappresenta i risultati di un percorso

`CLIPS::client::data::ProofResult`: classe che rappresenta il risultato di una prova

`CLIPS::client::data::urlrequest::URLDataConstants`: classe che contiene i dati per effettuare le richieste al server che potrebbero variare nel tempo

3.5.2.16 `CLIPS::client::data::urlrequest::ServerError`

- **Descrizione:** classe che rappresenta l'errore che il client riceve quando la richiesta al server non ha successo
- **Utilizzo:** permette di rappresentare l'errore ricevuto in una forma più semplice da usare nel resto dell'applicazione
- **Attributi:**
 - `debugMessage`: `string` il messaggio destinato agli sviluppatori, per aiutarli ad individuare dove è avvenuto l'errore
 - `errorCode`: `int` il codice dell'errore segnalato
 - `userMessage`: `string` il messaggio destinato all'utente, per comunicargli che l'applicazione ha riscontrato un errore

3.5.2.17 `CLIPS::client::data::urlrequest::URLDataConstants`

- **Descrizione:** classe che contiene i dati per effettuare le richieste al server che potrebbero variare nel tempo
- **Utilizzo:** permette di gestire facilmente eventuali cambiamenti futuri dei dati necessari ad effettuare le richieste al server
- **Attributi:**
 - `baseURL`: `string` Contiene l'url del server, facilita l'aggiornamento del client a seguito di modifiche del server

3.5.2.18 `CLIPS::client::data::urlrequest::URLRequest`

- **Descrizione:** classe che si occupa delle richieste da inviare al server, fornisce un'implementazione delle richieste in base ai dati forniti alle sue classi derivate
- **Utilizzo:** permette di effettuare tutte le richieste al server necessarie per far funzionare l'applicazione
- **Attributi:**
 - `body`: `JSONObject` è la parte della richiesta al server che contiene i dati necessari per portare a compimento la chiamata

buildingJSONError: `bool` valore booleano che indica se è avvenuto o meno un errore durante la costruzione del body, viene impostato dalle classi derivate di `URLRequest`

cx: `Context` è una classe delle librerie base di Android che costituisce il cuore dell'applicazione, e quindi è necessaria per poter effettuare alcune operazioni, tra cui le richieste al server

httpMethod: `int` è un'enumerazione che indica quale metodo HTTP eseguire, di fatto gli unici valori usati sono GET e POST

listener: `AbstractUrlRequestListener` il listener che verrà usato per informare l'applicazione se la richiesta al server ha avuto successo o meno

requiresAuthentication: `bool` indica se la chiamata necessita di aggiungere il token salvato all'header dell'oggetto inviato

url: `string` l'url completo a cui mandare la richiesta

- **Metodi:**

arrayRequest() : `JsonRequest<JSONArray>`

è un metodo di supporto che crea una chiamata la cui risposta è un array JSON

execute(type) : `void`

crea ed esegue la chiamata al server in base ai campi dati salvati

Argomenti:

* **type** : `ResponseExpected` indica se la chiamata si aspetta di ricevere, in caso di successo, un oggetto o un array JSON

objectRequest() : `JSONObjectRequest`

è un metodo di supporto che crea una chiamata la cui risposta è un oggetto JSON

signalError() : `void`

cambia il valore di `buildingJSONError` in "true" e segnala quindi all'`URLRequest` che c'è stato un errore in fase di costruzione di "body", viene invocato dalle classi derivate di `URLRequest`

- **Relazioni con altre classi:**

`CLIPS::client::data::urlrequest::AbstractUrlRequestListener`: interfaccia che definisce la struttura del listener astratto usato nel package `urlrequest`

`CLIPS::client::data::datamanager::LoginManager`: classe che gestisce le operazioni del profilo dell'utente

`CLIPS::client::data::urlrequest::ServerError`: classe che rappresenta l'errore che il client riceve quando la richiesta al server non ha successo

3.6 Componente `CLIPS::client::pathprogress`

3.6.1 Informazioni sul package

- **Descrizione:** componente che gestisce i dati del percorso e salva i risultati ottenuti nelle prove mentre si sta giocando
- **Padre:** `client`

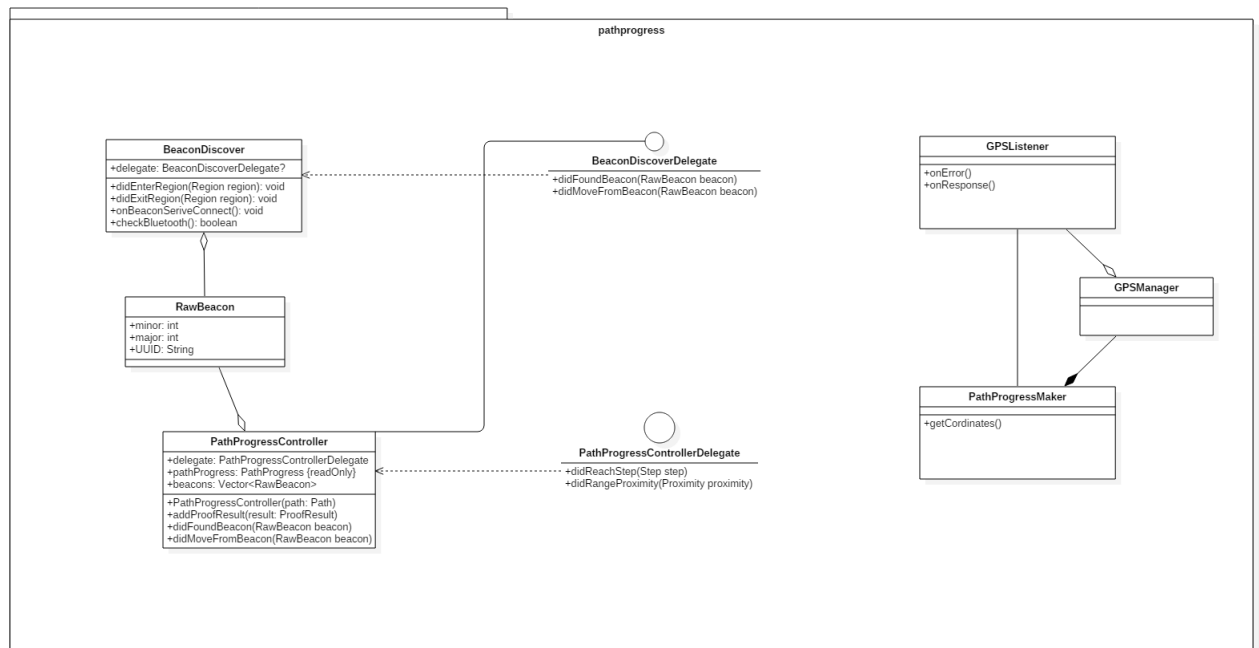


Figura 14: Schema package client::pathprogress

3.6.2 Classi

3.6.2.1 CLIPS::client::pathprogress::BeaconDiscover

- **Descrizione:** classe che si interfaccia direttamente con i beacon fisici
- **Utilizzo:** permette di rilevare i beacon
- **Attributi:**
 - delegate: BeaconDiscoverDelegate riferimento al delegate
- **Metodi:**
 - onBeaconServiceConnect()() : void
metodo chiamato una volta che il dispositivo dell'utente si è collegato ad un Beacon

3.6.2.2 CLIPS::client::pathprogress::BeaconDiscoverDelegate

- **Descrizione:** interfaccia utilizzata per notificare se il dispositivo è entrato o uscito dal raggio di un beacon
- **Utilizzo:** permette di notificare se si è entrati o usciti dal raggio di un beacon
- **Metodi:**
 - didFoundBeacon(rawBeacon) : void
segnala quando è stato trovato un beacon
 - Argomenti:
 - * rawBeacon : RawBeacon rappresenta un beacon fisico

`didMoveFromBeacon(rawBeacon) : void`
segnala che si è usciti dal raggio di un beacon

Argomenti:

* `rawBeacon` : `RawBeacon` rappresenta un beacon fisico

- **Relazioni con altre classi:**

`CLIPS::client::pathprogress::PathProgressController`: classe che gestisce l'avanzamento di un percorso

3.6.2.3 `CLIPS::client::pathprogress::GPSListener`

- **Descrizione:** interfaccia che definisce la struttura del listener astratto usato nel package `pathprogress`

- **Utilizzo:** permette alle classi che interagiscono con il `pathprogress` di definire un proprio listener concreto da poter usare per ottenere le informazioni richieste

- **Metodi:**

`onError(error) : void`

Restituisce il messaggio di errore quando ne avviene uno durante la rilevazione delle coordinate dell'utente con il GPS

Argomenti:

* `error` : `ServerError` L'errore restituito durante la rilevazione del GPS

`onResponse(latitude, longitude) : void`

Restituisce le coordinate GPS dell'utente rilevate come risposta

Argomenti:

* `latitude` : `double` La latitudine dell'utente rilevata dal GPS

* `longitude` : `double` La longitudine dell'utente rilevata dal GPS

- **Relazioni con altre classi:**

`CLIPS::client::data::urlrequest::ServerError`: classe che rappresenta l'errore che il client riceve quando la richiesta al server non ha successo

3.6.2.4 `CLIPS::client::pathprogress::GPSManager`

- **Descrizione:** classe che rileva la posizione dell'utente tramite il GPS

- **Utilizzo:** appena viene creato l'oggetto comunica con i satelliti GPS tramite i Google Play Services, appena riceve una posizione sufficientemente precisa restituisce le coordinate ricevute

- **Relazioni con altre classi:**

`CLIPS::client::pathprogress::GPSListener`: interfaccia che definisce la struttura del listener astratto usato nel package `pathprogress`

3.6.2.5 CLIPS::client::pathprogress::PathProgressController

- **Descrizione:** classe che gestisce l'avanzamento di un percorso
- **Utilizzo:** permette di gestire i dati dell'avanzamento di un percorso
- **Attributi:**
 - `delegate:` `PathProgressControllerDelegate` riferimento al delegate
 - `path:` `Path` rappresenta un il path di cui deve gestire l'avanzamento
 - `pathProgress:` `PathProgress` rappresenta il path progress nel database locale
 - `rawBeacons:` `Vector<RawBeacon>` rappresenta un vettore di raw beacon
- **Metodi:**
 - `addProofResult(result) : void`
aggiunge un risultato di una prova nel database locale
Argomenti:
 - * `result` : `ProofResult` risultato da aggiungere
 - `didFoundBeacon(rawBeacon) : void`
implementazione del metodo dell'interfaccia che stabilisce cosa fare quando viene trovato un beacon
Argomenti:
 - * `rawBeacon` : `RawBeacon` rappresenta un beacon fisico
 - `didMoveFromBeacon(rawBeacon) : void`
implementazione del metodo dell'interfaccia che stabilisce cosa fare qualora si esca dal raggio di un beacon
Argomenti:
 - * `rawBeacon` : `RawBeacon` rappresenta un beacon fisico
 - `getPathProgress() : PathProgress`
ritorna il pathprogress
- **Relazioni con altre classi:**
 - `CLIPS::client::pathprogress::BeaconDiscoverDelegate:` interfaccia utilizzata per notificare se il dispositivo è entrato o uscito dal raggio di un beacon
 - `CLIPS::client::data::PathProgress:` classe per il salvataggio in locale del progresso di un percorso
 - `CLIPS::client::pathprogress::PathProgressControllerDelegate:` interfaccia utilizzata per notificare se si è in presenza del beacon del prossimo step o di uno del proximity
 - `CLIPS::client::pathprogress::RawBeacon:` classe che rappresenta un beacon semplice

3.6.2.6 CLIPS::client::pathprogress::PathProgressControllerDelegate

- **Descrizione:** interfaccia utilizzata per notificare se si è in presenza del beacon del prossimo step o di uno del proximity

- **Utilizzo:** notifica se si è nel raggio del beacon del prossimo step o di uno di proximity
- **Metodi:**

`didRangeProximity(proximity) : void`
 segnala che si è nel raggio di un beacon del proximity

Argomenti:

* `proximity` : `Proximity` rappresenta il proximity nel quale ci si trova

`didReachStep(step) : void`
 segnala che si è nel raggio del beacon del prossimo step

Argomenti:

* `step` : `Step` rappresenta il step in cui ci si trova

3.6.2.7 CLIPS::client::pathprogress::PathProgressMaker

- **Descrizione:** classe che funziona come interfaccia tra le classi per rilevare la posizione con il GPS e il resto dell'applicazione
- **Utilizzo:** quando all'applicazione serve la posizione dell'utente invoca il relativo metodo della classe
- **Metodi:**

`getCoordinates(cx, listener) : void`
 costruisce l'oggetto per rilevare le coordinate dell'utente con il GPS

Argomenti:

* `cx` : `Context` è una classe delle librerie base di Android che costituisce il cuore dell'applicazione, e quindi è necessaria per poter effettuare alcune operazioni, tra cui le richieste al server

* `listener` : `GPSListener` il listener che verrà usato per informare l'applicazione se l'operazione prevista ha avuto successo o meno

- **Relazioni con altre classi:**

`CLIPS::client::pathprogress::GPSListener`: interfaccia che definisce la struttura del listener astratto usato nel package `pathprogress`

`CLIPS::client::pathprogress::GPSManager`: classe che rileva la posizione dell'utente tramite il GPS

3.6.2.8 CLIPS::client::pathprogress::RawBeacon

- **Descrizione:** classe che rappresenta un beacon semplice
- **Utilizzo:** utilizzato per rappresentare un beacon fisico
- **Attributi:**
 - `major` : `int` rappresenta il major del beacon fisico
 - `minor` : `int` rappresenta il minor del beacon fisico
 - `UUID` : `string` rappresenta l'UUID del beacon fisico

3.7 Componente CLIPS::client::viewController

3.7.1 Informazioni sul package

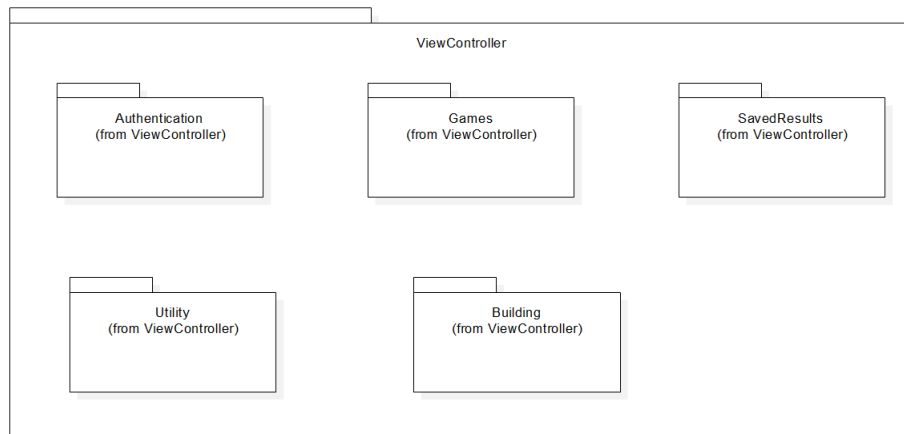


Figura 15: Schema package client::viewController

- **Descrizione:** componente che raggruppa tutte le view ed i controller relativi alle view
- **Padre:** client
- **Package contenuti:**
 - CLIPS::client::viewController::authentication: componente che si occupa di gestire l'autenticazione dell'utente
 - CLIPS::client::viewController::building: componente che gestisce le informazioni e le interazioni dell'utente con gli edifici abilitati
 - CLIPS::client::viewController::games: componente che gestisce le prove che l'utente deve completare all'interno di un percorso
 - CLIPS::client::viewController::savedresults: componente che raggruppa le le view e i controller dei risultati salvati e delle classifiche
 - CLIPS::client::viewController::utility: componente che raggruppa le view generali dell'app

3.7.2 Classi

3.7.2.1 CLIPS::client::viewController::ResultAdapter

- **Descrizione:** classe che permette la corretta visualizzazione dei risultati di un utente
- **Utilizzo:** viene usata da SavedResultActivity

3.8 Componente CLIPS::client::viewController::authentication

3.8.1 Informazioni sul package

- **Descrizione:** componente che si occupa di gestire l'autenticazione dell'utente

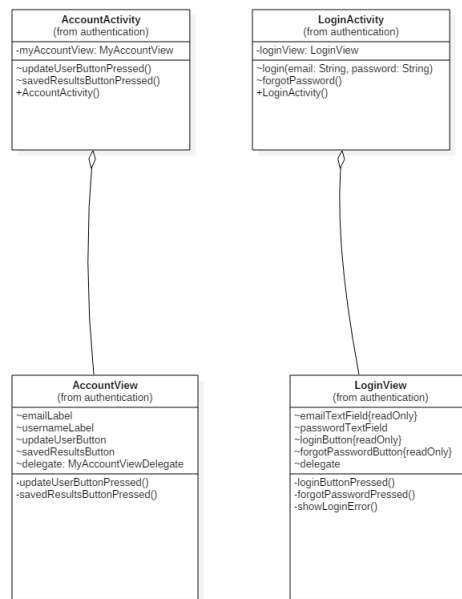


Figura 16: Prima parte schema package client::viewController::authentication

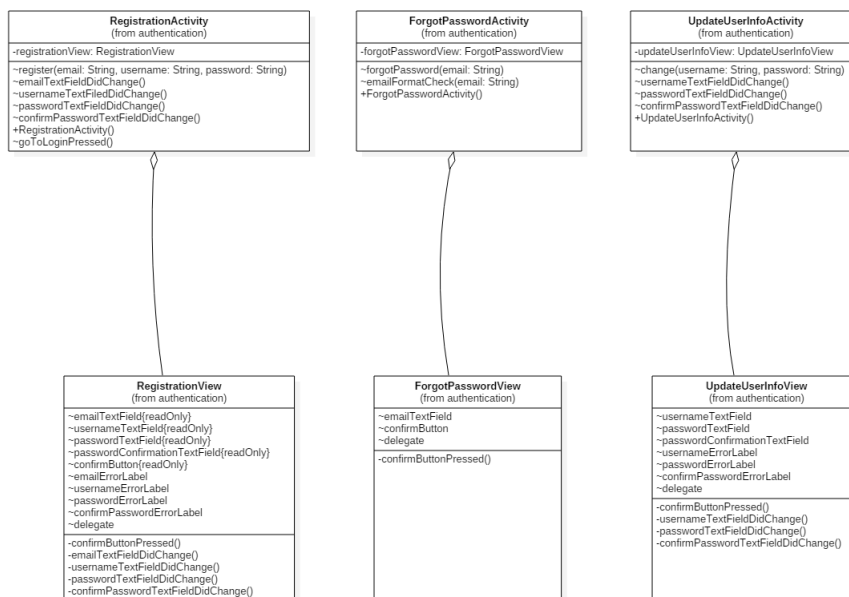


Figura 17: Seconda parte schema package client::viewController::authentication

· **Padre:** viewController

3.8.2 Classi

3.8.2.1 CLIPS::client::viewController::authentication::AccountActivity

- **Descrizione:** classe controller che si occupa di interagire con AccountView
- **Utilizzo:** si occupa di gestire le interazioni dell'utente con AccountView
- **Metodi:**

`onCreate() : void`

si occupa di inizializzare l'activity con la relativa view

`setButtons() : void`

si occupa di inizializzare i collegamenti dei bottoni presenti nella view

3.8.2.2 CLIPS::client::viewController::authentication::AccountView

- **Descrizione:** classe che rappresenta la schermata delle informazioni dell'utente
- **Utilizzo:** permette all'utente di visualizzare le informazioni del proprio profilo
- **Attributi:**
 - `emailLabel: TextView` rappresenta l'email dell'utente
 - `savedResultsButton: Button` rappresenta il bottone per visualizzare i risultati salvati
 - `updateUserButton: Button` rappresenta il bottone per modificare le informazioni dell'utente
 - `usernameLabel: TextView` indica l'username dell'utente
- **Relazioni con altre classi:**
 - `CLIPS::client::viewController::authentication::AccountActivity:` classe controller che si occupa di interagire con `AccountView`

3.8.2.3 CLIPS::client::viewController::authentication::ChangeProfileActivity

- **Descrizione:** permette all'utente di cambiare le proprie credenziali
- **Utilizzo:** l'utente grazie a `ChangeProfileView` potrà cambiare le proprie credenziali

3.8.2.4 CLIPS::client::viewController::authentication::ConfirmRegistration

- **Descrizione:** classe che interagisce con `RegistrationActivity` e permette all'utente di visualizzare i propri dati di registrazione
- **Utilizzo:** l'utente può visualizzare i dati inseriti in precedenza
- **Relazioni con altre classi:**
 - `CLIPS::client::viewController::authentication::RegistrationActivity:` classe controller che si occupa di interagire con `RegistrationView`

3.8.2.5 CLIPS::client::viewController::authentication::ForgotPasswordActivity

- **Descrizione:** classe controller che si occupa di interagire con `ForgotPasswordView`
- **Utilizzo:** si occupa di gestire le interazioni dell'utente con `ForgotPasswordView`
- **Metodi:**

`onCreate(SavedBundleInstance, savedInstanceState) : void`

si occupa di inizializzare l'activity con la relativa view

Argomenti:

- * `SavedBundleInstance` : `Bundle` contiene i dati dell'istanza creata o di un'attività precedente
- * `savedInstanceState` : `Bundle` contiene i dati dell'istanza creata o di un'attività precedente

- **Relazioni con altre classi:**

`CLIPS::client::viewController::authentication::ForgotPasswordView`: classe che si occupa della visualizzazione della schermata per la richiesta di una nuova password

3.8.2.6 `CLIPS::client::viewController::authentication::ForgotPasswordView`

- **Descrizione:** classe che si occupa della visualizzazione della schermata per la richiesta di una nuova password
- **Utilizzo:** consente all'utente di inserire la mail per ricevere una nuova password
- **Attributi:**
 - `confirmButton`: `Button` rappresenta il bottone per la conferma dell'email inserita
 - `emailTextField`: `EditText` rappresenta l'email da inserire da parte dell'utente

3.8.2.7 `CLIPS::client::viewController::authentication::LoginActivity`

- **Descrizione:** classe controller che si occupa di interagire con `LoginView`
- **Utilizzo:** si occupa di gestire le interazioni dell'utente con `LoginView`
- **Metodi:**
 - `setButtons()` : `void`
si occupa di inizializzare i collegamenti dei bottoni presenti nella view
- **Relazioni con altre classi:**
 - `CLIPS::client::viewController::authentication::LoginView`: classe che si occupa della visualizzazione della schermata per il login

3.8.2.8 `CLIPS::client::viewController::authentication::LoginView`

- **Descrizione:** classe che si occupa della visualizzazione della schermata per il login
- **Utilizzo:** consente all'utente di inserire i propri dati per effettuare il login
- **Attributi:**
 - `emailTextField`: `EditText` rappresenta l'email da inserire da parte dell'utente
 - `forgotPasswordButton`: `Button` rappresenta il button per la password dimenticata

`loginButton`: `Button` indica il button per effettuare il login

`passwordTextField`: `EditText` rappresenta la password da inserire da parte dell'utente

- **Metodi:**

`forgotPasswordPressed()` : `void`

invia un evento il quale notifica che il bottone di password dimenticata è stato premuto

`loginButtonPressed()` : `void`

invia un evento di bottone del login premuto

3.8.2.9 `CLIPS::client::viewController::authentication::RecoverPassword`

- **Descrizione:** classe che rappresenta l'avvenuto recupero della password di un utente
- **Utilizzo:** view che permette all'utente di sapere che la sua password è stata recuperata correttamente

3.8.2.10 `CLIPS::client::viewController::authentication::RegistrationActivity`

- **Descrizione:** classe controller che si occupa di interagire con `RegistrationView`
- **Utilizzo:** si occupa di gestire le interazioni dell'utente con `RegistrationView`
- **Metodi:**

`checkFields()` : `bool`

si occupa di controllare che i campi inseriti siano corretti

`clearFields()` : `void`

questo metodo si occupa di resettare i campi dati presenti nella relativa view

`onCreate(SavedBundleInstance)` : `void`

si occupa di inizializzare l'activity con la relativa view

Argomenti:

* `SavedBundleInstance` : `Bundle` contiene i dati dell'istanza creata o di un'attività precedente

`setButtons()` : `void`

si occupa di inizializzare i collegamenti dei bottoni presenti nella view

- **Relazioni con altre classi:**

`CLIPS::client::viewController::authentication::RegistrationView`: classe che si occupa della visualizzazione della schermata per la registrazione

3.8.2.11 `CLIPS::client::viewController::authentication::RegistrationView`

- **Descrizione:** classe che si occupa della visualizzazione della schermata per la registrazione
- **Utilizzo:** consente all'utente di inserire i propri dati per effettuare la registrazione

- **Attributi:**

`confirmButton`: `Button` indica il bottone di conferma di registrazione

`confirmPasswordErrorLabel`: `TextView` indica il messaggio di errore riguardante l'inserimento della conferma della password

`emailErrorLabel`: `TextView` indica il messaggio di errore riguardante l'inserimento dell'email

`emailTextField`: `EditText` rappresenta l'email da inserire da parte dell'utente

`passwordConfirmationTextField`: `EditText` rappresenta la conferma della password da inserire da parte dell'utente

`passwordLabel`: `TextView` indica il messaggio di errore riguardante l'inserimento della password

`passwordTextField`: `EditText` indica la password da inserire da parte dell'utente

`usernameErrorLabel`: `TextView` indica il messaggio di errore riguardante l'inserimento dell'username

`usernameTextField`: `EditText` rappresenta l'username da inserire da parte dell'utente

3.8.2.12 `CLIPS::client::viewController::authentication::UpdateUserInfoActivity`

- **Descrizione:** classe controller che si occupa di interagire con `UpdateUserInfoView`
- **Utilizzo:** si occupa di gestire le interazioni dell'utente con `UpdateUserInfoView`
- **Metodi:**

`onCreate()` : `void`
 si occupa di inizializzare l'activity con la relativa view

`setButtons()` : `void`
 si occupa di inizializzare i collegamenti dei bottoni presenti nella view

- **Relazioni con altre classi:**

`CLIPS::client::viewController::authentication::UpdateUserInfoView`: classe che si occupa della visualizzazione della schermata per il cambio delle credenziali

3.8.2.13 `CLIPS::client::viewController::authentication::UpdateUserInfoView`

- **Descrizione:** classe che si occupa della visualizzazione della schermata per il cambio delle credenziali
- **Utilizzo:** consente all'utente di inserire i nuovi dati per cambiare le sue credenziali
- **Attributi:**

`confirmPasswordErrorLabel`: `TextView` indica il messaggio di errore riguardante l'inserimento della conferma della password

`passwordConfirmationTextField`: `EditText` rappresenta la conferma della password da inserire da parte dell'utente

`passwordErrorLabel`: `TextView` indica il messaggio di errore riguardante l'inserimento della password

passwordTextField: EditText rappresenta la password da inserire da parte dell'utente

usernameErrorLabel: TextView indica il messaggio di errore riguardante l'inserimento dell'username

usernameTextField: EditText rappresenta l'username da inserire da parte dell'utente

3.9 Componente CLIPS::client::viewController::building

3.9.1 Informazioni sul package

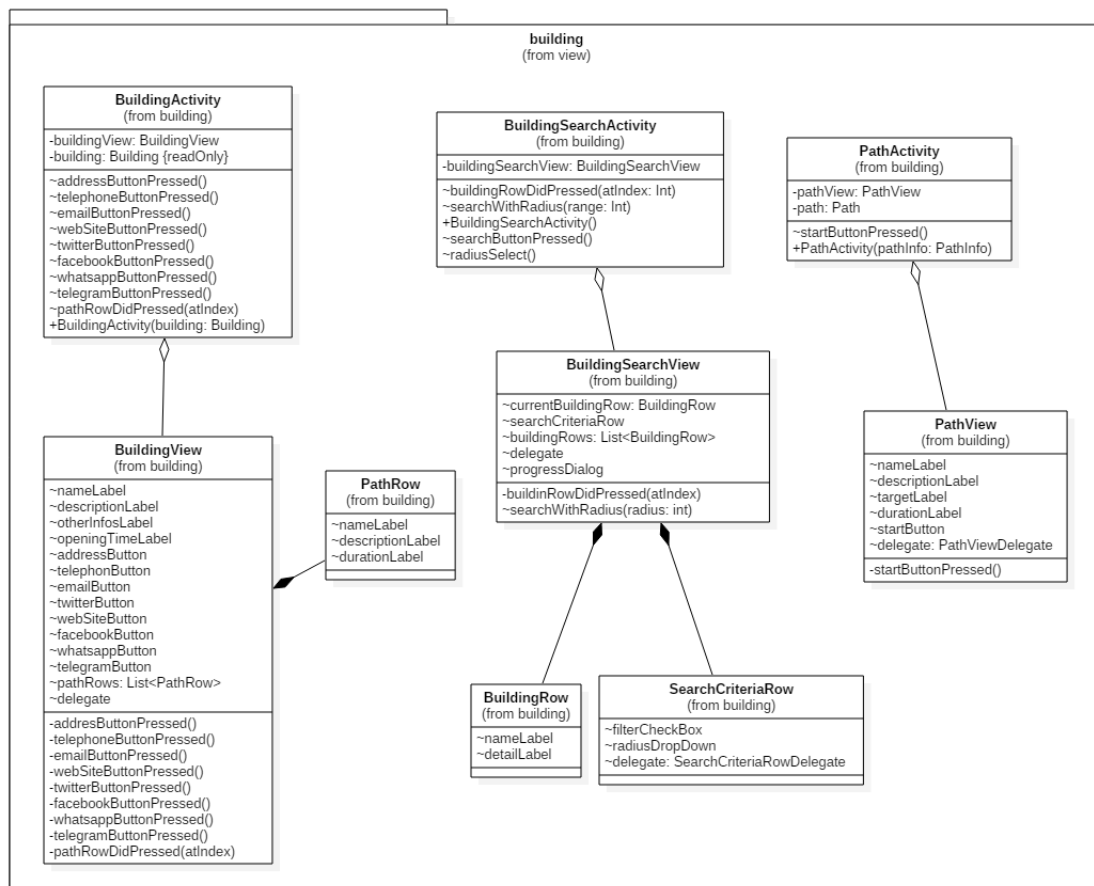


Figura 18: Schema package client::viewController::building

- **Descrizione:** componente che gestisce le informazioni e le interazioni dell'utente con gli edifici abilitati
- **Padre:** viewController

3.9.2 Classi

3.9.2.1 CLIPS::client::viewController::building::BuildingActivity

- **Descrizione:** classe controller che si occupa di interagire con BuildingView
- **Utilizzo:** classe controller che si occupa di interagire con BuildingView
- **Metodi:**
 - `onCreate(savedInstanceState) : void`
si occupa di inizializzare l'activity con la relativa view
 - Argomenti:*
 - * `savedInstanceState : Bundle` contiene i dati dell'istanza creata o di un'attività precedente
 - `setButtons() : void`
si occupa di inizializzare i collegamenti dei bottoni presenti nella view
 - `setItems() : void`
questo metodo rende cliccabile la lista dei risultati e rende visualizzabili i dettagli del percorso selezionato
- **Relazioni con altre classi:**
 - CLIPS::client::viewController::building::BuildingView: classe per la visualizzazione di un edificio

3.9.2.2 CLIPS::client::viewController::building::BuildingRow

- **Descrizione:** classe per la rappresentazione di un edificio nella ricerca
- **Utilizzo:** permette di visualizzare le informazioni di un edificio nella lista dei risultati di ricerca
- **Attributi:**
 - `detaillabel: TextView` rappresenta informazioni aggiuntive riguardanti l'edificio
 - `nameLabel: TextView` rappresenta il nome dell'edificio

3.9.2.3 CLIPS::client::viewController::building::BuildingSearchActivity

- **Descrizione:** classe controller che si occupa di interagire con BuildingSearchView
- **Utilizzo:** si occupa di gestire le interazioni dell'utente con BuildingSearchView
- **Metodi:**
 - `onCreate(savedInstanceState) : void`
si occupa di inizializzare l'activity con la relativa view
 - Argomenti:*
 - * `savedInstanceState : Bundle` contiene i dati dell'istanza creata o di un'attività precedente

`setButtons() : void`

consente l'interazione con i bottoni presenti nella view

`setCheckBoxSignal() : void`

permette l'interazione per la selezione di ricerca per raggio

`setSeekBarSignal() : void`

permette di interagire con lo slider per impostare il raggio di ricerca

- **Relazioni con altre classi:**

`CLIPS::client::viewController::building::BuildingSearchView`: classe per la ricerca degli edifici

3.9.2.4 `CLIPS::client::viewController::building::BuildingSearchView`

- **Descrizione:** classe per la ricerca degli edifici

- **Utilizzo:** permette all'utente di cercare gli edifici vicini per visualizzare le informazioni e/o i percorsi disponibili

- **Attributi:**

`buildingRows`: `ListView` indica la lista di edifici risultanti dalla ricerca

`searchCriteria`: `SearchCriteriaRow` indica i criteri di ricerca

- **Relazioni con altre classi:**

`CLIPS::client::viewController::building::BuildingRow`: classe per la rappresentazione di un edificio nella ricerca

`CLIPS::client::viewController::building::SearchCriteriaRow`: classe utilizzata per determinare i criteri di ricerca degli edifici

3.9.2.5 `CLIPS::client::viewController::building::BuildingView`

- **Descrizione:** classe per la visualizzazione di un edificio

- **Utilizzo:** permette all'utente di visualizzare le informazioni relative all'edificio selezionato

- **Attributi:**

`addressButton`: `button` rappresenta l'indirizzo dell'edificio

`descriptionLabel`: `TextView` indica la descrizione dell'edificio

`emailButton`: `button` indica l'indirizzo email dell'edificio

`facebookButton`: `button` indica l'indirizzo facebook dell'edificio

`nameLabel`: `TextView` rappresenta il nome dell'edificio

`openingTimeLabel`: `TextView` rappresenta gli orari di apertura dell'edificio

`otherInfosLabel`: `TextView` rappresenta le altre informazioni sull'edificio

`pathRows`: `List<TextView>` rappresenta la lista dei percorsi dell'edificio

`telegramButton`: `button` rappresenta il contatto Telegram dell'edificio

`telephoneButton`: `button` rappresenta il numero telefonico dell'edificio

`twitterButton`: `button` indica l'indirizzo twitter dell'edificio

`webSiteButton`: `button` rappresenta l'indirizzo web dell'edificio

`whatsappButton`: `button` indica il contatto WhatsApp dell'edificio

- **Metodi:**

`addressButtonPressed()` : `void`

notifica al controller l'interazione con `addressButton`

`emailButtonPressed()` : `void`

notifica al controller l'interazione con `emailButton`

`facebookButtonPressed()` : `void`

notifica al controller l'interazione con `facebookButton`

`pathRowDidPressed()` : `void`

notifica al controller l'interazione con `pathRow`

`telegramButtonPressed()` : `void`

notifica al controller l'interazione con `telegramButton`

`telephoneButtonPressed()` : `void`

notifica al controller l'interazione con `telephoneButton`

`twitterButtonPressed()` : `void`

notifica al controller l'interazione con `twitterButton`

`websiteButtonPressed()` : `void`

notifica al controller l'interazione con `webSiteButton`

`whatstappButtonPressed()` : `void`

notifica al controller l'interazione con `whatsappButton`

- **Relazioni con altre classi:**

`CLIPS::client::viewController::building::PathRow`: classe per la rappresentazione di un percorso nella ricerca

3.9.2.6 `CLIPS::client::viewController::building::PathActivity`

- **Descrizione:** classe controller che si occupa di interagire con `PathView`

- **Utilizzo:** si occupa di gestire le interazioni dell'utente con `PathView`

- **Metodi:**

`onCreate(savedInstanceState)` : `void`

si occupa di inizializzare l'activity con la relativa view

Argomenti:

* `savedInstanceState` : `Bundle` contiene i dati dell'istanza creata o di un'attività precedente

- **Relazioni con altre classi:**

`CLIPS::client::viewController::building::PathView`: classe che si occupa della visualizzazione della schermata riguardante un percorso

3.9.2.7 CLIPS::client::viewController::building::PathRow

- **Descrizione:** classe per la rappresentazione di un percorso nella ricerca
- **Utilizzo:** permette un percorso disponibile nella lista dei risultati di ricerca
- **Attributi:**
 - pathName: TextView contiene il nome del percorso

3.9.2.8 CLIPS::client::viewController::building::PathView

- **Descrizione:** classe che si occupa della visualizzazione della schermata riguardante un percorso
- **Utilizzo:** consente all'utente di visualizzare le informazioni riguardanti un percorso e se l'utente si trova nell'edificio del percorso consente di iniziarlo
- **Attributi:**
 - descriptionLabel: TextView rappresenta la descrizione del percorso
 - durationLabel: TextView rappresenta la durata stimata del percorso
 - nameLabel: TextView indica il nome del percorso
 - startButton: Button indica il bottone per iniziare il percorso
 - targetLabel: TextView indica a chi è consigliato questo percorso

3.9.2.9 CLIPS::client::viewController::building::SearchCriteriaRow

- **Descrizione:** classe utilizzata per determinare i criteri di ricerca degli edifici
- **Utilizzo:** permette all'utente di visualizzare i criteri di ricerca disponibili
- **Attributi:**
 - distanceLabel: TextView rappresenta la distanza selezionata nella ricerca per raggio
 - filterCheckBox: CheckBox viene usata per abilitare/disabilitare la ricerca per raggio
 - kmFilter: SeekBar indica i km da selezionare per la ricerca per raggio
 - searchBuilding: Button viene utilizzato per iniziare la ricerca degli edifici

3.10 Componente CLIPS::client::viewController::games**3.10.1 Informazioni sul package**

- **Descrizione:** componente che gestisce le prove che l'utente deve completare all'interno di un percorso
- **Padre:** viewController

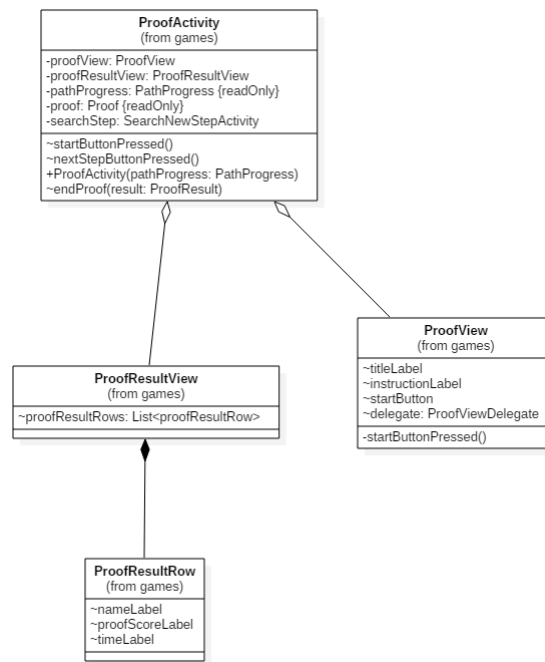


Figura 19: Prima parte schema package client::viewController::games

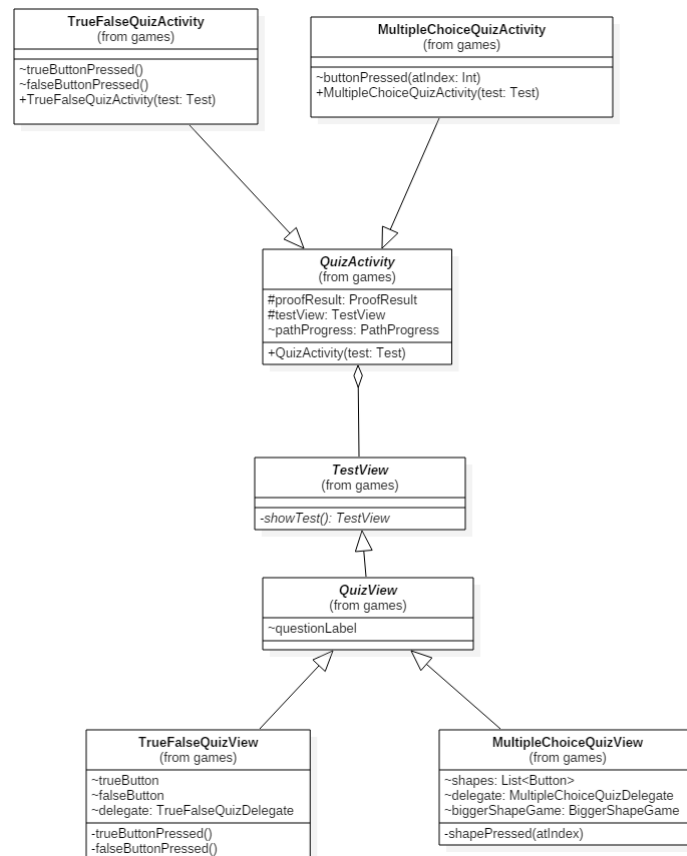


Figura 20: Seconda parte schema package client::viewController::games

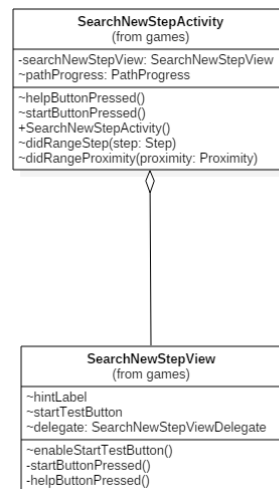


Figura 21: Terza parte schema package client::viewController::games

3.10.2 Classi

3.10.2.1 CLIPS::client::viewController::games::MultipleChoiceQuizActivity

- **Descrizione:** classe controller che si occupa di interagire con MultipleChoiceView
- **Utilizzo:** si occupa di gestire le interazioni dell'utente con MultipleChoiceView
- **Metodi:**
 - onCreate() : void
si occupa di inizializzare l'activity con la relativa view
- **Relazioni con altre classi:**
 - CLIPS::client::viewController::games::QuizActivity: interfaccia per segnalare gli eventi della classe QuizView

3.10.2.2 CLIPS::client::viewController::games::MultipleChoiceQuizView

- **Descrizione:** classe per il quiz a risposta multipla
- **Utilizzo:** si occupa di fornire un'interfaccia per il quiz a risposta multipla
- **Attributi:**
 - choice1: Button prima risposta alla domanda
 - choice2: Button seconda possibile risposta alla domanda
 - choice3: Button terza possibile risposta alla domanda
 - choice4: Button quarta possibile risposta alla domanda
- **Relazioni con altre classi:**
 - CLIPS::client::viewController::games::QuizView: classe base per i quiz

3.10.2.3 CLIPS::client::viewController::games::ProofActivity

- **Descrizione:** classe controller che si occupa di interagire con ProofView
- **Utilizzo:** si occupa di gestire le interazioni dell'utente con ProofView
- **Metodi:**
 - `onCreate() : void`
si occupa di inizializzare l'activity con la relativa view
- **Relazioni con altre classi:**
 - CLIPS::client::viewController::games::ProofResultView: classe che rappresenta la schermata dei risultati

3.10.2.4 CLIPS::client::viewController::games::ProofResultActivity

- **Descrizione:** classe che permette di visualizzare il risultato di una prova
- **Utilizzo:** l'utente può vedere il risultato della prova che ha appena svolto
- **Metodi:**
 - `onCreate() : void`
si occupa di inizializzare l'activity con la relativa view

3.10.2.5 CLIPS::client::viewController::games::ProofResultRow

- **Descrizione:** classe che rappresenta una risultato rappresentato all'interno di una riga che può essere cliccata
- **Utilizzo:** permette all'utente di visualizzare le informazioni generali di un risultato di una prova e di cliccarci per visualizzarne le informazioni dettagliate

3.10.2.6 CLIPS::client::viewController::games::ProofResultView

- **Descrizione:** classe che rappresenta la schermata dei risultati
- **Utilizzo:** permette all'utente di visualizzare le informazioni generali dei risultati delle prove e cliccare sulle prove delle quali vuole visualizzare le informazioni dettagliate
- **Attributi:**
 - `result: TextView` contiene il risultato della prova appena svolta
- **Relazioni con altre classi:**
 - CLIPS::client::viewController::games::ProofResultRow: classe che rappresenta una risultato rappresentato all'interno di una riga che può essere cliccata

3.10.2.7 CLIPS::client::viewController::games::ProofView

- **Descrizione:** classe che rappresenta la schermata della prova
- **Utilizzo:** permette all'utente di visualizzare la prova da giocare
- **Attributi:**
 - `instructionLabel:` `TextView` contiene le istruzioni relative alla prova
 - `startButton:` `Button` bottone che una volta premuto permette di iniziare la prova

3.10.2.8 CLIPS::client::viewController::games::QuizActivity

- **Descrizione:** interfaccia per segnalare gli eventi della classe `QuizView`
- **Utilizzo:** si occupa di fornire i metodi necessari alla classe `QuizView` per notificare gli eventi

3.10.2.9 CLIPS::client::viewController::games::QuizResultView

- **Descrizione:** classe per la visualizzazione del risultato di un quiz
- **Utilizzo:** fornisce all'utente un'interfaccia affinché visualizzi il risultato del quiz
- **Attributi:**
 - `continueButton:` `Button` button per chiudere la schermata e continuare il percorso
 - `feedbackLabel:` `string` mostra la frase di successo/fallimento del quiz
- **Metodi:**
 - `continueButtonPressed() : void`
notifica il controller che il button per continuare è stato premuto
 - `showFailureResult(correctAnswer) : void`
mostra la risposta corretta se il quiz è stato fallito
 - Argomenti:*
 - * `correctAnswer : string` indica la risposta corretta
 - `showSuccessfulResult(score) : void`
mostra il risultato ottenuto se il quiz è stato superato
 - Argomenti:*
 - * `score : int` indica il punteggio ottenuto

3.10.2.10 CLIPS::client::viewController::games::QuizView

- **Descrizione:** classe base per i quiz
- **Utilizzo:** fornisce una base per i vari tipi di test da istanziare

- **Attributi:**

`questionLabel: string` rappresenta la domanda da porre nel quiz

- **Relazioni con altre classi:**

`CLIPS::client::viewController::games::TestView`: classe che fornisce una base dalla quale è possibile creare vari tipi di giochi

3.10.2.11 `CLIPS::client::viewController::games::SearchNewStepActivity`

- **Descrizione:** classe controller che si occupa di interagire con `SearchNewStepView`

- **Utilizzo:** si occupa di gestire le interazioni dell'utente con `SearchNewStepView`

- **Metodi:**

`onCreate(savedInstanceState) : void`

si occupa di inizializzare l'activity con la relativa view

Argomenti:

* `savedInstanceState : Bundle` stato dell'istanza

- **Relazioni con altre classi:**

`CLIPS::client::viewController::games::SearchNewStepView`: classe che rappresenta la schermata per la ricerca della prossima prova del percorso

3.10.2.12 `CLIPS::client::viewController::games::SearchNewStepView`

- **Descrizione:** classe che rappresenta la schermata per la ricerca della prossima prova del percorso

- **Utilizzo:** permette all'utente di cercare in modo semplificato la ricerca della prossima prova del percorso

- **Attributi:**

`descriptionlabel: TextView` invita ad andare verso la prossima prova

`hintlabel: TextView` contiene il suggerimento per cercare la prossima prova

3.10.2.13 `CLIPS::client::viewController::games::TestResultView`

- **Descrizione:** classe che fornisce una base per la visualizzazione del risultato della prova

- **Utilizzo:** permette all'utente di visualizzare il risultato della prova

- **Metodi:**

`showResult() : void`

restituisce la view con il risultato

3.10.2.14 CLIPS::client::viewController::games::TestView

- **Descrizione:** classe che fornisce una base dalla quale è possibile creare vari tipi di giochi
- **Utilizzo:** viene utilizzata per visualizzare un'interfaccia di gioco all'utente
- **Metodi:**
 - `showTest() : TestView`
restituisce l'interfaccia grafica del test

3.10.2.15 CLIPS::client::viewController::games::TrueFalseQuizActivity

- **Descrizione:** classe controller che si occupa di interagire con TrueFalseQuizView
- **Utilizzo:** si occupa di gestire le interazioni dell'utente con TrueFalseQuizView
- **Metodi:**
 - `onCreate(savedInstanceState) : void`
si occupa di inizializzare l'activity con la relativa view
 - Argomenti:*
 - * `savedInstanceState : Bundle` stato dell'istanza
- **Relazioni con altre classi:**
 - `CLIPS::client::viewController::games::QuizActivity:` interfaccia per segnalare gli eventi della classe QuizView

3.10.2.16 CLIPS::client::viewController::games::TrueFalseQuizView

- **Descrizione:** classe per il quiz vero/falso
- **Utilizzo:** si occupa di fornire un'interfaccia per la prova di tipo vero/falso
- **Attributi:**
 - `falseButton: Button` button grafico per rispondere falso al quiz
 - `trueButton: Button` button grafico per rispondere vero al quiz
- **Metodi:**
 - `falseButtonPressed() : void`
questo metodo si occupa di notificare al controller che è stato premuto falseButton
 - `trueButtonPressed() : void`
questo metodo si occupa di notificare al controller che è stato premuto trueButton
- **Relazioni con altre classi:**
 - `CLIPS::client::viewController::games::QuizView:` classe base per i quiz

3.11 Componente CLIPS::client::viewController::savedresults

3.11.1 Informazioni sul package

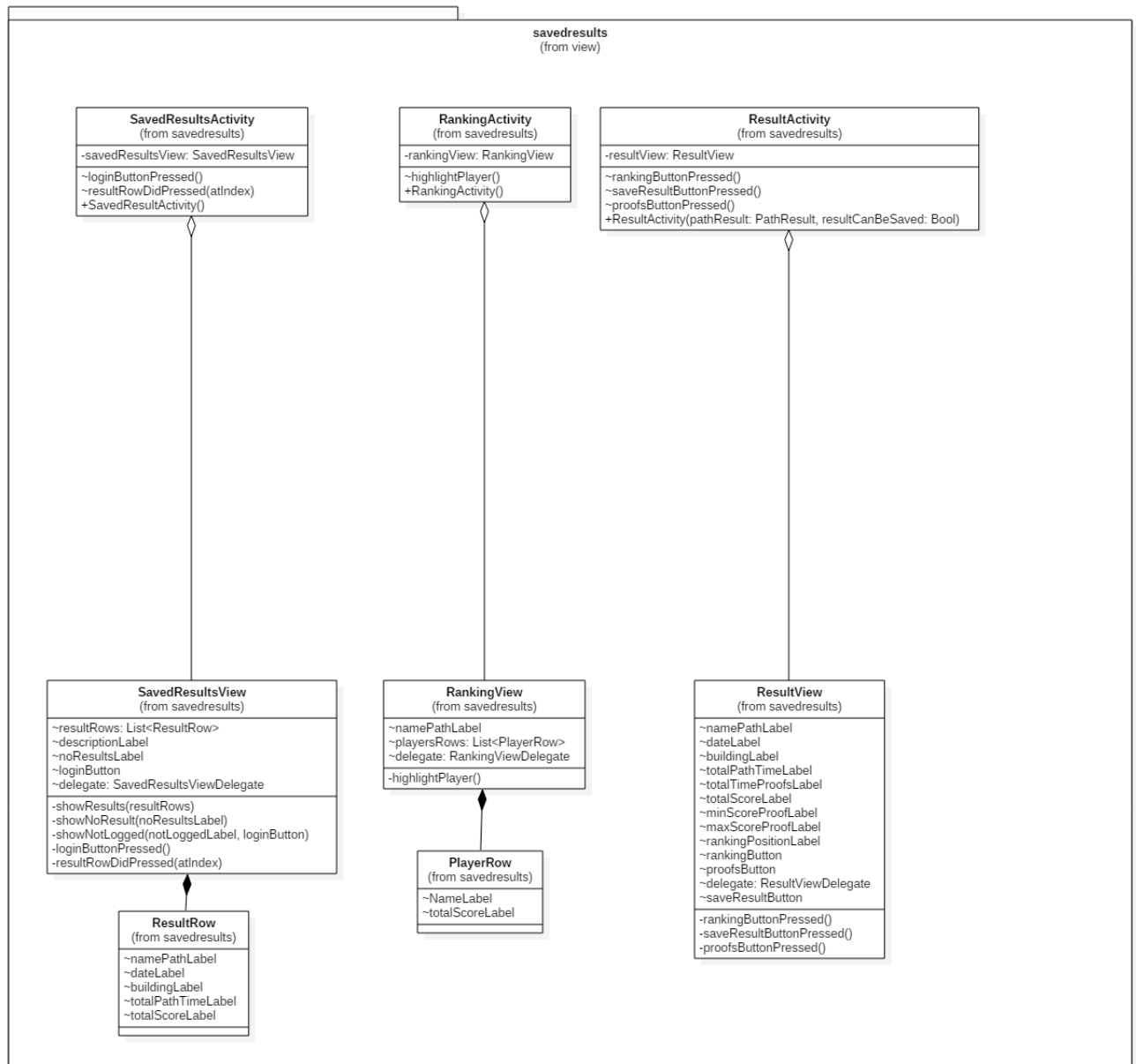


Figura 22: Schema package client::viewController::savedresults

- **Descrizione:** componente che raggruppa le le view e i controller dei risultati salvati e delle classifiche
- **Padre:** viewController

3.11.2 Classi

3.11.2.1 CLIPS::client::viewController::savedresults::PlayerRow

- **Descrizione:** classe che rappresenta una riga all'interno di una classifica
- **Utilizzo:** permette all'utente di visualizzare il nome del giocatore e il suo punteggio nel percorso d'interesse
- **Attributi:**
 - `nameLabel: TextView` indica il nome del giocatore
 - `totalScoreLabel: TextView` indica il punteggio conseguito dal giocatore

3.11.2.2 CLIPS::client::viewController::savedresults::RankingActivity

- **Descrizione:** classe controller che si occupa di interagire con RankingView
- **Utilizzo:** si occupa di gestire le interazioni dell'utente con RankingView
- **Metodi:**
 - `onCreate(savedInstanceState) : void`
si occupa di inizializzare l'activity con la relativa view
 - Argomenti:*
 - * `savedInstanceState : Bundle` contiene i dati dell'istanza creata o di un'attività precedente

3.11.2.3 CLIPS::client::viewController::savedresults::RankingView

- **Descrizione:** classe per rappresentare la classifica di un percorso
- **Utilizzo:** permette all'utente di visualizzare la classifica del percorso
- **Attributi:**
 - `namePathLabel: TextView` indica il nome del percorso
 - `playerRows: ListView` rappresenta la lista dei giocatori in classifica
- **Relazioni con altre classi:**
 - `CLIPS::client::viewController::savedresults::RankingActivity:` classe controller che si occupa di interagire con RankingView

3.11.2.4 CLIPS::client::viewController::savedresults::ResultActivity

- **Descrizione:** classe controller che si occupa di interagire con ResultView
- **Utilizzo:** si occupa di gestire le interazioni dell'utente con ResultView
- **Metodi:**

`onCreate()` : void
 si occupa di inizializzare l'activity con la relativa view

- **Relazioni con altre classi:**

`CLIPS::client::viewController::savedresults::ResultView`: classe che rappresenta la schermata nel quali si possono visualizzare i risultati di un percorso

`CLIPS::client::viewController::savedresults::ResultView`: classe che rappresenta la schermata nel quali si possono visualizzare i risultati di un percorso

3.11.2.5 `CLIPS::client::viewController::savedresults::ResultRow`

- **Descrizione:** classe che rappresenta una riga di un risultato
- **Utilizzo:** permette all'utente di visualizzare le informazioni generali di un risultato e di cliccarci per visualizzare quelle dettagliate
- **Attributi:**

`buildingLabel`: `TextView` rappresenta l'edificio dove si è svolto il percorso

`dateLabel`: `TextView` indica la data del percorso

`namePathLabel`: `TextView` indica il nome del percorso

`totalPathTimeLabel`: `TextView` indica il tempo totale speso per completare percorso

`totalScoreLabel`: `TextView` rappresenta il punteggio totale ottenuto svolgendo il percorso

3.11.2.6 `CLIPS::client::viewController::savedresults::ResultView`

- **Descrizione:** classe che rappresenta la schermata nel quali si possono visualizzare i risultati di un percorso
- **Utilizzo:** permette all'utente di visualizzare le informazioni dettagliate del risultato di un percorso
- **Attributi:**

`buildingLabel`: `TextView` rappresenta il nome dell'edificio dove si è svolto il percorso

`dateLabel`: `TextView` indica la data in cui si è svolto il percorso

`maxScoreProofLabel`: `TextView` rappresenta il punteggio massimo conseguito in una prova

`minScoreProofLabel`: `TextView` rappresenta il punteggio minimo conseguito in una prova

`namePathLabel`: `TextView` indica il nome del percorso

`proofsButton`: `Button` rappresenta il button delle prove

`rankingButton`: `Button` rappresenta il button per la classifica

`rankingPositionLabel`: `TextView` indica la posizione in classifica

`saveResultButton`: `Button` indica il bottone per salvare i risultati del percorso
`totalPathTimeLabel`: `TextView` indica il tempo totale speso per finire il percorso
`totalScoreLabel`: `TextView` indica il punteggio totale conseguito
`totalTimeProofsLabel`: `TextView` rappresenta il tempo totale impiegato per le prove del percorso

3.11.2.7 CLIPS::client::viewController::savedresults::SavedResultsActivity

- **Descrizione:** classe controller che si occupa di interagire con `SavedResultView`
- **Utilizzo:** si occupa di gestire le interazioni dell'utente con `SavedResultView`
- **Metodi:**
 - `isLoggedIn()` : `bool`
controlla se l'utente è loggiato per poter visualizzare i suoi risultati
 - `onCreate()` : `void`
si occupa di inizializzare l'activity con la relativa view
 - `setButtons()` : `void`
si occupa di impostare i collegamenti dei bottoni presenti nella view
- **Relazioni con altre classi:**
 - `CLIPS::client::viewController::savedresults::SavedResultView`: classe che rappresenta la schermata dei risultati salvati di un utente

3.11.2.8 CLIPS::client::viewController::savedresults::SavedResultView

- **Descrizione:** classe che rappresenta la schermata dei risultati salvati di un utente
- **Utilizzo:** permette all'utente di visualizzare i propri risultati salvati
- **Attributi:**
 - `descriptionLabel`: `TextView` rappresenta la descrizione del percorso
 - `loginButton`: `Button` rappresenta il button per effettuare il login
 - `noResultsLabel`: `TextView` indica se non è presente nessun risultato
 - `resultRows`: `ListView` indica la lista di risultati
- **Relazioni con altre classi:**
 - `CLIPS::client::viewController::savedresults::ResultRow`: classe che rappresenta una riga di un risultato

3.12 Componente CLIPS::client::viewController::utility

3.12.1 Informazioni sul package

- **Descrizione:** componente che raggruppa le view generali dell'app
- **Padre:** `viewController`

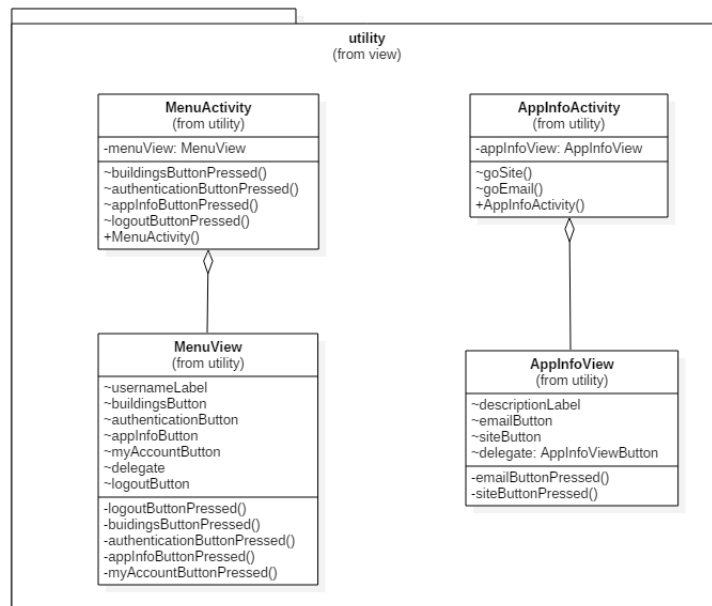


Figura 23: Schema package client::viewController::utility

3.12.2 Classi

3.12.2.1 CLIPS::client::viewController::utility::AppInfoActivity

- **Descrizione:** classe controller che si occupa di interagire con AppInfoView
- **Utilizzo:** si occupa di gestire le interazioni dell'utente con AppInfoView
- **Metodi:**

`onCreate(savedInstanceState) : void`
 si occupa di inizializzare l'activity con la relativa view

Argomenti:

* `savedInstanceState : Bundle` stato dell'istanza

- **Relazioni con altre classi:**

CLIPS::client::viewController::utility::AppInfoView: classe che si occupa delle informazioni generali dell'app

3.12.2.2 CLIPS::client::viewController::utility::AppInfoView

- **Descrizione:** classe che si occupa delle informazioni generali dell'app
- **Utilizzo:** permette all'utente di visualizzare le informazioni generali dell'app
- **Attributi:**

`textContainer: LinearLayout` layout che contiene le TextView necessarie a descrivere le informazioni dell'app

3.12.2.3 CLIPS::client::viewController::utility::MenuActivity

- **Descrizione:** classe controller che si occupa di interagire con MenuView
- **Utilizzo:** si occupa di gestire le interazioni dell'utente con MenuView
- **Metodi:**

```
selectDrawerItem() : void
si occupa di gestire i bottoni presenti nel menu

setContentView(layoutResID) : void
si occupa di gestire gli elementi del layout della classe
```

Argomenti:

```
* layoutResID : int layout da inserire

setContent() : void
si occupa di impostare il contenuto della classe

useDrawerToggle() : bool
ritorna true se si intende utilizzare il drawer toggle, false altrimenti

useToolbar() : bool
ritorna true se si intende utilizzare la toolbar, false altrimenti
```

- **Relazioni con altre classi:**

CLIPS::client::viewController::utility::MenuView: classe che si occupa di far visualizzare il menu

3.12.2.4 CLIPS::client::viewController::utility::MenuView

- **Descrizione:** classe che si occupa di far visualizzare il menu
- **Utilizzo:** consente all'utente di navigare nell'app tramite il menu

3.13 Componente CLIPS::server

3.13.1 Informazioni sul package

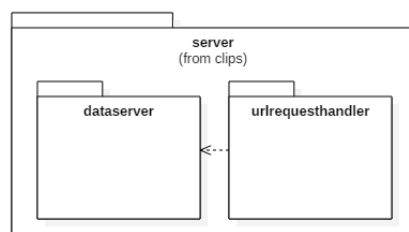


Figura 24: Schema package server

- **Descrizione:** componente globale per il back end del prodotto
- **Padre:** CLIPS
- **Package contenuti:**

- CLIPS::server::dataserver: package per la gestione dei dati sul server
- CLIPS::server::urlrequesthandler: componente che gestisce le richieste inviate al server e le risposte da inviare al client

3.14 Componente CLIPS::server::dataserver

3.14.1 Informazioni sul package

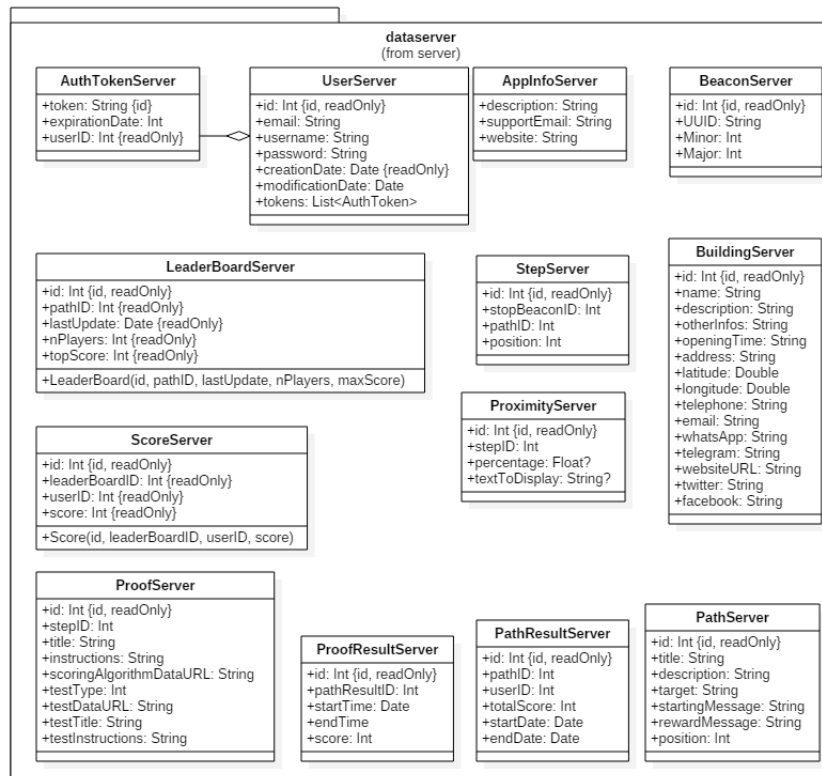


Figura 25: Schema package server::dataserver

- **Descrizione:** package per la gestione dei dati sul server
- **Padre:** server

3.14.2 Classi

3.14.2.1 CLIPS::server::dataserver::AppInfoServer

- **Descrizione:** classe che rappresenta le informazioni dell'app sul server
- **Utilizzo:** permette di salvare sul server le informazioni dell'app

3.14.2.2 CLIPS::server::dataserver::AuthTokenServer

- **Descrizione:** Classe che rappresenta un token che si riferisce ad un utente loggato

- **Utilizzo:** permette utilizzare un token per riferirsi ad utente loggato

3.14.2.3 CLIPS::server::dataserver::BeaconServer

- **Descrizione:** classe che rappresenta un beacon nel server
- **Utilizzo:** permette di salvare sul server un beacon

3.14.2.4 CLIPS::server::dataserver::BuildingServer

- **Descrizione:** classe che rappresenta un edificio sul server
- **Utilizzo:** permette di salvare e modificare i dati di un edificio sul server

3.14.2.5 CLIPS::server::dataserver::LeaderBoardServer

- **Descrizione:** classe che rappresenta la classifica sul server
- **Utilizzo:** permette di salvare i dati della classifica sul server

3.14.2.6 CLIPS::server::dataserver::PathResultServer

- **Descrizione:** classe che rappresenta il risultato di un percorso sul server
- **Utilizzo:** consente di salvare i risultati di un percorso sul server

3.14.2.7 CLIPS::server::dataserver::PathServer

- **Descrizione:** classe che rappresenta un percorso sul server
- **Utilizzo:** permette di creare, modificare ed eliminare un percorso sul server

3.14.2.8 CLIPS::server::dataserver::ProofResultServer

- **Descrizione:** classe che rappresenta i risultati di una prova sul server
- **Utilizzo:** consente di salvare il risultato di una prova sul server

3.14.2.9 CLIPS::server::dataserver::ProofServer

- **Descrizione:** classe che rappresenta una prova sul server
- **Utilizzo:** permette di creare, modificare ed eliminare una prova sul server

3.14.2.10 CLIPS::server::dataserver::ProximityServer

- **Descrizione:** classe che rappresenta sul server un beacon indicato alla segnalazione della distanza dalla prossima prova
- **Utilizzo:** permette di salvare i beacon di segnalazione sul server

3.14.2.11 CLIPS::server::dataserver::ScoreManager

- **Descrizione:** classe che rappresenta un risultato sul server
- **Utilizzo:** permette di salvare un risultato nel server

3.14.2.12 CLIPS::server::dataserver::StepServer

- **Descrizione:** classe che rappresenta uno step di un percorso nel server
- **Utilizzo:** permette di salvare sul server uno step di un percorso

3.14.2.13 CLIPS::server::dataserver::UserServer

- **Descrizione:** classe che rappresenta un utente nel server
- **Utilizzo:** permette di salvare sul server un utente
- **Relazioni con altre classi:**

CLIPS::server::dataserver::AuthTokenServer: Classe che rappresenta un token che si riferisce ad un utente loggato

3.15 Componente CLIPS::server::urlrequesthandler**3.15.1 Informazioni sul package**

- **Descrizione:** componente che gestisce le richieste inviate al server e le risposte da inviare al client
- **Padre:** server

3.15.2 Classi**3.15.2.1 CLIPS::server::urlrequesthandler::AppInfoProvider**

- **Descrizione:** classe che gestisce la richiesta di informazioni sull'app (come la descrizione dell'app, l'indirizzo del sito di supporto, l'email del supporto tecnico ed altro)
- **Utilizzo:** si occupa di restituire le informazioni dell'app richieste

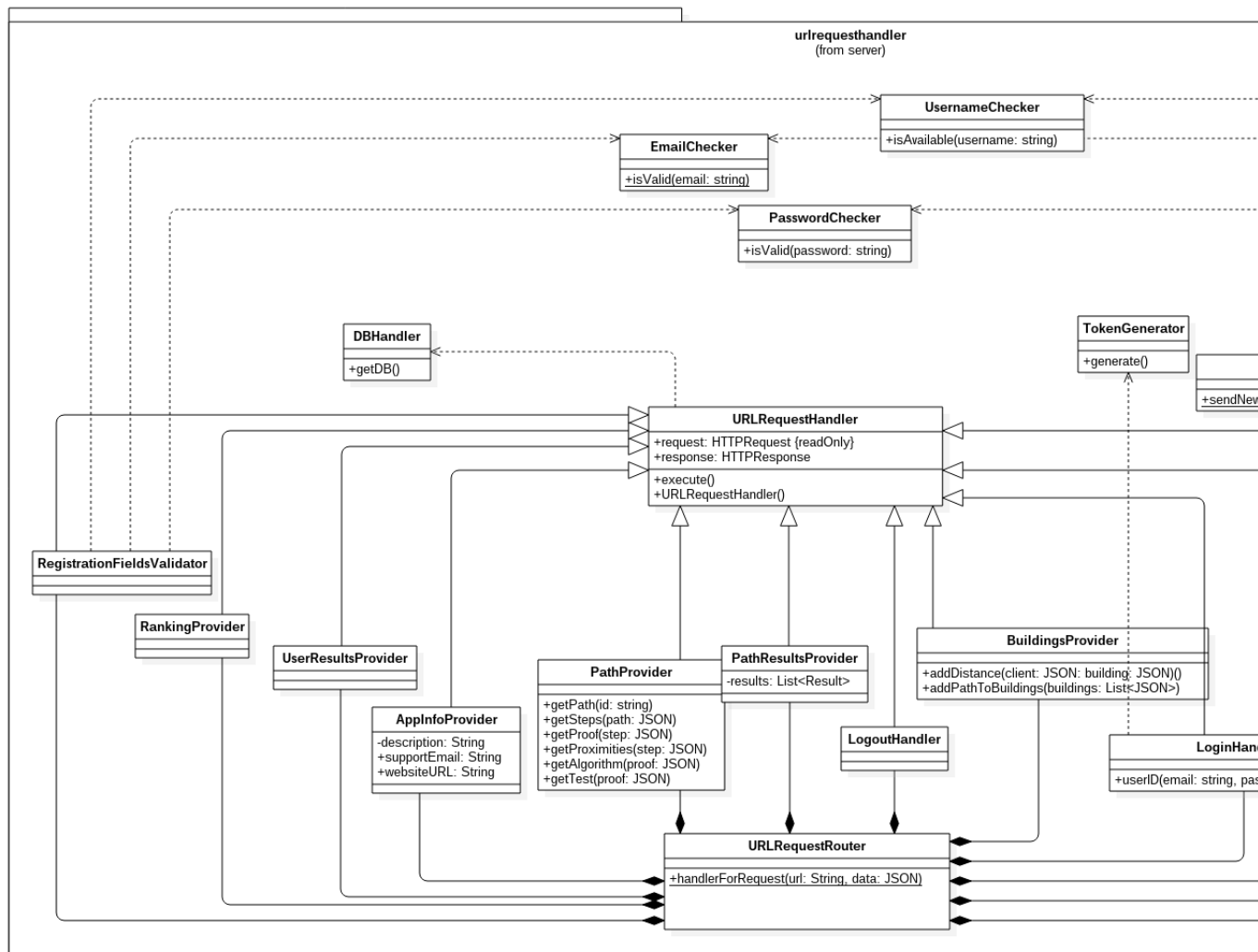


Figura 26: Schema package server::urlrequesthandler

3.15.2.2 CLIPS::server::urlrequesthandler::BuildingsProvider

- **Descrizione:** classe che gestisce la richiesta degli edifici dal client al server
- **Utilizzo:** si occupa di restituire le informazioni sugli edifici richieste
- **Metodi:**

`addDistance() : void`

aggiunge ad un edificio la distanza dal client

`addPathToBuildings(buildings) : Promise`

ritorna una Promise che aggiunge le descrizioni dei percorsi agli edifici in input

Argomenti:

- * **buildings** : JSON array di oggetti JSON contenenti gli edifici cui si vogliono aggiungere le descrizioni dei percorsi: in particolare ciascun oggetto JSON deve contenere almeno la chiave id con l'id dell'edificio

- **Relazioni con altre classi:**

`CLIPS::server::urlrequesthandler::DBHandler`: classe che si occupa di gestire il DB e di ritornare un oggetto configurato della libreria knex

3.15.2.3 `CLIPS::server::urlrequesthandler::DBHandler`

- **Descrizione:** classe che si occupa di gestire il DB e di ritornare un oggetto configurato della libreria knex

- **Utilizzo:** creando un oggetto di questa classe è possibile creare query per interagire con il database

- **Metodi:**

`getDB()` : `JSON`

ritorna un oggetto database di knex che si può interrogare per ottenere o inserire dati nel database

3.15.2.4 `CLIPS::server::urlrequesthandler::EmailChecker`

- **Descrizione:** fornisce i metodi per la validazione di un indirizzo email

- **Utilizzo:** attraverso il metodo `isValid(string)` è possibile controllare se l'indirizzo email passato in input è valido

- **Metodi:**

`isValid(email)` : `bool`

ritorna vero se l'indirizzo mail è valido

Argomenti:

* `email` : `string` email che si intende validare verificare

3.15.2.5 `CLIPS::server::urlrequesthandler::EmailSender`

- **Descrizione:** classe usata dal server per inviare un email

- **Utilizzo:** si occupa di inviare un'email da `beaconstrips.swe@gmail.com`

- **Metodi:**

`sendMail()` : `Promise`

invia la mail dall'indirizzo `beaconstrips.swe@gmail.com`

3.15.2.6 `CLIPS::server::urlrequesthandler::GetUserData`

- **Descrizione:** classe del server che gestisce la richiesta dei dati utente

- **Utilizzo:** si occupa di tornare al client i dati sull'utente che ha effettuato la chiamata

- **Metodi:**

`userData(userID)` : `Promise`

metodo che a partire dall'`userID` restituisce i dati utente (email e username)

Argomenti:

* `userID` : `string` id dell'utente

3.15.2.7 `CLIPS::server::urlrequesthandler::LoginHandler`

- **Descrizione:** classe che gestisce le richieste di login da parte del client
- **Utilizzo:** si occupa di gestire il login lato server
- **Metodi:**

`userID(password)` : `Promise`
ritorna una `Promise` che ottiene in maniera asincrona l'id dell'utente se la coppia email/password è valida, un messaggio di errore altrimenti

Argomenti:

* `password` : `string` password dell'utente che intende autenticarsi

- **Relazioni con altre classi:**

`CLIPS::server::urlrequesthandler::DBHandler`: classe che si occupa di gestire il DB e di ritornare un oggetto configurato della libreria knex

3.15.2.8 `CLIPS::server::urlrequesthandler::LogoutHandler`

- **Descrizione:** classe che gestisce la richiesta di logout da parte del client (eliminando il token associato al client)
- **Utilizzo:** si occupa di effettuare il logout lato server

3.15.2.9 `CLIPS::server::urlrequesthandler::PasswordChecker`

- **Descrizione:** verifica che la password soddisfi i requisiti minimi di sicurezza
- **Utilizzo:** utilizzare il metodo `isValid(string)` per verificare la sicurezza sufficiente della password inserita
- **Attributi:**

`instructions`: `string` istruzioni da far visualizzare all'utente per fornire una password valida

- **Metodi:**

`isValid(password)` : `bool`
ritorna vero se la password è valida

Argomenti:

* `password` : `string` la password che si intende verificare

3.15.2.10 `CLIPS::server::urlrequesthandler::PasswordResetEmailSender`

- **Descrizione:** classe che invia una email contenente i dettagli per il reset di una password dimenticata

- **Utilizzo:** si occupa di notificare l'utente della nuova password impostata dopo averne chiesto il ripristino

- **Metodi:**

`sendNewPassword(email, password) : Promise`

invia una mail contenente la nuova password all'indirizzo email indicato

Argomenti:

- * `email` : `string` l'email dell'utente cui comunicare la nuova password
- * `password` : `string` la nuova password da comunicare all'utente

3.15.2.11 CLIPS::server::urlrequesthandler::PasswordResetHandler

- **Descrizione:** classe del server per gestire la richiesta di reset di una password dimenticata

- **Utilizzo:** si occupa di verificare se l'indirizzo email fornito corrisponde con quello di qualche account ed eventualmente di resecare la password ed inviare una mail di avvenuto ripristino

- **Metodi:**

`getEmail() : Promise`

ritorna una promise che si risolve con successo solamente se con la richiesta viene fornita una email associata a qualche utente registrato

3.15.2.12 CLIPS::server::urlrequesthandler::PathProvider

- **Descrizione:** classe che gestisce la richiesta del percorso dal client

- **Utilizzo:** si occupa di restituire le informazioni sul percorso richieste

- **Metodi:**

`fulfilledProof(proof) : Promise`

ritorna una Promise in grado di aggiungere in maniera asincrona i dati derivati alla prova in ingresso

Argomenti:

- * `proof` : `JSON` la prova che si vuole completare inserendo dati derivati; proof deve essere un JSON contenente almeno le chiavi `testData` e `scoringAlgorithmData` con l'id del test da integrare e l'id dell'algoritmo per i punteggi

`fulfilledProximity(proximity) : Promise`

ritorna una Promise in grado di aggiungere in maniera asincrona i dati derivati alla Proximity in ingresso

Argomenti:

- * `proximity` : `JSON` la Proximity che si vuole completare inserendo dati derivati; `proximity [U+FFFD]` un JSON che deve contenere almeno la chiave `beaconID` con l'id del beacon che viene usato per la prossimità `[U+FFFD]`

`fulfilledStep(step) : Promise`

ritorna una Promise in grado di aggiungere in maniera asincrona i dati derivati alla tappe in ingresso

Argomenti:

* **step** : JSON la tappa che si vuole completare inserendo dati derivati; in particolare **step** [U+FFFD] un JSON object che deve contenere almeno le chiavi **stopbeaconID** con l'id del beacon della stazione, **proof** con un oggetto di tipo Proof e **proximities** con un array di oggetti Proximity

getAlgorithm(proof) : Promise

ritorna una Promise in grado di tornare l'algoritmo per il calcolo dei punteggi della prova in input

Argomenti:

* **proof** : JSON la prova di cui si vuole ottenere l'algoritmo; [U+FFFD] un oggetto JSON che rappresenta l'oggetto Proof

getBeacon(beaconID) : Promise

metodo che ritorna una Promise in grado di recuperare in maniera asincrona i dati di un Beacon con ID in ingresso

Argomenti:

* **beaconID** : int ID del beacon di cui si vogliono recuperare le informazioni

getPath(pathID) : Promise

ritorna una Promise in grado di recuperare in maniera asincrona i dati del percorso associate all'ID in ingresso

Argomenti:

* **pathID** : int id del percorso di cui si vogliono recuperare i dati

getProof(step) : Promise

metodo che ritorna una Promise in grado di recuperare in maniera asincrona i dati della prova associata alla tappa in ingresso

Argomenti:

* **step** : JSON tappa di cui si vogliono recuperare i dati relativi alla prova, l'oggetto JSON deve contenere almeno la chiave **proofID** con l'id della Proof

getProximities(step) : Promise

ritorna una Promise in grado di recuperare in maniera asincrona i dati delle Proximity associate alla tappa in ingresso

Argomenti:

* **step** : JSON tappa di cui si vogliono avere le proximities; **step** [U+FFFD] un JSON che deve contenere almeno la chiave **id** con l'id della stazione

getSteps(path) : Promise

metodo che ritorna una Promise in grado di recuperare in maniera asincrona i dati delle tappe associate al path in ingresso

Argomenti:

* **path** : JSON path di cui si vogliono avere le tappe, in particolare l'oggetto JSON path deve contenere la chiave **id** con l'id del percorso

getTest(proof) : Promise

ritorna una Promise in grado di ottenere i dati relativi ai giochi nella prova da effettuare

Argomenti:

* **proof** : JSON la prova di cui si vogliono ottenere i dati: in particolare il JSON deve contenere almeno le chiavi **testData** e **algorithmData** con l'id dei test e degli algoritmi

- **Relazioni con altre classi:**

`CLIPS::server::urlrequesthandler::DBHandler`: classe che si occupa di gestire il DB e di ritornare un oggetto configurato della libreria knex

3.15.2.13 `CLIPS::server::urlrequesthandler::PathResultsProvider`

- **Descrizione:** classe che gestisce la richiesta del risultato del percorso dal client
- **Utilizzo:** si occupa di restituire le informazioni del risultato sul percorso richieste

- **Relazioni con altre classi:**

`CLIPS::server::urlrequesthandler::DBHandler`: classe che si occupa di gestire il DB e di ritornare un oggetto configurato della libreria knex

3.15.2.14 `CLIPS::server::urlrequesthandler::PostUserData`

- **Descrizione:** classe del server che si occupa della modifica dei dati utente
- **Utilizzo:** si occupa di modificare nel database i dati dell'utente quando un client chiede che vengano modificati
- **Metodi:**

`getNewData()` : `Promise`
estrapola dalla richiesta i nuovi dati da salvare nel database

`validateData(data)` : `Promise`
ritorna una promise che viene risolta con successo solamente quando tutti i dati in input sono validi

Argomenti:

* `data` : `JSON` un `JSON` con almeno una tra: le chiave `username` con il nuovo username, la chiave `email` con la nuova email, la coppia di chiavi `oldPassword` e `newPassword` con la vecchia e la nuova password.

`validateEmail()` : `Promise`
ritorna una promise che risolve senza errori solamente se la nuova password è valida

`validateUsername(username)` : `Promise`
ritorna una promise che risolve senza errori solamente se l'username è valido e non ne esiste uno di uguale già registrato

Argomenti:

* `username` : `string` l'username da validare

3.15.2.15 `CLIPS::server::urlrequesthandler::RankingProvider`

- **Descrizione:** provider della classifica per il percorso specificato
- **Utilizzo:** si occupa di restituire al richiedente la classifica del percorso specificato

3.15.2.16 CLIPS::server::urlrequesthandler::RegistrationFieldsValidator

- **Descrizione:** classe del server che si occupa di validare i dati di registrazione
- **Utilizzo:** la classe presenta un metodo `execute` che verifica la richiesta effettuata dal client e quali sono i dati forniti e risponde con le indicazioni di quali sono campi validi
- **Relazioni con altre classi:**
 - CLIPS::server::urlrequesthandler::EmailChecker: fornisce i metodi per la validazione di un indirizzo email
 - CLIPS::server::urlrequesthandler::PasswordChecker: verifica che la password soddisfi i requisiti minimi di sicurezza
 - CLIPS::server::urlrequesthandler::UsernameChecker: si occupa di verificare la validità dell'username (in particolare se è già in uso da un utente)

3.15.2.17 CLIPS::server::urlrequesthandler::RegistrationHandler

- **Descrizione:** classe che si occupa di registrare i nuovi utenti
- **Utilizzo:** è necessario impostare gli attributi `response` e `request` e successivamente chiamare il metodo `execute()`

3.15.2.18 CLIPS::server::urlrequesthandler::TokenGenerator

- **Descrizione:** classe per la creazione di stringhe casuali univoche
- **Utilizzo:** si occupa di creare un token casuale
- **Metodi:**
 - `generate() : string`
genera una stringa casuale univoca

3.15.2.19 CLIPS::server::urlrequesthandler::URLRequestHandler

- **Descrizione:** classe astratta che rappresenta un gestore di richieste di dati fatte al server
- **Utilizzo:** si occupa di gestire le richieste ricevute dal client
- **Attributi:**
 - `request: HTTP.Request` richiesta http effettuata da un client
 - `response: HTTP.Response` risposta che il server invia al client
- **Metodi:**
 - `execute() : void`
prende in carico la richiesta, ne studia i parametri in ingresso e invia al client la risposta

3.15.2.20 CLIPS::server::urlrequesthandler::URLRequestRouter

- **Descrizione:** classe che si occupa di creare il corretto URLRequestHandler per gestire la richiesta HTTP rivolta al server

- **Utilizzo:** crea la classe URLRequestHandler appropriata

- **Relazioni con altre classi:**

CLIPS::server::urlrequesthandler::AppInfoProvider: classe che gestisce la richiesta di informazioni sull'app (come la descrizione dell'app, l'indirizzo del sito di supporto, l'email del supporto tecnico ed altro)

CLIPS::server::urlrequesthandler::BuildingsProvider: classe che gestisce la richiesta degli edifici dal client al server

CLIPS::server::urlrequesthandler::LoginHandler: classe che gestisce le richieste di login da parte del client

CLIPS::server::urlrequesthandler::LogoutHandler: classe che gestisce la richiesta di logout da parte del client (eliminando il token associato al client)

CLIPS::server::urlrequesthandler::PathProvider: classe che gestisce la richiesta del percorso dal client

CLIPS::server::urlrequesthandler::PathResultsProvider: classe che gestisce la richiesta del risultato del percorso dal client

CLIPS::server::urlrequesthandler::RegistrationFieldsValidator: classe del server che si occupa di validare i dati di registrazione

CLIPS::server::urlrequesthandler::UserResultProvider: classe che gestisce la richiesta di un utente

3.15.2.21 CLIPS::server::urlrequesthandler::UserDataRequest

- **Descrizione:** classe che si occupa di gestire una richiesta che opera sui dati dell'utente

- **Utilizzo:** la classe mette a disposizione delle sottoclassi una varietà di metodi per gestire agevolmente i dati degli utenti

- **Metodi:**

`getToken() : string`

metodo che restituisce il token dell'utente che effettuata la richiesta dal client

`getUserID(token) : string`

metodo che a partire dal token di autenticazione ritorna l'id dell'utente che effettua la richiesta

Argomenti:

* `token : string` token di autenticazione dell'utente

3.15.2.22 CLIPS::server::urlrequesthandler::UsernameChecker

- **Descrizione:** si occupa di verificare la validità dell'username (in particolare se è già in uso da un utente)

- **Utilizzo:** è utile per verificare che un username non sia ancora stato utilizzato

3.15.2.23 CLIPS::server::urlrequesthandler::UserResultProvider

- **Descrizione:** classe che gestisce la richiesta di un utente
- **Utilizzo:** si occupa di restituire gli utenti richiesti dal client

4 Struttura del database

Viene illustrato di seguito lo schema del database *MySQL_g* presente nella parte server dell'applicazione.

Essendo un database relazionale le tabelle sono collegate da frecce che rappresentano le varie relazioni tra di esse; sono inoltre presentati i vari campi della tabella con relativo tipo.

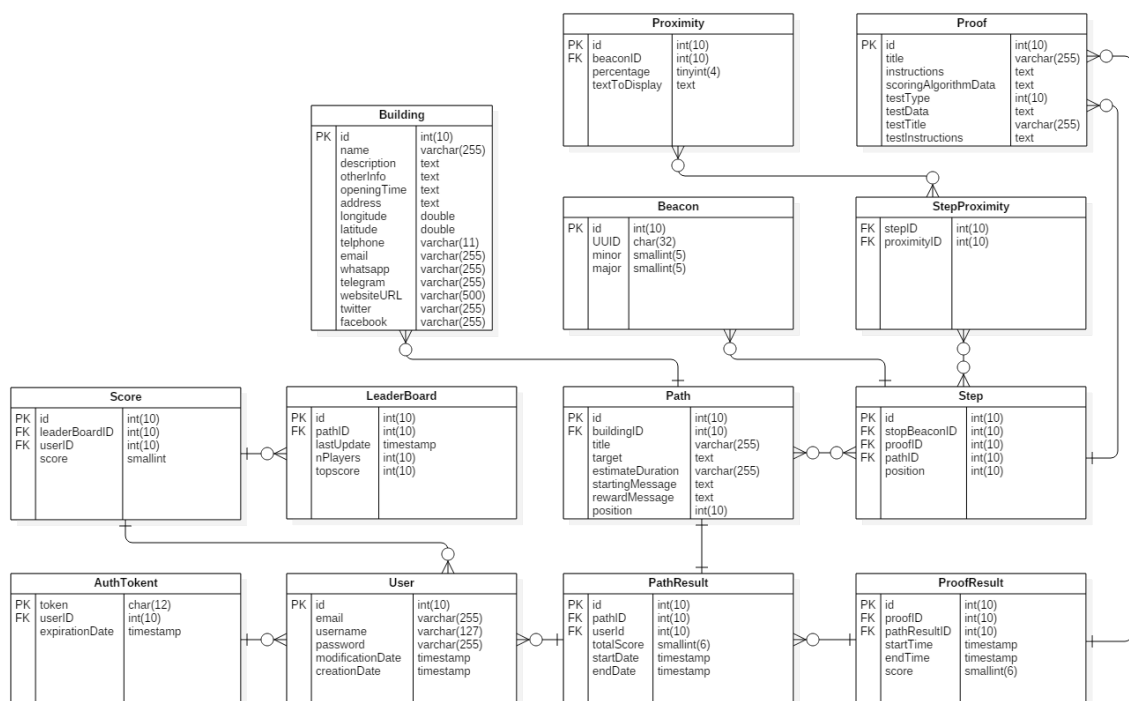


Figura 27: Struttura del database dell'applicazione

5 Diagrammi di Sequenza

In questa sezione saranno riportati i diagrammi di sequenza riguardo alle operazioni dell'applicazione ritenute significative che descrivono le interazioni tra attori ed oggetti o entità di sistema.

5.1 Registrazione

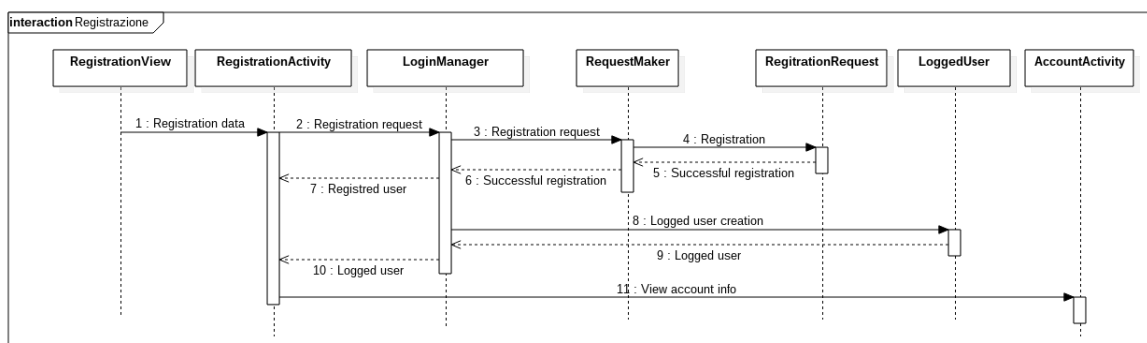


Figura 28: Diagramma di Sequenza della registrazione

Il diagramma in figura illustra come viene gestita la registrazione di un utente nel sistema. **RegistrationView** permette all'utente di inserire i dati di registrazione manualmente dall'applicazione. Successivamente i dati sono gestiti dalla classe **RegistrationActivity** la quale chiama il metodo "registration" di **LoginManager** la quale poi permetterà la registrazione dell'utente grazie alle classi **RequestMaker** e **RegistrationRequest**. **LoginManager** poi permette la creazione di un utente autenticato grazie alla creazione di un oggetto **LoggedUser**, quindi una volta registrato l'utente sarà anche automaticamente autenticato. La registrazione viene conclusa mostrando all'utente il proprio profilo, questo avviene grazie all'istanziamento della classe **AccountActivity**.

5.2 Login

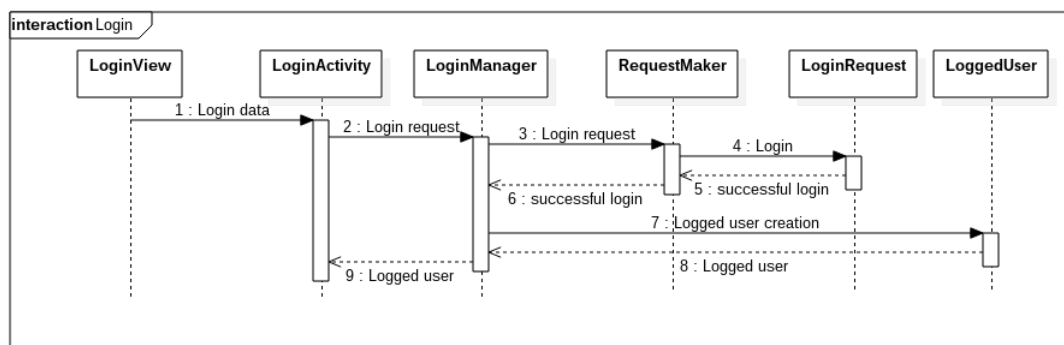


Figura 29: Diagramma di Sequenza della login

Il diagramma in figura illustra come viene gestita l'autenticazione di un utente nel sistema. L'utente inserisce i propri dati di login grazie a **LoginView**. Questi dati verranno poi gestiti da

`LoginActivity` la quale chiama il metodo “login” di `LoginManager` la quale permette l'autenticazione dell'utente grazie alle classi `RequestMaker` e `LoginRequest`. `LoginManager` poi crea un oggetto `LoggedUser` che descrive l'utente autenticato.

5.3 Richiesta lista edifici abilitati

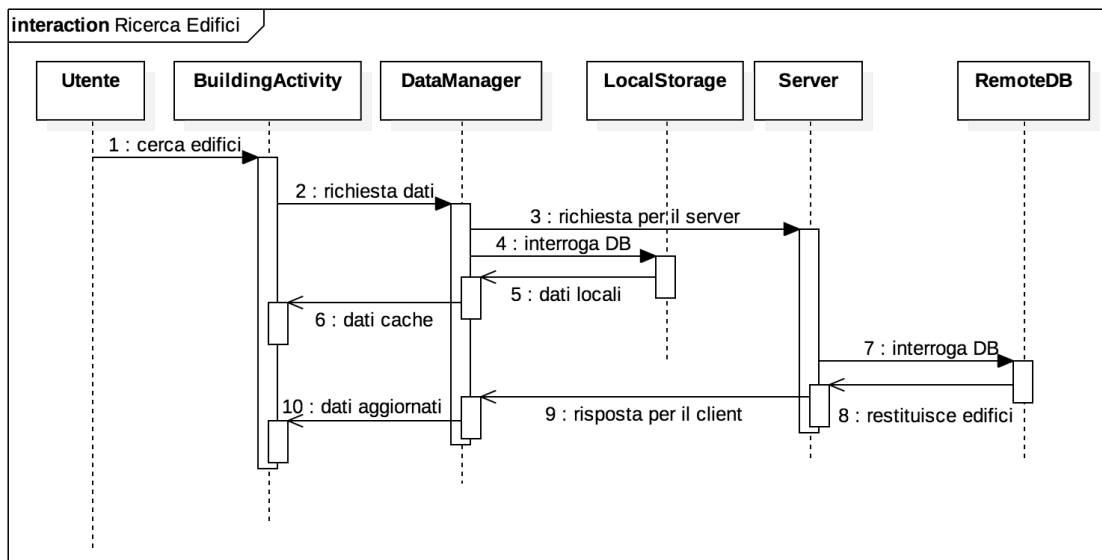


Figura 30: Diagramma di Sequenza della richiesta di edifici abilitati ai percorsi

Il diagramma in figura illustra come viene gestita la visualizzazione nel client della lista di edifici abilitati nelle vicinanze. La sequenza ha inizio ogni volta che l'utente accede alla schermata di ricerca degli edifici abilitati dal proprio terminale.

L'activity che si occupa di gestire la lista degli edifici (`BuildingsActivity`) richiede i dati attraverso la classe `DataManager`, situata nel package `CLIPS::data::datamanager`. A questo punto il `DataManager` interrogherà sia il database locale alla ricerca di dati in cache (per fornire nel più breve tempo possibile i dati all'utente), sia il server (per ottenere i dati più aggiornati). Solitamente il database locale risponderà in tempi brevissimi, ritornando quindi l'informazione richiesta al `DataManager` che provvederà a inviarla alla activity che l'ha richiesta. Intanto la richiesta inviata al server viene instradata dal `Server` alla corretta classe che si occuperà di interrogare il DB remoto per ottenere i dati più aggiornati. Tali dati vengono poi inviati dal server fino al `DataManager` e poi di nuovo all'activity, che avrà ora i dati più aggiornati da mostrare all'utente.

5.4 Scoperta nuovi *Beacon_g*

Il diagramma in figura illustra come viene gestita dall'app la situazione in cui, durante lo svolgimento di un percorso, l'utente si avvicina al *beacon_g* che segnala il raggiungimento della prova seguente.

Il *beacon_g* trasmette costantemente i propri dati nelle vicinanze; quando il dispositivo ne rivela la presenza lo segnala all'applicazione attraverso la libreria *AltBeacon_g* che effettua una chiamata al metodo `didEnterRegion` di un oggetto di tipo `BeaconDiscover`. A sua volta il `BeaconDiscover` chiamerà il metodo `didFoundBeacon` nel proprio delegate (ovvero un'istanza di `PathProgressController`) che si occuperà di capire se il *beacon_g* è effettivamente uno del percorso ed eventualmente notificarlo all'activity `SearchNewStepActivity` che ha il compito di informare l'utente.

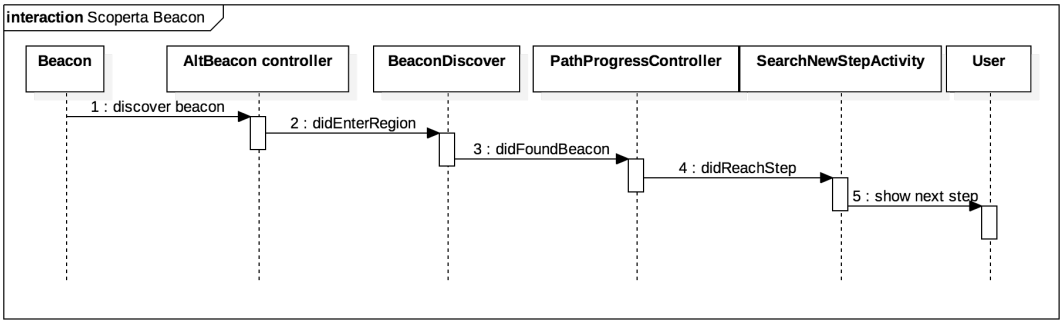


Figura 31: Diagramma di Sequenza della scoperta di nuovi beacon

6 Tabella tracciamento Requisiti-Componenti

Requisiti	Descrizione	Componenti
R0F1	L'utente deve poter possedere un account registrato nel database	CLIPS::server::dataserver CLIPS::server
R0F1.1	Il sistema deve permettere all'utente di autenticarsi	CLIPS::client ::authentication CLIPS::client CLIPS::client::data CLIPS::client::data::datamanager CLIPS::server CLIPS::client::data:urlrequest CLIPS::server::urlrequesthandler
R0F1.1.1	Il sistema deve chiedere all'utente l'email per la procedura di autenticazione	CLIPS::client ::authentication CLIPS::client
R0F1.1.2	Il sistema deve chiedere all'utente la password per la procedura di autenticazione	CLIPS::client ::authentication CLIPS::client
R0F1.1.3	Il sistema deve permettere all'utente di confermare i dati inseriti per la procedura di autenticazione	CLIPS::client ::authentication CLIPS::client
R0F1.1.4	Se i dati confermati dall'utente sono validi, egli viene autenticato	CLIPS::client ::authentication CLIPS::client CLIPS::client::data CLIPS::client::data::datamanager CLIPS::server CLIPS::client::data:urlrequest CLIPS::server::urlrequesthandler

Requisiti	Descrizione	Componenti
R0F1.1.5	Il sistema deve segnalare all'utente ogni eventuale errore durante la procedura di autenticazione	CLIPS::client ::authentication CLIPS::client CLIPS::client::data CLIPS::client::data::datamanager CLIPS::server CLIPS::client::data:urlrequest CLIPS::server::urlrequesthandler
R0F1.1.5.1	Il sistema interrompe la procedura di autenticazione se l'email non corrisponde a nessuna di quelle registrate	CLIPS::client ::authentication CLIPS::client CLIPS::client::data CLIPS::client::data::datamanager CLIPS::server CLIPS::client::data:urlrequest CLIPS::server::urlrequesthandler
R0F1.1.5.2	Il sistema interrompe la procedura di autenticazione se la password non corrisponde a quella associata all'email	CLIPS::client ::authentication CLIPS::client CLIPS::client::data CLIPS::client::data::datamanager CLIPS::server CLIPS::client::data:urlrequest CLIPS::server::urlrequesthandler
R0F1.1.5.3	L'applicazione informa l'utente dell'errore nell'inserimento dei dati	CLIPS::client ::authentication CLIPS::client
R0F1.1.5.4	L'utente può chiedere una nuova password inserendo l'email	CLIPS::client ::authentication

Requisiti	Descrizione	Componenti
R0F1.1.5.4.1	L'utente deve inserire l'email del proprio account per proseguire al recupero delle credenziali	CLIPS::client ::authentication CLIPS::client
R0F1.1.5.4.2	L'utente riceve un'email all'indirizzo inserito con una password casuale.	CLIPS::client ::authentication CLIPS::client
R0F1.1.5.4.3	La password dell'utente viene sostituita con la password inviata	CLIPS::server::dataserver CLIPS::server
R0F1.2	L'utente deve poter creare un proprio account	CLIPS::client ::authentication CLIPS::client CLIPS::client::data CLIPS::server::dataserver CLIPS::server CLIPS::client::data:urlrequest CLIPS::server::urlrequesthandler
R0F1.2.1	La registrazione richiede l'email all'utente	CLIPS::client ::authentication CLIPS::client
R0F1.2.2	La registrazione richiede l'username all'utente	CLIPS::client ::authentication CLIPS::client
R0F1.2.3	La registrazione richiede la password all'utente	CLIPS::client ::authentication CLIPS::client
R0F1.2.4	L'utente reinserisce la nuova password	CLIPS::client ::authentication CLIPS::client

Requisiti	Descrizione	Componenti
R0F1.2.5	Il sistema chiede all'utente di confermare i dati della propria registrazione	CLIPS::client ::authentication CLIPS::client
R0F1.2.6	Se i dati confermati dall'utente sono validi, il suo account viene registrato	CLIPS::client ::authentication CLIPS::client CLIPS::client::data CLIPS::server::dataserver CLIPS::server CLIPS::client::data:urlrequest CLIPS::server::urlrequesthandler
R0F1.2.6.1	L'utente viene automaticamente autenticato	CLIPS::client ::authentication CLIPS::client CLIPS::client::data CLIPS::server::dataserver CLIPS::server CLIPS::client::data:urlrequest CLIPS::server::urlrequesthandler
R0F1.2.7	Il sistema deve segnalare all'utente i vari problemi di registrazione	CLIPS::client ::authentication CLIPS::client CLIPS::client::data CLIPS::server::dataserver CLIPS::server CLIPS::client::data:urlrequest CLIPS::server::urlrequesthandler
R0F1.2.7.1	L'utente ha inserito un'email non valida	CLIPS::client ::authentication CLIPS::client CLIPS::server::dataserver CLIPS::server CLIPS::client::data:urlrequest CLIPS::server::urlrequesthandler

Requisiti	Descrizione	Componenti
R0F1.2.7.1.1	L'email deve avere un formato valido: deve contenere una @ preceduto da altri caratteri seguito da un dominio valido	CLIPS::server::dataserver CLIPS::server
R0F1.2.7.1.2	L'email non deve essere già in uso	CLIPS::server::dataserver CLIPS::server
R0F1.2.7.2	L'utente ha inserito un username non valido	CLIPS::client ::authentication CLIPS::client CLIPS::client::data CLIPS::server::dataserver CLIPS::server CLIPS::client::data:urlrequest CLIPS::server::urlrequesthandler
R0F1.2.7.2.1	L'username deve avere un formato valido: deve contenere solo caratteri alfanumerici	CLIPS::server::dataserver CLIPS::server
R0F1.2.7.2.2	L'username non deve essere già in uso da un altro utente	CLIPS::server::dataserver CLIPS::server
R0F1.2.7.3	La password deve contenere un minimo di 6 carattere ed un massimo di 16	CLIPS::server::dataserver CLIPS::server
R0F1.2.7.4	L'utente ha reinserito una password che non corrisponde a quella inserita in precedenza	CLIPS::server::dataserver CLIPS::server

Requisiti	Descrizione	Componenti
R0F2	L'utente deve poter fare il logout	CLIPS::client ::authentication CLIPS::client CLIPS::client::data CLIPS::client::data::datamanager CLIPS::server CLIPS::client::data:urlrequest CLIPS::server::urlrequesthandler
R0F3	L'utente deve poter effettuare un percorso tra quelli disponibili nel luogo in cui si trova	CLIPS::client ::CLIPS::client::viewController::building CLIPS::client CLIPS::client::data CLIPS::client::data::datamanager CLIPS::server::dataserver CLIPS::client::viewController::games CLIPS::client::location CLIPS::client::pathprogress CLIPS::client::data:urlrequest CLIPS::server::urlrequesthandler CLIPS::client::viewController
R0F3.1	L'utente gioca il percorso selezionato fino alla sua conclusione	CLIPS::client ::CLIPS::client::viewController::building CLIPS::client CLIPS::client::data CLIPS::client::data::datamanager CLIPS::client::viewController::games CLIPS::client::location CLIPS::client::pathprogress CLIPS::client::viewController
R0F3.1.1	L'utente cerca il beacon corrispondente alla prima stazione	CLIPS::client ::CLIPS::client::viewController::building CLIPS::client CLIPS::client::viewController::games CLIPS::client::location CLIPS::client::pathprogress CLIPS::client::viewController

Requisiti	Descrizione	Componenti
R0F3.1.1.1	Se il dispositivo dell'utente trova il beacon corrispondente alla stazione corretta l'utente può cominciare a giocare	CLIPS::client ::CLIPS::client::viewController::building CLIPS::client CLIPS::client::viewController::games CLIPS::client::pathprogress CLIPS::client::viewController
R0F3.1.1.2	I beacon che il dispositivo rileva non inerenti con la stazione cercata devono essere ignorati dall'app	CLIPS::client ::CLIPS::client::viewController::building CLIPS::client CLIPS::client::viewController::games CLIPS::client::viewController
R0F3.1.2	L'utente gioca la prova relativa a quella stazione	CLIPS::client CLIPS::client::viewController::games CLIPS::client::viewController
R0F3.1.2.1	La prova proposta all'utente è una fra quelle disponibili	CLIPS::server::dataserver CLIPS::server
R0F3.1.2.2	L'app mostra le istruzioni della prova	CLIPS::client CLIPS::client::viewController::games CLIPS::client::viewController
R0F3.1.2.3	L'utente deve poter svolgere la prova	CLIPS::client CLIPS::client::viewController::games CLIPS::client::viewController
R0F3.1.2.4	Finita la prova il dispositivo deve mostrare il risultato ottenuto	CLIPS::client CLIPS::client::viewController::games CLIPS::client::viewController

Requisiti	Descrizione	Componenti
R0F3.1.2.5	Finita la prova l'utente può proseguire il percorso o averlo terminato	CLIPS::client ::CLIPS::client::viewController::building CLIPS::client CLIPS::client::viewController::games CLIPS::client::pathprogress CLIPS::client::viewController
R0F3.1.2.5.1	Se la prova non è l'ultima l'app indica qual'è la prossima stazione	CLIPS::client CLIPS::client::viewController::games CLIPS::client::viewController
R0F3.1.2.5.1.1	L'app mostra all'utente le informazioni per trovare la prossima stazione	CLIPS::client CLIPS::client::viewController::games CLIPS::client::viewController
R0F3.1.2.5.2	Se la prova è l'ultima l'utente ha finito il percorso	CLIPS::client ::CLIPS::client::viewController::building CLIPS::client CLIPS::client::pathprogress CLIPS::client::viewController
R0F3.2	Quando il percorso è finito il dispositivo mostra all'utente il riepilogo dei dati suoi e generali	CLIPS::client CLIPS::client::viewController::savedresults CLIPS::client::viewController
R0F3.2.1	Il dispositivo mostra il tempo totale del percorso	CLIPS::client CLIPS::client::viewController::savedresults CLIPS::client::viewController
R0F3.2.2	Il dispositivo mostra il tempo totale impiegato per eseguire le prove	CLIPS::client CLIPS::client::viewController::savedresults CLIPS::client::viewController

Requisiti	Descrizione	Componenti
R0F3.2.2.1	Il tempo totale per eseguire le prove viene calcolato sommando i tempi impiegati per eseguire ogni prova (è il tempo totale senza considerare gli spostamenti effettuati per cambiare stazione)	CLIPS::client CLIPS::client::data
R0F3.2.2.1.1	La durata della prova viene misurata da quando si accetta di affrontarla fino alla conferma della soluzione	CLIPS::client CLIPS::client::data
R0F3.2.3	Il dispositivo mostra il punteggio totale ottenuto	CLIPS::client CLIPS::client::viewController::savedresults CLIPS::client::viewController
R0F3.2.3.1	Il punteggio totale viene calcolato sommando il numero di punti ottenuto in ogni singola prova	CLIPS::client CLIPS::client::data
R0F3.2.4	Il dispositivo mostra la posizione in classifica che l'utente ha raggiunto con il punteggio ottenuto	CLIPS::client CLIPS::client::viewController::savedresults CLIPS::client::viewController
R0F3.2.4.1	L'utente deve poter visualizzare la classifica generale	CLIPS::client::viewController::savedresults CLIPS::client::viewController

Requisiti	Descrizione	Componenti
R0F3.2.5	Il dispositivo visualizza la prova con il maggior numero di punti realizzati	CLIPS::client CLIPS::server::dataserver CLIPS::client::viewController
R0F3.2.6	Il dispositivo visualizza la prova con il minor numero di punti realizzati	CLIPS::client CLIPS::client::viewController::savedresults CLIPS::client::viewController
R0F3.3	L'utente autenticato può salvare il risultato ottenuto	CLIPS::client CLIPS::client::data::datamanager CLIPS::server::dataserver CLIPS::client::viewController::games CLIPS::server CLIPS::client::data:urlrequest CLIPS::server::urlrequesthandler CLIPS::client::viewController
R0F4	L'utente deve poter visualizzare alcune informazioni sull'app	CLIPS::client CLIPS::client::data CLIPS::client::data::datamanager CLIPS::server::dataserver CLIPS::server CLIPS::client::data:urlrequest CLIPS::server::urlrequesthandler CLIPS::client::viewController::utility
R0F4.1	L'utente deve poter visualizzare una schermata con una descrizione generale dell'app	CLIPS::client CLIPS::client::data CLIPS::client::data::datamanager CLIPS::server::dataserver CLIPS::server CLIPS::client::data:urlrequest CLIPS::server::urlrequesthandler CLIPS::client::viewController::utility

Requisiti	Descrizione	Componenti
R0F4.2	L'utente deve poter accedere alla pagina web relativa all'app tramite un link presente all'interno dell'app	CLIPS::client CLIPS::client::data CLIPS::client::data::datamanager CLIPS::server::dataserver CLIPS::server CLIPS::client::data:urlrequest CLIPS::server::urlrequesthandler CLIPS::client::viewController::utility
R0F4.3	L'utente deve poter inviare una segnalazione tramite mail in caso di errore	CLIPS::client CLIPS::client::data CLIPS::client::data::datamanager CLIPS::server::dataserver CLIPS::server CLIPS::client::data:urlrequest CLIPS::server::urlrequesthandler CLIPS::client::viewController::utility
R0F4.3.1	Quando l'utente clicca il pulsante per inviare la segnalazione viene aperta una nuova email da scrivere tramite il gestore di email predefinito. Nell'email il destinatario viene impostato con l'email destinata alle segnalazioni	CLIPS::client CLIPS::client::viewController::utility

Requisiti	Descrizione	Componenti
R0F5	L'utente autenticato deve poter visualizzare i risultati dei percorsi effettuati precedentemente	CLIPS::client ::CLIPS::client::viewController::building CLIPS::client CLIPS::client::data CLIPS::server::dataserver CLIPS::server CLIPS::client::data:urlrequest CLIPS::server::urlrequesthandler CLIPS::client::viewController
R0F5.1	Se l'utente è autenticato e ha percorsi salvati deve poterne vedere l'elenco	CLIPS::client ::CLIPS::client::viewController::building CLIPS::client CLIPS::client::data CLIPS::client::data::datamanager CLIPS::server::dataserver CLIPS::server CLIPS::client::data:urlrequest CLIPS::server::urlrequesthandler CLIPS::client::viewController
R0F5.1.1	Se l'utente clicca su un percorso l'app deve fornire tutte le informazioni salvate	CLIPS::client ::CLIPS::client::viewController::building CLIPS::client CLIPS::client::data CLIPS::client::data::datamanager CLIPS::server::dataserver CLIPS::server CLIPS::client::data:urlrequest CLIPS::server::urlrequesthandler CLIPS::client::viewController
R0F5.1.1.1	Il dispositivo deve visualizzare il nome del percorso	CLIPS::client ::CLIPS::client::viewController::building CLIPS::client CLIPS::client::data CLIPS::client::data::datamanager CLIPS::server::dataserver CLIPS::server CLIPS::client::data:urlrequest CLIPS::server::urlrequesthandler CLIPS::client::viewController

Requisiti	Descrizione	Componenti
R0F5.1.1.2	Il dispositivo deve visualizzare il nome del luogo dove si è svolto il percorso	CLIPS::client ::CLIPS::client::viewController::building CLIPS::client CLIPS::client::data CLIPS::client::data::datamanager CLIPS::server::dataserver CLIPS::server CLIPS::client::data:urlrequest CLIPS::server::urlrequesthandler CLIPS::client::viewController
R0F5.1.1.3	Il dispositivo deve visualizzare la CLIPS::client::data in cui si è svolto il percorso	CLIPS::client ::CLIPS::client::viewController::building CLIPS::client CLIPS::client::data CLIPS::client::data::datamanager CLIPS::server::dataserver CLIPS::server CLIPS::client::data:urlrequest CLIPS::server::urlrequesthandler
R0F5.1.1.4	Il dispositivo deve visualizzare il punteggio totale del percorso	CLIPS::client ::CLIPS::client::viewController::building CLIPS::client CLIPS::client::data CLIPS::client::data::datamanager CLIPS::server::dataserver CLIPS::server CLIPS::client::data:urlrequest CLIPS::server::urlrequesthandler CLIPS::client::viewController
R0F5.1.1.5	Il dispositivo deve visualizzare il tempo totale per lo svolgimento del percorso	CLIPS::client ::CLIPS::client::viewController::building CLIPS::client CLIPS::client::data CLIPS::client::data::datamanager CLIPS::server::dataserver CLIPS::server CLIPS::client::data:urlrequest CLIPS::server::urlrequesthandler CLIPS::client::viewController

Requisiti	Descrizione	Componenti
R0F5.1.1.6	Il dispositivo può visualizzare la posizione attuale nella classifica generale del risultato ottenuto	CLIPS::client CLIPS::client::viewController::savedresults CLIPS::server CLIPS::client::data:urlrequest CLIPS::server::urlrequesthandler CLIPS::client::viewController
R0F5.2	Se l'utente è autenticato ma non ha percorsi svolti salvati viene mostrato un invito a svolgere un percorso.	CLIPS::client CLIPS::client::viewController::savedresults CLIPS::client::viewController
R0F5.2.1	Nella spiegazione si informa l'utente che in quella schermata è possibile visualizzare i percorsi svolti quando si salveranno.	CLIPS::client ::CLIPS::client::viewController::building CLIPS::client CLIPS::client::viewController
R0F5.3	Se l'utente non è autenticato viene invitato ad autenticarsi.	CLIPS::client ::authentication CLIPS::client
R0F5.3.1	L'app deve spiegare che l'utente, quando è autenticato, può vedere l'elenco dei percorsi salvati in quella schermata.	CLIPS::client ::CLIPS::client::viewController::building CLIPS::client CLIPS::client::viewController

Requisiti	Descrizione	Componenti
R0F6	L'utente deve poter visualizzare dove si trovano gli edifici con percorsi più vicini alla sua posizione.	CLIPS::client ::CLIPS::client::viewController::building CLIPS::client CLIPS::client::data::datamanager CLIPS::server::dataserver CLIPS::client::location CLIPS::server CLIPS::client::data:urlrequest CLIPS::server::urlrequesthandler CLIPS::client::viewController
R0F6.1	L'utente deve poter cercare gli edifici con percorsi più vicini inserendo un raggio massimo.	CLIPS::client ::CLIPS::client::viewController::building CLIPS::client CLIPS::client::data CLIPS::client::data::datamanager CLIPS::server::dataserver CLIPS::server CLIPS::client::data:urlrequest CLIPS::server::urlrequesthandler
R0F6.1.1	L'utente deve poter inserire il raggio in chilometri per poter cercare gli edifici con percorsi più vicini.	CLIPS::client ::CLIPS::client::viewController::building CLIPS::client CLIPS::client::viewController
R0F6.1.1.1	Se la ricerca ha esito positivo vengono elencati tutti gli edifici con percorsi presenti nell'area cercata.	CLIPS::client ::CLIPS::client::viewController::building CLIPS::client CLIPS::client::data::datamanager CLIPS::server::dataserver CLIPS::client::location CLIPS::server CLIPS::client::data:urlrequest CLIPS::server::urlrequesthandler CLIPS::client::viewController

Requisiti	Descrizione	Componenti
R0F6.2	Cliccando su un edificio della lista l'utente deve poter visualizzare tutte le informazioni su di esso.	CLIPS::client ::CLIPS::client::viewController::building CLIPS::client CLIPS::client::data CLIPS::client::data::datamanager CLIPS::server::dataserver CLIPS::client::location CLIPS::server CLIPS::client::data:urlrequest CLIPS::server::urlrequesthandler CLIPS::client::viewController
R0F6.2.1	L'utente deve poter visualizzare delle informazioni specifiche sull'edificio.	CLIPS::client ::CLIPS::client::viewController::building CLIPS::client CLIPS::client::data CLIPS::client::data::datamanager CLIPS::server::dataserver CLIPS::client::location CLIPS::server CLIPS::client::data:urlrequest CLIPS::server::urlrequesthandler CLIPS::client::viewController
R0F6.2.1.1	L'utente deve poter visualizzare il nome dell'edificio.	CLIPS::client ::CLIPS::client::viewController::building CLIPS::client CLIPS::client::data CLIPS::client::data::datamanager CLIPS::server::dataserver CLIPS::client::location CLIPS::server CLIPS::client::data:urlrequest CLIPS::server::urlrequesthandler CLIPS::client::viewController

Requisiti	Descrizione	Componenti
R0F6.2.1.2	L'utente deve poter visualizzare la destinazione d'uso dell'edificio (ad esempio museo di storia egizia).	CLIPS::client ::CLIPS::client::viewController::building CLIPS::client CLIPS::client::data
R0F6.2.1.3	L'utente deve poter visualizzare l'indirizzo dell'edificio.	CLIPS::client ::CLIPS::client::viewController::building CLIPS::client CLIPS::client::data
R0F6.2.2	L'utente deve poter visualizzare il link al sito dell'edificio.	CLIPS::client ::CLIPS::client::viewController::building CLIPS::client CLIPS::client::data
R0F6.2.3	L'utente deve poter visualizzare alcuni contatti dell' edificio.	CLIPS::client ::CLIPS::client::viewController::building CLIPS::client CLIPS::client::data
R0F6.2.3.1	L'utente deve poter visualizzare il contatto telefonico dell'edificio.	CLIPS::client ::CLIPS::client::viewController::building CLIPS::client CLIPS::client::data
R0F6.2.3.2	L'utente deve poter contattare l'edificio per email.	CLIPS::client ::CLIPS::client::viewController::building CLIPS::client CLIPS::client::data
R0F6.2.3.3	L'utente deve poter contattare l'edificio tramite Facebook (se la struttura possiede un account Facebook).	CLIPS::client ::CLIPS::client::viewController::building CLIPS::client CLIPS::client::data

Requisiti	Descrizione	Componenti
R0F6.2.3.4	L'utente deve poter contattare l'edificio tramite Twitter (se la struttura possiede un account Twitter)	CLIPS::client ::CLIPS::client::viewController::building CLIPS::client CLIPS::client::data
R0F6.2.3.5	L'utente deve poter cotattare l'edificio tramite WhatsApp (se la struttura possiede un contatto pubblico di WhatsApp).	CLIPS::client ::CLIPS::client::viewController::building CLIPS::client CLIPS::client::data
R0F6.2.3.6	L'utente deve poter contattare l'edificio tramite Telegram (se la struttura ha un contatto pubblico di Telegram).	CLIPS::client ::CLIPS::client::viewController::building CLIPS::client CLIPS::client::data
R0F7	L'utente autenticato deve poter modificare le proprie credenziali d'accesso.	CLIPS::client ::authentication CLIPS::client CLIPS::client::data CLIPS::server::dataserver CLIPS::server CLIPS::client::data:urlrequest CLIPS::server::urlrequesthandler
R0F7.1	L'utente deve poter modificare la propria password.	CLIPS::client ::authentication CLIPS::client CLIPS::client::data CLIPS::server::dataserver CLIPS::server CLIPS::client::data:urlrequest CLIPS::server::urlrequesthandler

Requisiti	Descrizione	Componenti
R0F7.1.1	La password deve rispettare il requisito R0F1.2.7.3.	CLIPS::server::dataserver CLIPS::server
R0F7.2	L'utente deve poter modificare il proprio username.	CLIPS::client ::authentication CLIPS::client CLIPS::client::data CLIPS::server::dataserver CLIPS::server CLIPS::client::data:urlrequest CLIPS::server::urlrequesthandler
R0F7.2.1	L'username deve rispettare i requisiti R0F1.2.7.2.1 e R0F1.2.7.2.2.	CLIPS::server::dataserver CLIPS::server
R0F7.3	In caso di errore su qualche campo l'app deve informare l'utente	CLIPS::client ::authentication CLIPS::client CLIPS::server::dataserver CLIPS::server CLIPS::client::data:urlrequest CLIPS::server::urlrequesthandler

Tabella 1: Tracciamento requisiti-componenti

7 Tabella tracciamento Componenti-Requisiti

Componenti	Requisiti
CLIPS::client::authentication	R0F1.1
	R0F1.1.1
	R0F1.1.2
	R0F1.1.3
	R0F1.1.4
	R0F1.1.5
	R0F1.1.5.1
	R0F1.1.5.2
	R0F1.1.5.3
	R0F1.1.5.4
	R0F1.1.5.4.1
	R0F1.1.5.4.2
	R0F1.2
	R0F1.2.1
	R0F1.2.2
	R0F1.2.3
	R0F1.2.4
	R0F1.2.5
	R0F1.2.6
	R0F1.2.7
	R0F1.2.7.1
	R0F1.2.7.2
	R0F2
	R0F5.3
	R0F7
	R0F7.1
	R0F7.2
	R0F7.3

Componenti	Requisiti
CLIPS::client::viewController::building	R0F3
	R0F3.1
	R0F3.1.1
	R0F3.1.1.1
	R0F3.1.1.2
	R0F3.1.2.5
	R0F3.1.2.5.2
	R0F5
	R0F5.1
	R0F5.1.1
	R0F5.1.1.1
	R0F5.1.1.2
	R0F5.1.1.3
	R0F5.1.1.4
	R0F5.1.1.5
	R0F5.2.1
	R0F5.3.1
	R0F6
	R0F6.1
	R0F6.1.1
	R0F6.1.1.1
	R0F6.2
	R0F6.2.1
	R0F6.2.1.1
	R0F6.2.1.2
	R0F6.2.1.3
	R0F6.2.2
	R0F6.2.3
	R0F6.2.3.1
	R0F6.2.3.2
	R0F6.2.3.3
	R0F6.2.3.4
	R0F6.2.3.5
	R0F6.2.3.6

Componenti	Requisiti
CLIPS::client	R0F1.1
	R0F1.1.1
	R0F1.1.2
	R0F1.1.3
	R0F1.1.4
	R0F1.1.5
	R0F1.1.5.1
	R0F1.1.5.2
	R0F1.1.5.3
	R0F1.1.5.4.1
	R0F1.1.5.4.2
	R0F1.2
	R0F1.2.1
	R0F1.2.2
	R0F1.2.3
	R0F1.2.4
	R0F1.2.5
	R0F1.2.6
	R0F1.2.6.1
	R0F1.2.7
	R0F1.2.7.1
	R0F1.2.7.2
	R0F2
	R0F3
	R0F3.1
	R0F3.1.1
	R0F3.1.1.1
	R0F3.1.1.2
	R0F3.1.2
	R0F3.1.2.2
	R0F3.1.2.3
	R0F3.1.2.4
	R0F3.1.2.5
	R0F3.1.2.5.1
	R0F3.1.2.5.1.1

Componenti	Requisiti
CLIPS::client	R0F3.1.2.5.2
	R0F3.2
	R0F3.2.1
	R0F3.2.2
	R0F3.2.2.1
	R0F3.2.2.1.1
	R0F3.2.3
	R0F3.2.3.1
	R0F3.2.4
	R0F3.2.4.1
	R0F3.2.5
	R0F3.2.6
	R0F3.3
	R0F4
	R0F4.1
	R0F4.2
	R0F4.3
	R0F4.3.1
	R0F5
	R0F5.1
	R0F5.1.1
	R0F5.1.1.1
	R0F5.1.1.2
	R0F5.1.1.3
	R0F5.1.1.4
	R0F5.1.1.5
	R0F5.1.1.6
	R0F5.2
	R0F5.2.1
	R0F5.3
	R0F5.3.1
	R0F6
	R0F6.1
	R0F6.1.1
	R0F6.1.1.1

Componenti	Requisiti
CLIPS::client	R0F6.2
	R0F6.2.1
	R0F6.2.1.1
	R0F6.2.1.2
	R0F6.2.1.3
	R0F6.2.2
	R0F6.2.3
	R0F6.2.3.1
	R0F6.2.3.2
	R0F6.2.3.3
	R0F6.2.3.4
	R0F6.2.3.5
	R0F6.2.3.6
	R0F7
	R0F7.1
	R0F7.2
	R0F7.3
	R1F3.2.3.2

Componenti	Requisiti
CLIPS::client::data	R0F1.1
	R0F1.1.4
	R0F1.1.5
	R0F1.1.5.1
	R0F1.1.5.2
	R0F1.2
	R0F1.2.6
	R0F1.2.6.1
	R0F1.2.7.2
	R0F2
	R0F3
	R0F3.1
	R0F3.2.2.1
	R0F3.2.2.1.1
	R0F3.2.3.1
	R0F4
	R0F4.1
	R0F4.2
	R0F4.3
	R0F5
	R0F5.1
	R0F5.1.1
	R0F5.1.1.1
	R0F5.1.1.2
	R0F5.1.1.3
	R0F5.1.1.4
	R0F5.1.1.5
	R0F6.1
	R0F6.2
	R0F6.2.1
	R0F6.2.1.1
	R0F6.2.1.2
	R0F6.2.1.3
	R0F6.2.2
	R0F6.2.3

Componenti	Requisiti
CLIPS::client::data	R0F6.2.3.1
	R0F6.2.3.2
	R0F6.2.3.3
	R0F6.2.3.4
	R0F6.2.3.5
	R0F6.2.3.6
	R0F7
	R0F7.1
	R0F7.2

Componenti	Requisiti
CLIPS::client::data::datamanager	R0F1.1
	R0F1.1.4
	R0F1.2.6.1
	R0F2
	R0F3
	R0F3.1
	R0F3.3
	R0F4
	R0F4.1
	R0F4.2
	R0F4.3
	R0F5.1
	R0F5.1.1
	R0F5.1.1.1
	R0F5.1.1.2
	R0F5.1.1.3
	R0F5.1.1.4
	R0F5.1.1.5
	R0F6
	R0F6.1
	R0F6.1.1.1
	R0F6.2
	R0F6.2.1
	R0F6.2.1.1
	R0F1
	R0F1.1.5
	R0F1.1.5.1
	R0F1.1.5.4.3
	R0F1.2
	R0F1.2.6
	R0F1.2.6.1
	R0F1.2.7
	R0F1.2.7.1
	R0F1.2.7.1.1

Componenti	Requisiti
CLIPS::client::data::datamanager	R0F1.2.7.1.2
	R0F1.2.7.2
	R0F1.2.7.2.1
	R0F1.2.7.2.2
	R0F1.2.7.3
	R0F1.2.7.4
	R0F3
	R0F3.1.2.1
	R0F3.2.5
	R0F3.3
	R0F4
	R0F4.1
	R0F4.2
	R0F4.3
	R0F5
	R0F5.1
	R0F5.1.1
	R0F5.1.1.1
	R0F5.1.1.2
	R0F5.1.1.3
	R0F5.1.1.4
	R0F5.1.1.5
	R0F6
	R0F6.1
	R0F6.1.1.1
	R0F6.2.1
	R0F6.2.1.1
	R0F7
	R0F7.1
	R0F7.1.1
	R0F7.2
	R0F7.2.1
	R0F7.3

	Componenti	Requisiti	
Definizione di Prodotto v2.0.0	CLIPS::server::dataserver	R0F1	
		R0F1.1.5	
		R0F1.1.5.1	
		R0F1.1.5.4.3	
		R0F1.2	
		R0F1.2.6	
		R0F1.2.6.1	
		R0F1.2.7	
		R0F1.2.7.1	
		R0F1.2.7.2	
		R0F1.2.7.3	
		R0F1.2.7.4	
		R0F3	
		R0F3.1.2.1	
		R0F3.2.5	
		R0F3.3	
		R0F4	
		R0F4.1	
		R0F4.2	
		R0F4.3	
		R0F5	
		R0F5.1	
		R0F5.1.1	
		R0F5.1.1.1	
		R0F5.1.1.2	
		R0F5.1.1.3	
		R0F5.1.1.4	
		R0F5.1.1.5	
		R0F6	
		R0F6.1	
		R0F6.1.1.1	
		R0F6.2.1	
		R0F7	
		R0F7.1	
		R0F7.1.1	
		R0F7.2	
		R0F7.2.1	
Definizione di Prodotto v2.0.0	Beacon Strips	R0F7.3	Pagina 133 di 195

Componenti	Requisiti
CLIPS::client::viewController::games	R0F3
	R0F3.1
	R0F3.1.1
	R0F3.1.1.1
	R0F3.1.1.2
	R0F3.1.2
	R0F3.1.2.2
	R0F3.1.2.3
	R0F3.1.2.4
	R0F3.1.2.5
	R0F3.1.2.5.1
	R0F3.1.2.5.1.1
	R0F3.3
CLIPS::client::location	R0F3 R0F3.1
	R0F3.1.1
	R0F6
	R0F6.1.1.1
CLIPS::client::pathprogress	R0F3
	R0F3.1
	R0F3.1.1
	R0F3.1.1.1
	R0F3.1.2.5
	R0F3.1.2.5.2
CLIPS::viewController::savedresults	R0F3.2
	R0F3.2.1
	R0F3.2.2
	R0F3.2.3
	R0F3.2.4
	R0F3.2.4.1
	R0F3.2.6
	R0F5.1.1.6
	R0F5.2
	R1F3.2.3.2

Componenti	Requisiti
CLIPS::server	R0F1
	R0F1.1
	R0F1.1.4
	R0F1.1.5
	R0F1.1.5.1
	R0F1.1.5.2
	R0F1.1.5.4.3
	R0F1.2
	R0F1.2.6
	R0F1.2.6.1
	R0F1.2.7
	R0F1.2.7.1
	R0F1.2.7.1.1
	R0F1.2.7.1.2
	R0F1.2.7.2
	R0F1.2.7.2.1
	R0F1.2.7.2.2
	R0F1.2.7.3
	R0F1.2.7.4
	R0F2
	R0F3.1.2.1
	R0F3.3
	R0F4
	R0F4.1
	R0F4.2
	R0F4.3
	R0F5
	R0F5.1
	R0F5.1.1
	R0F5.1.1.1
	R0F5.1.1.2
	R0F5.1.1.3
	R0F5.1.1.4
	R0F5.1.1.5
	R0F5.1.1.6

Componenti	Requisiti
CLIPS::server	R0F6
	R0F6.1
	R0F6.1.1.1
	R0F6.2
	R0F6.2.1
	R0F6.2.1.1
	R0F7
	R0F7.1
	R0F7.1.1
	R0F7.2
	R0F7.2.1
	R0F7.3

Componenti	Requisiti
CLIPS::client::data:urlrequest	R0F1.1 R0F1.1.4
	R0F1.1.5
	R0F1.1.5.1
	R0F1.1.5.2
	R0F1.2
	R0F1.2.6
	R0F1.2.6.1
	R0F1.2.7
	R0F1.2.7.1
	R0F1.2.7.2
	R0F2
	R0F3
	R0F3.3
	R0F4
	R0F4.1
	R0F4.2
	R0F4.3
	R0F5
	R0F5.1
	R0F5.1.1
	R0F5.1.1.1
	R0F5.1.1.2
	R0F5.1.1.3
	R0F5.1.1.4
	R0F5.1.1.5
	R0F5.1.1.6
	R0F6
	R0F6.1
	R0F6.1.1.1
	R0F6.2
	R0F6.2.1
	R0F6.2.1.1
	R0F7
	R0F7.1
	R0F7.2
	R0F7.3

Componenti	Requisiti
CLIPS::server::urlrequesthandler	R0F1.1
	R0F1.1.4
	R0F1.1.5
	R0F1.1.5.1
	R0F1.1.5.2
	R0F1.2
	R0F1.2.6
	R0F1.2.6.1
	R0F1.2.7
	R0F1.2.7.1
	R0F1.2.7.2
	R0F2
	R0F3
	R0F3.3
	R0F4
	R0F4.1
	R0F4.2
	R0F4.3
	R0F5
	R0F5.1
	R0F5.1.1
	R0F5.1.1.1
	R0F5.1.1.2
	R0F5.1.1.3
	R0F5.1.1.4
	R0F5.1.1.5
	R0F5.1.1.6
	R0F6
	R0F6.1
	R0F6.1.1.1
	R0F6.2
	R0F6.2.1
	R0F6.2.1.1
	R0F7
	R0F7.1
	R0F7.2
	R0F7.3

Componenti	Requisiti
CLIPS::client::viewController::utility	R0F4
	R0F4.1
	R0F4.2
	R0F4.3
	R0F4.3.1

Componenti	Requisiti
CLIPS::client::viewController	R0F3
	R0F3.1
	R0F3.1.1
	R0F3.1.1.1
	R0F3.1.1.2
	R0F3.1.2
	R0F3.1.2.2
	R0F3.1.2.3
	R0F3.1.2.4
	R0F3.1.2.5
	R0F3.1.2.5.1
	R0F3.1.2.5.1.1
	R0F3.1.2.5.2
	R0F3.2
	R0F3.2.1
	R0F3.2.2
	R0F3.2.3
	R0F3.2.4
	R0F3.2.4.1
	R0F3.2.5
	R0F3.2.6
	R0F3.3
	R0F5
	R0F5.1
	R0F5.1.1
	R0F5.1.1.1
	R0F5.1.1.2
	R0F5.1.1.4
	R0F5.1.1.5
	R0F5.1.1.6
	R0F5.2
	R0F5.2.1
	R0F5.3.1
	R0F6

Componenti	Requisiti
CLIPS::client::viewController	R0F6.1.1
	R0F6.1.1.1
	R0F6.2
	R0F6.2.1
	R0F6.2.1.1
	R1F3.2.3.2

Tabella 2: Tracciamento componenti-requisiti

8 Tabella tracciamento Classi-Requisiti

Classe	Requisiti
CLIPS::client::viewController::authentication::AccountActivity	R0F2
CLIPS::client::viewController::authentication::AccountView	R0F2
CLIPS::client::viewController::authentication::ChangeProfileActivity	R0F7
	R0F7.2
	R0F7.2.1
	R0F7.3
CLIPS::client::viewController::authentication::ConfirmRegistration	R0F1.2.6
CLIPS::client::viewController::authentication::ForgotPasswordActivity	R0F1.1.5.4
CLIPS::client::viewController::authentication::ForgotPasswordView	R0F1.1.5.4
	R0F1.1.5.4.1
	R0F1.1.5.4.5
CLIPS::client::viewController::authentication::LoginActivity	R0F1.1
	R0F1.1.2
	R0F1.1.4
	R0F1.1.5
	R0F1.1.5.3
CLIPS::client::viewController::authentication::LoginView	R0F1.1
	R0F1.1.1
	R0F1.1.2
	R0F1.1.3
	R0F1.1.5
	R0F1.1.5.3
CLIPS::client::viewController::authentication::RecoverPassword	R0F1.1.5.4
CLIPS::client::viewController::authentication::RegistrationActivity	R0F1.2
	R0F1.2.6.1
	R0F1.2.7

Classe	Requisiti
CLIPS::client::viewController::authentication::RegistrationView	R0F1.2 R0F1.2.1 R0F1.2.2 R0F1.2.3 R0F1.2.4 R0F1.2.5 R0F1.2.7
CLIPS::client::viewController::authentication::UpdateUserInfoActivity	R0F7.1
CLIPS::client::viewController::authentication::UpdateUserInfoView	R0F7.1
CLIPS::client::viewController::building::BuildingActivity	R0F6.2 R0F6.2.1 R0F6.2.1.1 R0F6.2.1.2 R0F6.2.1.3 R0F6.2.2 R0F6.2.3 R0F6.2.3.1 R0F6.2.3.2 R0F6.2.3.3 R0F6.2.3.4 R0F6.2.3.5 R0F6.2.3.6 R2F6.2.1.4
CLIPS::client::viewController::building::BuildingRow	R0F6.1.1.1
CLIPS::client::viewController::building::BuildingSearchActivity	R0F6 R0F6.1 R0F6.1.1 R2F6.1.1.2
CLIPS::client::viewController::building::BuildingSearchView	R0F6 R0F6.1 R0F6.1.1 R0F6.1.1.1 R2F6.1.1.2

Classe	Requisiti
CLIPS::client::viewController::building::BuildingView	R0F6.2
	R0F6.2.1
	R0F6.2.1.1
	R0F6.2.1.2
	R0F6.2.1.3
	R0F6.2.2
	R0F6.2.3
	R0F6.2.3.1
	R0F6.2.3.2
	R0F6.2.3.3
	R0F6.2.3.4
	R0F6.2.3.5
	R0F6.2.3.6
CLIPS::client::viewController::building::PathActivity	R0F3
CLIPS::client::viewController::building::PathRow	R0F6.2
CLIPS::client::viewController::building::PathView	R0F3
CLIPS::client::viewController::building::SearchCriteriaRow	R0F6.1.1
CLIPS::client::data::/textitAbstractQuiz	R0F3
	R0F3.1
	R0F3.1.2
	R0F3.1.2.1
CLIPS::client::data::/textitTest	R0F3
	R0F3.1
	R0F3.1.2
	R0F3.1.2.1
CLIPS::client::data::AppInfo	R0F4
	R0F4.1
	R0F4.2
	R0F4.3

Classe	Requisiti
CLIPS::client::data::Beacon	R0F3
	R0F3.1
	R0F3.1.1
	R0F3.1.1.1
	R0F3.1.1.2
	R2F3.1.1.4
CLIPS::client::data::Building	R0F6
	R0F6.1
	R0F6.2.1
	R0F6.2.1.1
	R0F6.2.1.2
	R0F6.2.1.3
	R0F6.2.2
	R0F6.2.3
	R0F6.2.3.1
	R0F6.2.3.2
	R0F6.2.3.3
	R0F6.2.3.4
	R0F6.2.3.5
	R0F6.2.3.6
	R2F6.2.1.4
CLIPS::client::data::GameCollection	R0F3
	R0F3.1
	R0F3.1.2
	R0F3.1.2.1
CLIPS::client::data::LeaderBoard	R0F3.2.4.1
CLIPS::client::data::LinearScoringAlgorithm	R0F3
	R0F3.1
	R0F3.1.2
	R0F3.1.2.4
	R0F3.2
	R0F3.2.3
	R0F3.2.3.1
	R0F3.2.3.1.1

Classe	Requisiti
CLIPS::client::data::LoggedInUser	R0F1 R0F1.1 R0F1.1.4 R0F1.2 R0F1.2.6 R0F1.2.6.1 R0F2 R0F7 R0F7.1
CLIPS::client::data::MultipleChoiceImageQuiz	R0F3 R0F3.1 R0F3.1.2 R0F3.1.2.1 R1F3.1.1.3
CLIPS::client::data::MultipleChoiceTest	R0F3 R0F3.1 R0F3.1.2 R0F3.1.2.1
CLIPS::client::data::MultipleChoiceTextQuiz	R0F3 R0F3.1 R0F3.1.2 R0F3.1.2.1 R1F3.1.1.3
CLIPS::client::data::Path	R0F3 R0F3.1 R0F3.1.2.5 R0F3.1.2.5.1 R0F3.1.2.5.1.1 R0F3.1.2.5.2
CLIPS::client::data::PathInfo	R0F6 R0F6.1 R2F6.2.1.4

Classe	Requisiti
CLIPS::client::data::PathProgress	R0F3.2
	R0F3.2.1
	R0F3.2.1.1
	R0F3.2.2
	R0F3.2.2.1
	R0F3.2.2.1.1
	R0F3.2.3
CLIPS::client::data::PathResult	R0F3.2.3.1
	R0F3.2
	R0F3.2.1
	R0F3.2.2
	R0F3.2.2.1
	R0F3.2.3
	R0F3.2.5
	R0F3.2.6
	R0F3.3
	R0F5
	R0F5.1
	R0F5.1.1
	R0F5.1.1.1
	R0F5.1.1.2
	R0F5.1.1.3
	R0F5.1.1.4
	R0F5.1.1.5
	R1F3.2.3.2
CLIPS::client::data::Proof	R0F3
	R0F3.1
	R0F3.1.2
	R0F3.1.2.2
	R0F3.1.2.5
	R0F3.1.2.5.1
	R0F3.1.2.5.1.1
	R0F3.2.3.1.1

Classe	Requisiti
CLIPS::client::data::ProofResult	R0F3.1.2.4 R0F3.2 R0F3.2.2 R0F3.2.2.1 R0F3.2.2.1.1 R0F3.2.3 R0F3.2.3.1 R0F3.2.3.1.1 R0F3.2.5 R0F3.2.6 R0F3.3 R0F5 R0F5.1
CLIPS::client::data::Proximity	R0F3 R0F3.1 R2F3.1.1.4 R2F3.1.1.4.1
CLIPS::client::data::Score	R0F3.2.4 R0F3.2.4.1 R0F5.1.1.6
CLIPS::client::data::ScoringAlgorithmFactory	R0F3 R0F3.1
CLIPS::client::data::Step	R0F3 R0F3.1 R0F3.1.1 R0F3.1.1.1 R0F3.1.1.2 R0F3.1.2 R0F3.1.2.5 R0F3.1.2.5.1 R0F3.1.2.5.1.1 R0F3.1.2.5.2
CLIPS::client::data::TestBuilder	R0F3 R0F3.1

Classe	Requisiti
CLIPS::client::data::TrueFalseImageQuiz	R0F3 R0F3.1 R0F3.1.2 R0F3.1.2.1 R1F3.1.1.3
CLIPS::client::data::TrueFalseTest	R0F3 R0F3.1 R0F3.1.2 R0F3.1.2.1
CLIPS::client::data::TrueFalseTextQuiz	R0F3 R0F3.1 R0F3.1.2 R0F3.1.2.1 R1F3.1.1.3
CLIPS::client::data::Utility	R0F3 R0F3.1 R0F5 R0F5.1 R0F6 R0F6.1
CLIPS::client::data::datamanager::/textitSaveDataManager	R0F3.3

Classe	Requisiti
CLIPS::client::data::datamanager::AbstractDataManagerListener	R0F1
	R0F1.1
	R0F1.1.4
	R0F1.1.5
	R0F1.1.5.1
	R0F1.1.5.2
	R0F1.1.5.4
	R0F1.1.5.4.5
	R0F1.2
	R0F1.2.6
	R0F1.2.7
	R0F2
	R0F3
	R0F3.1
	R0F3.2.4
	R0F3.2.4.1
	R0F3.3
	R0F4
	R0F5
	R0F5.1
	R0F5.1.1.6
	R0F6
	R0F6.1
	R0F7
	R0F7.1
	R0F7.2
	R0F7.3
	R1F3.2.3.2
CLIPS::client::data::datamanager::AppInfoDataRequest	R0F4
CLIPS::client::data::datamanager::BuildingsDataRequest	R0F6
	R0F6.1
CLIPS::client::data::datamanager::CheckResult	R0F1.2.7

Classe	Requisiti
CLIPS::client::data::datamanager::DataManager	R0F3 R0F3.1 R0F3.2.4 R0F3.2.4.1 R0F4 R0F5 R0F5.1 R0F5.1.1.6 R0F6 R0F6.1 R1F3.2.3.2
CLIPS::client::data::datamanager::DataRequestMaker	R0F3.1 R0F3.2.4 R0F3.2.4.1 R0F3.3 R0F4 R0F5 R0F5.1 R0F5.1.1.6 R0F6 R0F6.1 R1F3.2.3.2
CLIPS::client::data::datamanager::DBHandlerFIXME	R0F3 R0F5 R0F5.1 R0F6 R0F6.1 R1F3.2.3.2
CLIPS::client::data::datamanager::GetRankingDataRequest	R0F3.2.4 R0F3.2.4.1 R0F5.1.1.6
CLIPS::client::data::datamanager::GetResultsDataRequest	R0F5 R0F5.1 R1F3.2.3.2

Classe	Requisiti
CLIPS::client::data::datamanager::LoginManager	R0F1
	R0F1.1
	R0F1.1.4
	R0F1.1.5
	R0F1.1.5.1
	R0F1.1.5.2
	R0F1.1.5.4
	R0F1.1.5.4.5
	R0F1.2
	R0F1.2.6
	R0F1.2.6.1
	R0F1.2.7
	R0F2
	R0F7
CLIPS::client::data::datamanager::PathDataRequest	R0F7.1
	R0F7.2
CLIPS::client::data::datamanager::SaveResultDataRequest	R0F7.3
	R0F3
CLIPS::client::viewcontroller::games::MultipleChoiceQuizActivity	R0F3.1
	R0F3.3
CLIPS::client::viewcontroller::games::MultipleChoicew	R0F3.1.2.3
	R0F3.1.2.3
CLIPS::client::viewcontroller::games::ProofActivity	R0F3.1
	R0F3.1.2
CLIPS::client::viewcontroller::games::ProofResultActivity	R0F3.1.2.2
	R0F3.1.2.4
CLIPS::client::viewcontroller::games::ProofResultView	R0F3.1.2.4
	R0F3.1.2
CLIPS::client::viewcontroller::games::ProofView	R0F3.1.2.2
	R0F3.1.2.2

Classe	Requisiti
CLIPS::client::viewController::games::SearchNewStepActivity	R0F3.1 R0F3.1.1 R0F3.1.1.1 R0F3.1.1.2 R0F3.1.2.5.1 R0F3.1.2.5.1.1
CLIPS::client::viewController::games::SearchNewStepView	R0F3.1.1 R0F3.1.2.5.1 R0F3.1.2.5.1.1
CLIPS::client::viewController::games::TrueFalseQuizActivity	R0F3.1.2.3
CLIPS::client::viewController::games::TrueFalseQuizView	R0F3.1.2.3
CLIPS::client::pathprogress::BeaconDiscover	R0F3 R0F3.1 R0F3.1.1 R0F3.1.1.1
CLIPS::client::pathprogress::GPSListener	R0F6 R0F6.1
CLIPS::client::pathprogress::GPSManager	R0F6 R0F6.1
CLIPS::client::pathprogress::PathProgressController	R0F3.1 R0F3.1.1 R0F3.1.1.1 R0F3.1.1.2 R0F3.1.2.5 R0F3.1.2.5.1 R0F3.1.2.5.1.1 R0F3.1.2.5.2
CLIPS::client::pathprogress::PathProgressControllerDelegate	R0F3.1.1 R0F3.1.1.1 R0F3.1.1.2
CLIPS::client::pathprogress::PathProgressMaker	R0F6 R0F6.1

Classe	Requisiti
CLIPS::client::viewController::savedresults::RankingActivity	R0F3.2.4
CLIPS::client::viewController::savedresults::RankingView	R0F3.2.4
CLIPS::client::viewController::savedresults::ResultActivity	R0F5
CLIPS::client::viewController::savedresults::ResultRow	R0F3.2.4.1
CLIPS::client::viewController::savedresults::ResultView	R0F5.3
CLIPS::client::viewController::savedresults::SavedResultView	R0F5.1
	R0F5.1.1
	R0F5.1.1.1
	R0F5.1.1.2
	R0F5.1.1.3
	R0F5.1.1.4
	R0F5.1.1.5
	R0F5.1.1.6
	R0F5.2

Classe	Requisiti
CLIPS::client::data::urlrequest::AbstractUrlRequestListener	R0F1
	R0F1.1
	R0F1.1.4
	R0F1.1.5
	R0F1.1.5.1
	R0F1.1.5.2
	R0F1.1.5.4
	R0F1.1.5.4.5
	R0F1.2
	R0F1.2.6
	R0F1.2.7
	R0F2
	R0F3
	R0F3.1
	R0F3.2.4
	R0F3.2.4.1
	R0F3.3
	R0F4
	R0F5
	R0F5.1
	R0F5.1.1.6
	R0F6
	R0F6.1
	R0F7
	R0F7.1
	R0F7.2
	R0F7.3
	R1F3.2.3.2
CLIPS::client::data::urlrequest::AppInfoRequest	R0F4
CLIPS::client::data::urlrequest::BuildingsRequest	R0F6
	R0F6.1
CLIPS::client::data::urlrequest::ChangeProfileDataRequest	R0F7
	R0F7.1
	R0F7.2

Classe	Requisiti
CLIPS::client::data::urlrequest::CheckRequest	R0F1.2.7
CLIPS::client::data::urlrequest::ForgotPasswordRequest	R0F1.1.5.4
CLIPS::client::data::urlrequest::GetRankingRequest	R0F3.2.4 R0F3.2.4.1 R0F5.1.1.6
CLIPS::client::data::urlrequest::GetResultsRequest	R0F5 R0F5.1 R1F3.2.3.2
CLIPS::client::data::urlrequest::LoginRequest	R0F1 R0F1.1 R0F1.1.4 R0F1.1.5
CLIPS::client::data::urlrequest::LogoutRequest	R0F2
CLIPS::client::data::urlrequest::PathRequest	R0F3 R0F3.1
CLIPS::client::data::urlrequest::RegistrationRequest	R0F1 R0F1.2

Classe	Requisiti
CLIPS::client::data::urlrequest::RequestMaker	R0F1
	R0F1.1
	R0F1.1.5.4
	R0F1.2
	R0F1.2.7
	R0F2
	R0F3
	R0F3.1
	R0F3.2.4
	R0F3.2.4.1
	R0F3.3
	R0F4
	R0F5
	R0F5.1
	R0F5.1.1.6
	R0F6
	R0F6.1
	R0F7
	R0F7.1
	R0F7.2
	R1F3.2.3.2
CLIPS::client::data::urlrequest::SaveResultRequest	R0F3.3

Classe	Requisiti
CLIPS::client::data::urlrequest::ServerError	R0F1
	R0F1.1
	R0F1.1.5
	R0F1.1.5.1
	R0F1.1.5.2
	R0F1.1.5.4
	R0F1.1.5.4.5
	R0F1.2
	R0F1.2.7
	R0F2
	R0F3
	R0F3.1
	R0F3.2.4
	R0F3.2.4.1
	R0F3.3
	R0F4
	R0F5
	R0F5.1
	R0F5.1.1.6
	R0F6
	R0F6.1
	R0F7
	R0F7.1
	R0F7.2
	R0F7.3
	R1F3.2.3.2

Classe	Requisiti
CLIPS::client::data::urlrequest::URLDataConstants	R0F1
	R0F1.1
	R0F1.1.5.4
	R0F1.2
	R0F1.2.7
	R0F2
	R0F3
	R0F3.1
	R0F3.2.4
	R0F3.2.4.1
	R0F3.3
	R0F4
	R0F5
	R0F5.1
	R0F5.1.1.6
	R0F6
	R0F6.1
	R0F7
	R0F7.1
	R0F7.2
	R1F3.2.3.2

Classe	Requisiti
CLIPS::client::data::urlrequest::URLRequest	R0F1
	R0F1.1
	R0F1.1.4
	R0F1.1.5
	R0F1.1.5.1
	R0F1.1.5.2
	R0F1.1.5.4
	R0F1.1.5.4.5
	R0F1.2
	R0F1.2.6
	R0F1.2.7
	R0F2
	R0F3
	R0F3.1
	R0F3.2.4
	R0F3.2.4.1
	R0F3.3
	R0F4
	R0F5
	R0F5.1
	R0F5.1.1.6
	R0F6
	R0F6.1
	R0F7
	R0F7.1
	R0F7.2
	R0F7.3
	R1F3.2.3.2
CLIPS::server::urlrequesthandler::AppInfoProvider	R0F4

Classe	Requisiti
CLIPS::server::urlrequesthandler::BuildingsProvider	R0F6
	R0F6.1.1.1
	R0F6.2
	R0F6.2.1
	R0F6.2.1.1
	R0F6.2.1.2
	R0F6.2.1.3
CLIPS::server::urlrequesthandler::DBHandler	R0F1.1
	R0F1.1.4
	R0F1.1.5
	R0F1.1.5.4.2
	R0F1.1.5.4.3
	R0F1.2.6
	R0F1.2.6.1
	R0F2
	R0F3
	R0F3.1
	R0F3.2.4
	R0F3.2.4.1
	R0F3.3
	R0F4
	R0F5
	R0F5.1
	R0F6
	R0F6.1.1.1
	R0F7
	R2F6.1.1.2
CLIPS::server::urlrequesthandler::EmailChecker	R0F1.1.4
	R0F1.1.5.1
	R0F1.1.5.4.4
	R0F1.1.5.4.5
	R0F1.2.7.1
	R0F1.2.7.1.1
	R0F1.2.7.1.2

Classe	Requisiti
CLIPS::server::urlrequesthandler::EmailSender	R0F1.1.5.4 R0F1.1.5.4.3
CLIPS::server::urlrequesthandler::GetUserData	R0F7
CLIPS::server::urlrequesthandler::LoginHandler	R0F1.1 R0F1.1.4 R0F1.1.5 R0F1.1.5.1 R0F1.1.5.2
CLIPS::server::urlrequesthandler::LogoutHandler	R0F2
CLIPS::server::urlrequesthandler::PasswordChecker	R0F1.1.5.2 R0F1.2.6 R0F1.2.7.3 R0F1.2.7.4 R0F7.2.1
CLIPS::server::urlrequesthandler::PasswordResetEmailSender	R0F1.1.5.4.2
CLIPS::server::urlrequesthandler::PasswordResetHandler	R0F1.1.5.4 R0F1.1.5.4.4
CLIPS::server::urlrequesthandler::PathProvider	R0F3 R0F3.1 R0F6.1.1.1
CLIPS::server::urlrequesthandler::PathResultsProvider	R0F5 R0F5.1 R0F5.1.1 R0F5.2
CLIPS::server::urlrequesthandler::PostUserData	R0F7 R0F7.3
CLIPS::server::urlrequesthandler::RankingProvider	R0F5.1.1
CLIPS::server::urlrequesthandler::RegistrationFieldsValidator	R0F1.2 R0F1.2.6 R0F1.2.7

Classe	Requisiti
CLIPS::server::urlrequesthandler::RegistrationHandler	R0F1.2
	R0F1.2.6
	R0F1.2.7
CLIPS::server::urlrequesthandler::TokenGenerator	R0F1.1.4
	R0F1.2.6.1
CLIPS::server::urlrequesthandler::URLRequestHandler	R0F1.1
	R0F1.1.4
	R0F1.1.5
	R0F1.1.5.4.2
	R0F1.1.5.4.3
	R0F1.2.6
	R0F1.2.6.1
	R0F1.2.7
	R0F2
	R0F3
	R0F3.2.4
	R0F3.2.4.1
	R0F3.3
	R0F4
	R0F5
	R0F5.1
	R0F6
	R0F6.1.1.1
	R0F7
	R2F6.1.1.2

Classe	Requisiti
CLIPS::server::urlrequesthandler::URLRequestRouter	R0F1.1
	R0F1.1.4
	R0F1.1.5
	R0F1.1.5.4.2
	R0F1.1.5.4.3
	R0F1.2.6
	R0F1.2.6.1
	R0F1.2.7
	R0F2
	R0F3
	R0F3.2.4
	R0F3.2.4.1
	R0F3.3
	R0F4
	R0F5
	R0F5.1
	R0F6
	R0F6.1.1.1
	R0F7

Classe	Requisiti
CLIPS::server::urlrequesthandler::UserDataRequest	R0F1.1
	R0F1.1.4
	R0F1.1.5
	R0F1.1.5.4.2
	R0F1.1.5.4.3
	R0F1.2.6
	R0F1.2.6.1
	R0F2
	R0F3
	R0F3.2.4
	R0F3.2.4.1
	R0F3.3
	R0F4
	R0F5
	R0F5.1
	R0F6
CLIPS::server::urlrequesthandler::UsernameChecker	R0F6.1.1.1
	R0F7
	R2F6.1.1.2
	R0F1.2.7.2
CLIPS::server::urlrequesthandler::UserResultProvider	R0F1.2.7.2.1
	R0F1.2.7.2.2
	R0F7.1.1
CLIPS::client::viewController::utility::AppInfoActivity	R0F5
	R0F5.1
	R0F4
	R0F4.1
	R0F4.2
	R0F4.3
	R0F4.3.1

Classe	Requisiti
CLIPS::client::viewController::utility::AppInfoView	R0F4
	R0F4.1
	R0F4.2
	R0F4.3
	R0F4.3.1
CLIPS::client::viewController::utility::MenuActivity	R0F2
CLIPS::client::viewController::utility::MenuView	R0F2

Tabella 3: Tracciamento classi-requisiti

9 Tabella tracciamento Requisiti-Classi

Requisito	Classi
R0F1	CLIPS::client::data::datamanager::AbstractDataManagerListener CLIPS::client::data::urlrequest::AbstractUrlRequestListener CLIPS::client::data::LoggedUser CLIPS::client::data::datamanager::LoginManager CLIPS::client::data::urlrequest::LoginRequest CLIPS::client::data::urlrequest::RegistrationRequest CLIPS::client::data::urlrequest::RequestMaker CLIPS::client::data::urlrequest::ServerError CLIPS::client::data::urlrequest::URLDataConstants CLIPS::client::data::urlrequest::URLRequest CLIPS::server::dataserver::UserServer
R0F1.1	CLIPS::client::data::datamanager::AbstractDataManagerListener CLIPS::client::data::urlrequest::AbstractUrlRequestListener CLIPS::server::urlrequesthandler::DBHandler CLIPS::client::data::LoggedUser CLIPS::client::viewController::authentication::LoginActivity CLIPS::server::urlrequesthandler::LoginHandler CLIPS::client::data::datamanager::LoginManager CLIPS::client::data::urlrequest::LoginRequest CLIPS::client::viewController::authentication::LoginView CLIPS::client::data::urlrequest::RequestMaker CLIPS::client::data::urlrequest::ServerError CLIPS::client::data::urlrequest::URLDataConstants CLIPS::client::data::urlrequest::URLRequest CLIPS::server::urlrequesthandler::URLRequestHandler CLIPS::server::urlrequesthandler::URLRequestRouter CLIPS::server::urlrequesthandler::UserDataRequest
R0F1.1.1	CLIPS::client::viewController::authentication::LoginView
R0F1.1.2	CLIPS::client::viewController::authentication::LoginActivity CLIPS::client::viewController::authentication::LoginView
R0F1.1.3	CLIPS::client::viewController::authentication::LoginView

Requisito	Classi
R0F1.1.4	CLIPS::client::data::datamanager::AbstractDataManagerListener CLIPS::client::data::urlrequest::AbstractUrlRequestListener CLIPS::server::urlrequesthandler::DBHandler CLIPS::server::urlrequesthandler::EmailChecker CLIPS::client::data::LoggedInUser CLIPS::client::viewController::authentication::LoginActivity CLIPS::server::urlrequesthandler::LoginHandler CLIPS::client::data::datamanager::LoginManager CLIPS::client::data::urlrequest::LoginRequest CLIPS::server::urlrequesthandler::TokenGenerator CLIPS::client::data::urlrequest::URLRequest CLIPS::server::urlrequesthandler::URLRequestHandler CLIPS::server::urlrequesthandler::URLRequestRouter CLIPS::server::urlrequesthandler::UserDataRequest
R0F1.1.5	CLIPS::client::data::datamanager::AbstractDataManagerListener CLIPS::client::data::urlrequest::AbstractUrlRequestListener CLIPS::server::urlrequesthandler::DBHandler CLIPS::client::viewController::authentication::LoginActivity CLIPS::server::urlrequesthandler::LoginHandler CLIPS::client::data::datamanager::LoginManager CLIPS::client::data::urlrequest::LoginRequest CLIPS::client::viewController::authentication::LoginView CLIPS::client::data::urlrequest::ServerError CLIPS::client::data::urlrequest::URLRequest CLIPS::server::urlrequesthandler::URLRequestHandler CLIPS::server::urlrequesthandler::URLRequestRouter CLIPS::server::urlrequesthandler::UserDataRequest
R0F1.1.5.1	CLIPS::client::data::datamanager::AbstractDataManagerListener CLIPS::client::data::urlrequest::AbstractUrlRequestListener CLIPS::server::urlrequesthandler::EmailChecker CLIPS::server::urlrequesthandler::LoginHandler CLIPS::client::data::datamanager::LoginManager CLIPS::client::data::urlrequest::ServerError CLIPS::client::data::urlrequest::URLRequest

Requisito	Classi
R0F1.1.5.2	CLIPS::client::data::datamanager::AbstractDataManagerListener CLIPS::client::data::urlrequest::AbstractUrlRequestListener CLIPS::server::urlrequesthandler::LoginHandler CLIPS::client::data::datamanager::LoginManager CLIPS::server::urlrequesthandler::PasswordChecker CLIPS::client::data::urlrequest::ServerError CLIPS::client::data::urlrequest::URLRequest
R0F1.1.5.3	CLIPS::client::viewController::authentication::LoginActivity CLIPS::client::viewController::authentication::LoginView
R0F1.1.5.4	CLIPS::client::data::datamanager::AbstractDataManagerListener CLIPS::client::data::urlrequest::AbstractUrlRequestListener CLIPS::server::urlrequesthandler::EmailSender CLIPS::client::viewController::authentication::ForgotPasswordActivity CLIPS::client::data::urlrequest::ForgotPasswordRequest CLIPS::client::viewController::authentication::ForgotPasswordView CLIPS::client::data::datamanager::LoginManager CLIPS::server::urlrequesthandler::PasswordResetHandler CLIPS::client::viewController::authentication::RecoverPassword CLIPS::client::data::urlrequest::RequestMaker CLIPS::client::data::urlrequest::ServerError CLIPS::client::data::urlrequest::URLDataConstants CLIPS::client::data::urlrequest::URLRequest
R0F1.1.5.4.1	CLIPS::client::viewController::authentication::ForgotPasswordView
R0F1.1.5.4.2	CLIPS::server::urlrequesthandler::DBHandler CLIPS::server::urlrequesthandler::PasswordResetEmailSender CLIPS::server::urlrequesthandler::URLRequestHandler CLIPS::server::urlrequesthandler::URLRequestRouter CLIPS::server::urlrequesthandler::UserDataRequest
R0F1.1.5.4.3	CLIPS::server::urlrequesthandler::DBHandler CLIPS::server::urlrequesthandler::EmailSender CLIPS::server::urlrequesthandler::URLRequestHandler CLIPS::server::urlrequesthandler::URLRequestRouter CLIPS::server::urlrequesthandler::UserDataRequest

Requisito	Classi
R0F1.1.5.4.4	CLIPS::server::urlrequesthandler::EmailChecker CLIPS::server::urlrequesthandler::PasswordResetHandler
R0F1.1.5.4.5	CLIPS::client::data::datamanager::AbstractDataManagerListener CLIPS::client::data::urlrequest::AbstractUrlRequestListener CLIPS::server::urlrequesthandler::EmailChecker CLIPS::client::viewController::authentication::ForgotPasswordView CLIPS::client::data::datamanager::LoginManager CLIPS::client::data::urlrequest::ServerError CLIPS::client::data::urlrequest::URLRequest
R0F1.2	CLIPS::client::data::datamanager::AbstractDataManagerListener CLIPS::client::data::urlrequest::AbstractUrlRequestListener CLIPS::client::data::LoggedInUser CLIPS::client::data::datamanager::LoginManager CLIPS::client::viewController::authentication::RegistrationActivity CLIPS::server::urlrequesthandler::RegistrationFieldsValidator CLIPS::server::urlrequesthandler::RegistrationHandler CLIPS::client::data::urlrequest::RegistrationRequest CLIPS::client::viewController::authentication::RegistrationView CLIPS::client::data::urlrequest::RequestMaker CLIPS::client::data::urlrequest::ServerError CLIPS::client::data::urlrequest::URLDataConstants CLIPS::client::data::urlrequest::URLRequest
R0F1.2.1	CLIPS::client::viewController::authentication::RegistrationView
R0F1.2.2	CLIPS::client::viewController::authentication::RegistrationView
R0F1.2.3	CLIPS::client::viewController::authentication::RegistrationView
R0F1.2.4	CLIPS::client::viewController::authentication::RegistrationView
R0F1.2.5	CLIPS::client::viewController::authentication::RegistrationView

Requisito	Classi
R0F1.2.6	CLIPS::client::data::datamanager::AbstractDataManagerListener CLIPS::client::data::urlrequest::AbstractUrlRequestListener CLIPS::client::viewController::authentication::ConfirmRegistration CLIPS::server::urlrequesthandler::DBHandler CLIPS::client::data::LoggedUser CLIPS::client::data::datamanager::LoginManager CLIPS::server::urlrequesthandler::PasswordChecker CLIPS::server::urlrequesthandler::RegistrationFieldsValidator CLIPS::server::urlrequesthandler::RegistrationHandler CLIPS::client::data::urlrequest::URLRequest CLIPS::server::urlrequesthandler::URLRequestHandler CLIPS::server::urlrequesthandler::URLRequestRouter CLIPS::server::urlrequesthandler::UserDataRequest
R0F1.2.6.1	CLIPS::server::urlrequesthandler::DBHandler CLIPS::client::data::LoggedUser CLIPS::client::data::datamanager::LoginManager CLIPS::client::viewController::authentication::RegistrationActivity CLIPS::server::urlrequesthandler::TokenGenerator CLIPS::server::urlrequesthandler::URLRequestHandler CLIPS::server::urlrequesthandler::URLRequestRouter CLIPS::server::urlrequesthandler::UserDataRequest

Requisito	Classi
R0F1.2.7	CLIPS::client::data::datamanager::AbstractDataManagerListener CLIPS::client::data::urlrequest::AbstractUrlRequestListener CLIPS::client::data::urlrequest::CheckRequest CLIPS::client::data::datamanager::CheckResult CLIPS::client::data::datamanager::LoginManager CLIPS::client::viewController::authentication::RegistrationActivity CLIPS::server::urlrequesthandler::RegistrationFieldsValidator CLIPS::server::urlrequesthandler::RegistrationHandler CLIPS::client::viewController::authentication::RegistrationView CLIPS::client::data::urlrequest::RequestMaker CLIPS::client::data::urlrequest::ServerError CLIPS::client::data::urlrequest::URLDataConstants CLIPS::client::data::urlrequest::URLRequest CLIPS::server::urlrequesthandler::URLRequestHandler CLIPS::server::urlrequesthandler::URLRequestRouter
R0F1.2.7.1	CLIPS::server::urlrequesthandler::EmailChecker
R0F1.2.7.1.1	CLIPS::server::urlrequesthandler::EmailChecker
R0F1.2.7.1.2	CLIPS::server::urlrequesthandler::EmailChecker
R0F1.2.7.2	CLIPS::server::urlrequesthandler::UsernameChecker
R0F1.2.7.2.1	CLIPS::server::urlrequesthandler::UsernameChecker
R0F1.2.7.2.2	CLIPS::server::urlrequesthandler::UsernameChecker
R0F1.2.7.3	CLIPS::server::urlrequesthandler::PasswordChecker
R0F1.2.7.4	CLIPS::server::urlrequesthandler::PasswordChecker

Requisito	Classi
R0F2	CLIPS::client::data::datamanager::AbstractDataManagerListener CLIPS::client::data::urlrequest::AbstractUrlRequestListener CLIPS::client::viewController::authentication::AccountActivity CLIPS::client::viewController::authentication::AccountView CLIPS::server::urlrequesthandler::DBHandler CLIPS::client::data::LoggedUser CLIPS::client::data::datamanager::LoginManager CLIPS::server::urlrequesthandler::LogoutHandler CLIPS::client::data::urlrequest::LogoutRequest CLIPS::client::viewController::utility::MenuActivity CLIPS::client::viewController::utility::MenuView CLIPS::client::data::urlrequest::RequestMaker CLIPS::client::data::urlrequest::ServerError CLIPS::client::data::urlrequest::URLDataConstants CLIPS::client::data::urlrequest::URLRequest CLIPS::server::urlrequesthandler::URLRequestHandler CLIPS::server::urlrequesthandler::URLRequestRouter CLIPS::server::urlrequesthandler::UserDataRequest

Requisito	Classi
R0F3	CLIPS::client::data::/textitAbstractQuiz
	CLIPS::client::data::/textitTest
	CLIPS::client::data::datamanager::AbstractDataManagerListener
	CLIPS::client::data::urlrequest::AbstractUrlRequestListener
	CLIPS::client::data::Beacon
	CLIPS::client::pathprogress::BeaconDiscover
	CLIPS::client::data::datamanager::DataManager
	CLIPS::server::urlrequesthandler::DBHandler
	CLIPS::client::data::datamanager::DBHandlerFIXME
	CLIPS::client::data::GameCollection
	CLIPS::client::data::LinearScoringAlgorithm
	CLIPS::client::data::MultipleChoiceImageQuiz
	CLIPS::client::data::MultipleChoiceTest
	CLIPS::client::data::MultipleChoiceTextQuiz
	CLIPS::client::data::Path
	CLIPS::client::viewController::building::PathActivity
	CLIPS::client::data::datamanager::PathDataRequest
	CLIPS::server::urlrequesthandler::PathProvider
	CLIPS::client::data::urlrequest::PathRequest
	CLIPS::client::viewController::building::PathView
	CLIPS::client::data::Proof
	CLIPS::client::data::Proximity
	CLIPS::client::data::urlrequest::RequestMaker
	CLIPS::client::data::ScoringAlgorithmFactory
	CLIPS::client::data::urlrequest::ServerError
	CLIPS::client::data::Step
	CLIPS::client::data::TestBuilder
	CLIPS::client::data::TrueFalseImageQuiz
	CLIPS::client::data::TrueFalseTest
	CLIPS::client::data::TrueFalseTextQuiz
	CLIPS::client::data::urlrequest::URLDataConstants
	CLIPS::client::data::urlrequest::URLRequest
	CLIPS::server::urlrequesthandler::URLRequestHandler
	CLIPS::server::urlrequesthandler::URLRequestRouter
	CLIPS::server::urlrequesthandler::UserDataRequest
	CLIPS::client::data::Utility

Requisito	Classi
R0F3.1	CLIPS::client::data::/textitAbstractQuiz
	CLIPS::client::data::/textitTest
	CLIPS::client::data::datamanager::AbstractDataManagerListener
	CLIPS::client::data::urlrequest::AbstractUrlRequestListener
	CLIPS::client::data::Beacon
	CLIPS::client::pathprogress::BeaconDiscover
	CLIPS::client::data::datamanager::DataManager
	CLIPS::client::data::datamanager::DataRequestMaker
	CLIPS::server::urlrequesthandler::DBHandler
	CLIPS::client::data::GameCollection
	CLIPS::client::data::LinearScoringAlgorithm
	CLIPS::client::data::MultipleChoiceImageQuiz
	CLIPS::client::data::MultipleChoiceTest
	CLIPS::client::data::MultipleChoiceTextQuiz
	CLIPS::client::data::Path
	CLIPS::client::data::datamanager::PathDataRequest
	CLIPS::client::pathprogress::PathProgressController
	CLIPS::server::urlrequesthandler::PathProvider
	CLIPS::client::data::urlrequest::PathRequest
	CLIPS::client::data::Proof
	CLIPS::client::viewController::games::ProofActivity
	CLIPS::client::data::Proximity
	CLIPS::client::data::urlrequest::RequestMaker
	CLIPS::client::data::ScoringAlgorithmFactory
	CLIPS::client::viewController::games::SearchNewStepActivity
	CLIPS::client::data::urlrequest::ServerError
	CLIPS::client::data::Step
	CLIPS::client::data::TestBuilder
	CLIPS::client::data::TrueFalseImageQuiz
	CLIPS::client::data::TrueFalseTest
	CLIPS::client::data::TrueFalseTextQuiz
	CLIPS::client::data::urlrequest::URLDataConstants
	CLIPS::client::data::urlrequest::URLRequest
	CLIPS::client::data::Utility

Requisito	Classi
R0F3.1.1	CLIPS::client::data::Beacon CLIPS::client::pathprogress::BeaconDiscover CLIPS::client::pathprogress::PathProgressController CLIPS::client::pathprogress::PathProgressControllerDelegate CLIPS::client::viewController::games::SearchNewStepActivity CLIPS::client::viewController::games::SearchNewStepView CLIPS::client::data::Step
R0F3.1.1.1	CLIPS::client::data::Beacon CLIPS::client::pathprogress::BeaconDiscover CLIPS::client::pathprogress::PathProgressController CLIPS::client::pathprogress::PathProgressControllerDelegate CLIPS::client::viewController::games::SearchNewStepActivity CLIPS::client::data::Step
R0F3.1.1.2	CLIPS::client::data::Beacon CLIPS::client::pathprogress::PathProgressController CLIPS::client::pathprogress::PathProgressControllerDelegate CLIPS::client::viewController::games::SearchNewStepActivity CLIPS::client::data::Step
R1F3.1.1.3	CLIPS::client::data::MultipleChoiceImageQuiz CLIPS::client::data::MultipleChoiceTextQuiz CLIPS::client::data::TrueFalseImageQuiz CLIPS::client::data::TrueFalseTextQuiz
R2F3.1.1.4	CLIPS::client::data::Beacon CLIPS::client::data::Proximity
R2F3.1.1.4.1	CLIPS::client::data::Proximity

Requisito	Classi
R0F3.1.2	CLIPS::client::data::/textitAbstractQuiz CLIPS::client::data::/textitTest CLIPS::client::data::GameCollection CLIPS::client::data::LinearScoringAlgorithm CLIPS::client::data::MultipleChoiceImageQuiz CLIPS::client::data::MultipleChoiceTest CLIPS::client::data::MultipleChoiceTextQuiz CLIPS::client::data::Proof CLIPS::client::viewController::games::ProofActivity CLIPS::client::viewController::games::ProofView CLIPS::client::data::Step CLIPS::client::data::TrueFalseImageQuiz CLIPS::client::data::TrueFalseTest CLIPS::client::data::TrueFalseTextQuiz
R0F3.1.2.1	CLIPS::client::data::/textitAbstractQuiz CLIPS::client::data::/textitTest CLIPS::client::data::GameCollection CLIPS::client::data::MultipleChoiceImageQuiz CLIPS::client::data::MultipleChoiceTest CLIPS::client::data::MultipleChoiceTextQuiz CLIPS::client::data::TrueFalseImageQuiz CLIPS::client::data::TrueFalseTest CLIPS::client::data::TrueFalseTextQuiz
R0F3.1.2.2	CLIPS::client::data::Proof CLIPS::client::viewController::games::ProofActivity CLIPS::client::viewController::games::ProofView
R0F3.1.2.3	CLIPS::client::viewController::games::MultipleChoiceQuizActivity CLIPS::client::viewController::games::MultipleChoiceQuizView CLIPS::client::viewController::games::TrueFalseQuizActivity CLIPS::client::viewController::games::TrueFalseQuizView
R0F3.1.2.4	CLIPS::client::data::LinearScoringAlgorithm CLIPS::client::data::ProofResult CLIPS::client::viewController::games::ProofResultActivity CLIPS::client::viewController::games::ProofResultView

Requisito	Classi
R0F3.1.2.5	CLIPS::client::data::Path CLIPS::client::pathprogress::PathProgressController CLIPS::client::data::Proof CLIPS::client::data::Step
R0F3.1.2.5.1	CLIPS::client::data::Path CLIPS::client::pathprogress::PathProgressController CLIPS::client::data::Proof CLIPS::client::viewController::games::SearchNewStepActivity CLIPS::client::viewController::games::SearchNewStepView CLIPS::client::data::Step
R0F3.1.2.5.1.1	CLIPS::client::data::Path CLIPS::client::pathprogress::PathProgressController CLIPS::client::data::Proof CLIPS::client::viewController::games::SearchNewStepActivity CLIPS::client::viewController::games::SearchNewStepView CLIPS::client::data::Step
R0F3.1.2.5.2	CLIPS::client::data::Path CLIPS::client::pathprogress::PathProgressController CLIPS::client::data::Step
R0F3.2	CLIPS::client::data::LinearScoringAlgorithm CLIPS::client::data::PathProgress CLIPS::client::data::PathResult CLIPS::client::data::ProofResult
R0F3.2.1	CLIPS::client::data::PathProgress CLIPS::client::data::PathResult
R0F3.2.1.1	CLIPS::client::data::PathProgress
R0F3.2.2	CLIPS::client::data::PathProgress CLIPS::client::data::PathResult CLIPS::client::data::ProofResult
R0F3.2.2.1	CLIPS::client::data::PathProgress CLIPS::client::data::PathResult CLIPS::client::data::ProofResult

Requisito	Classi
R0F3.2.2.1.1	CLIPS::client::data::PathProgress CLIPS::client::data::ProofResult
R0F3.2.3	CLIPS::client::data::LinearScoringAlgorithm CLIPS::client::data::PathProgress CLIPS::client::data::PathResult CLIPS::client::data::ProofResult
R0F3.2.3.1	CLIPS::client::data::LinearScoringAlgorithm CLIPS::client::data::PathProgress CLIPS::client::data::ProofResult
R0F3.2.3.1.1	CLIPS::client::data::LinearScoringAlgorithm CLIPS::client::data::Proof CLIPS::client::data::ProofResult
R1F3.2.3.2	CLIPS::client::data::datamanager::AbstractDataManagerListener CLIPS::client::data::urlrequest::AbstractUrlRequestListener CLIPS::client::data::datamanager::DataManager CLIPS::client::data::datamanager::DataRequestMaker CLIPS::client::data::datamanager::DBHandlerFIXME CLIPS::client::data::datamanager::GetResultsDataRequest CLIPS::client::data::urlrequest::GetResultsRequest CLIPS::client::data::PathResult CLIPS::client::data::urlrequest::RequestMaker CLIPS::client::data::urlrequest::ServerError CLIPS::client::data::urlrequest::URLDataConstants CLIPS::client::data::urlrequest::URLRequest

Requisito	Classi
R0F3.2.4	CLIPS::client::data::datamanager::AbstractDataManagerListener CLIPS::client::data::urlrequest::AbstractUrlRequestListener CLIPS::client::data::datamanager::DataManager CLIPS::client::data::datamanager::DataRequestMaker CLIPS::server::urlrequesthandler::DBHandler CLIPS::client::data::datamanager::GetRankingDataRequest CLIPS::client::data::urlrequest::GetRankingRequest CLIPS::client::viewController::savedresults::RankingActivity CLIPS::client::viewController::savedresults::RankingView CLIPS::client::data::urlrequest::RequestMaker CLIPS::client::data::Score CLIPS::client::data::urlrequest::ServerError CLIPS::client::data::urlrequest::URLDataConstants CLIPS::client::data::urlrequest::URLRequest CLIPS::server::urlrequesthandler::URLRequestHandler CLIPS::server::urlrequesthandler::URLRequestRouter CLIPS::server::urlrequesthandler::UserDataRequest
R0F3.2.4.1	CLIPS::client::data::datamanager::AbstractDataManagerListener CLIPS::client::data::urlrequest::AbstractUrlRequestListener CLIPS::client::data::datamanager::DataManager CLIPS::client::data::datamanager::DataRequestMaker CLIPS::server::urlrequesthandler::DBHandler CLIPS::client::data::datamanager::GetRankingDataRequest CLIPS::client::data::urlrequest::GetRankingRequest CLIPS::client::data::urlrequest::RequestMaker CLIPS::client::viewController::savedresults::ResultRow CLIPS::client::data::Score CLIPS::client::data::urlrequest::ServerError CLIPS::client::data::urlrequest::URLDataConstants CLIPS::client::data::urlrequest::URLRequest CLIPS::server::urlrequesthandler::URLRequestHandler CLIPS::server::urlrequesthandler::URLRequestRouter CLIPS::server::urlrequesthandler::UserDataRequest
R0F3.2.5	CLIPS::client::data::PathResult CLIPS::client::data::ProofResult

Requisito	Classi
R0F3.2.6	CLIPS::client::data::PathResult CLIPS::client::data::ProofResult
R0F3.3	CLIPS::client::data::datamanager::/textitSaveDataManager CLIPS::client::data::datamanager::AbstractDataManagerListener CLIPS::client::data::urlrequest::AbstractUrlRequestListener CLIPS::client::data::datamanager::DataRequestMaker CLIPS::server::urlrequesthandler::DBHandler CLIPS::client::data::PathResult CLIPS::client::data::ProofResult CLIPS::client::data::urlrequest::RequestMaker CLIPS::client::data::datamanager::SaveResultDataRequest CLIPS::client::data::urlrequest::SaveResultRequest CLIPS::client::data::urlrequest::ServerError CLIPS::client::data::urlrequest::URLDataConstants CLIPS::client::data::urlrequest::URLRequest CLIPS::server::urlrequesthandler::URLRequestHandler CLIPS::server::urlrequesthandler::URLRequestRouter CLIPS::server::urlrequesthandler::UserDataRequest

Requisito	Classi
R0F4	CLIPS::client::data::datamanager::AbstractDataManagerListener CLIPS::client::data::urlrequest::AbstractUrlRequestListener CLIPS::client::data::AppInfo CLIPS::client::viewController::utility::AppInfoActivity CLIPS::client::data::datamanager::AppInfoDataRequest CLIPS::server::urlrequesthandler::AppInfoProvider CLIPS::client::data::urlrequest::AppInfoRequest CLIPS::client::viewController::utility::AppInfoView CLIPS::client::data::datamanager::DataManager CLIPS::client::data::datamanager::DataRequestMaker CLIPS::server::urlrequesthandler::DBHandler CLIPS::client::data::urlrequest::RequestMaker CLIPS::client::data::urlrequest::ServerError CLIPS::client::data::urlrequest::URLDataConstants CLIPS::client::data::urlrequest::URLRequest CLIPS::server::urlrequesthandler::URLRequestHandler CLIPS::server::urlrequesthandler::URLRequestRouter CLIPS::server::urlrequesthandler::UserDataRequest
R0F4.1	CLIPS::client::data::AppInfo CLIPS::client::viewController::utility::AppInfoActivity CLIPS::client::viewController::utility::AppInfoView
R0F4.2	CLIPS::client::data::AppInfo CLIPS::client::viewController::utility::AppInfoActivity CLIPS::client::viewController::utility::AppInfoView
R0F4.3	CLIPS::client::data::AppInfo CLIPS::client::viewController::utility::AppInfoActivity CLIPS::client::viewController::utility::AppInfoView
R0F4.3.1	CLIPS::client::viewController::utility::AppInfoActivity CLIPS::client::viewController::utility::AppInfoView

Requisito	Classi
R0F5	CLIPS::client::data::datamanager::AbstractDataManagerListener
	CLIPS::client::data::urlrequest::AbstractUrlRequestListener
	CLIPS::client::data::datamanager::DataManager
	CLIPS::client::data::datamanager::DataRequestMaker
	CLIPS::server::urlrequesthandler::DBHandler
	CLIPS::client::data::datamanager::DBHandlerFIXME
	CLIPS::client::data::datamanager::GetResultsDataRequest
	CLIPS::client::data::urlrequest::GetResultsRequest
	CLIPS::client::data::PathResult
	CLIPS::server::urlrequesthandler::PathResultsProvider
	CLIPS::client::data::ProofResult
	CLIPS::client::data::urlrequest::RequestMaker
	CLIPS::client::viewController::savedresults::ResultActivity
	CLIPS::client::data::urlrequest::ServerError
	CLIPS::client::data::urlrequest::URLDataConstants
	CLIPS::client::data::urlrequest::URLRequest
	CLIPS::server::urlrequesthandler::URLRequestHandler
	CLIPS::server::urlrequesthandler::URLRequestRouter
	CLIPS::server::urlrequesthandler::UserDataRequest
	CLIPS::server::urlrequesthandler::UserResultProvider
	CLIPS::client::data::Utility

Requisito	Classi
R0F5.1	CLIPS::client::data::datamanager::AbstractDataManagerListener CLIPS::client::data::urlrequest::AbstractUrlRequestListener CLIPS::client::data::datamanager::DataManager CLIPS::client::data::datamanager::DataRequestMaker CLIPS::server::urlrequesthandler::DBHandler CLIPS::client::data::datamanager::DBHandlerFIXME CLIPS::client::data::datamanager::GetResultsDataRequest CLIPS::client::data::urlrequest::GetResultsRequest CLIPS::client::data::PathResult CLIPS::server::urlrequesthandler::PathResultsProvider CLIPS::client::data::ProofResult CLIPS::client::data::urlrequest::RequestMaker CLIPS::client::viewController::savedresults::SavedResultView CLIPS::client::data::urlrequest::ServerError CLIPS::client::data::urlrequest::URLDataConstants CLIPS::client::data::urlrequest::URLRequest CLIPS::server::urlrequesthandler::URLRequestHandler CLIPS::server::urlrequesthandler::URLRequestRouter CLIPS::server::urlrequesthandler::UserDataRequest CLIPS::server::urlrequesthandler::UserResultProvider CLIPS::client::data::Utility
R0F5.1.1	CLIPS::client::data::PathResult CLIPS::server::urlrequesthandler::PathResultsProvider CLIPS::server::urlrequesthandler::RankingProvider CLIPS::client::viewController::savedresults::SavedResultView
R0F5.1.1.1	CLIPS::client::data::PathResult CLIPS::client::viewController::savedresults::SavedResultView
R0F5.1.1.2	CLIPS::client::data::PathResult CLIPS::client::viewController::savedresults::SavedResultView
R0F5.1.1.3	CLIPS::client::data::PathResult CLIPS::client::viewController::savedresults::SavedResultView
R0F5.1.1.4	CLIPS::client::data::PathResult CLIPS::client::viewController::savedresults::SavedResultView

Requisito	Classi
R0F5.1.1.5	CLIPS::client::data::PathResult CLIPS::client::viewController::savedresults::SavedResultView
R0F5.1.1.6	CLIPS::client::data::datamanager::AbstractDataManagerListener CLIPS::client::data::urlrequest::AbstractUrlRequestListener CLIPS::client::data::datamanager::DataManager CLIPS::client::data::datamanager::DataRequestMaker CLIPS::client::data::datamanager::GetRankingDataRequest CLIPS::client::data::urlrequest::GetRankingRequest CLIPS::client::data::urlrequest::RequestMaker CLIPS::client::viewController::savedresults::SavedResultView CLIPS::client::data::Score CLIPS::client::data::urlrequest::ServerError CLIPS::client::data::urlrequest::URLDataConstants CLIPS::client::data::urlrequest::URLRequest
R0F5.2	CLIPS::server::urlrequesthandler::PathResultsProvider CLIPS::client::viewController::savedresults::SavedResultView
R0F5.3	CLIPS::client::viewController::savedresults::ResultView

Requisito	Classi
R0F6	<p>CLIPS::client::data::datamanager::AbstractDataManagerListener</p> <p>CLIPS::client::data::urlrequest::AbstractUrlRequestListener</p> <p>CLIPS::client::data::Building</p> <p>CLIPS::client::data::datamanager::BuildingsDataRequest</p> <p>CLIPS::client::viewController::building::BuildingSearchActivity</p> <p>CLIPS::client::viewController::building::BuildingSearchView</p> <p>CLIPS::server::urlrequesthandler::BuildingsProvider</p> <p>CLIPS::client::data::urlrequest::BuildingsRequest</p> <p>CLIPS::client::data::datamanager::DataManager</p> <p>CLIPS::client::data::datamanager::DataRequestMaker</p> <p>CLIPS::server::urlrequesthandler::DBHandler</p> <p>CLIPS::client::data::datamanager::DBHandlerFIXME</p> <p>CLIPS::client::pathprogress::GPSListener</p> <p>CLIPS::client::pathprogress::GPSManager</p> <p>CLIPS::client::data::PathInfo</p> <p>CLIPS::client::pathprogress::PathProgressMaker</p> <p>CLIPS::client::data::urlrequest::RequestMaker</p> <p>CLIPS::client::data::urlrequest::ServerError</p> <p>CLIPS::client::data::urlrequest::URLDataConstants</p> <p>CLIPS::client::data::urlrequest::URLRequest</p> <p>CLIPS::server::urlrequesthandler::URLRequestHandler</p> <p>CLIPS::server::urlrequesthandler::URLRequestRouter</p> <p>CLIPS::server::urlrequesthandler::UserDataRequest</p> <p>CLIPS::client::data::Utility</p>

Requisito	Classi
R0F6.1	CLIPS::client::data::datamanager::AbstractDataManagerListener CLIPS::client::data::urlrequest::AbstractUrlRequestListener CLIPS::client::data::Building CLIPS::client::data::datamanager::BuildingsDataRequest CLIPS::client::viewController::building::BuildingSearchActivity CLIPS::client::viewController::building::BuildingSearchView CLIPS::client::data::urlrequest::BuildingsRequest CLIPS::client::data::datamanager::DataManager CLIPS::client::data::datamanager::DataRequestMaker CLIPS::client::data::datamanager::DBHandlerFIXME CLIPS::client::pathprogress::GPSListener CLIPS::client::pathprogress::GPSManager CLIPS::client::data::PathInfo CLIPS::client::pathprogress::PathProgressMaker CLIPS::client::data::urlrequest::RequestMaker CLIPS::client::data::urlrequest::ServerError CLIPS::client::data::urlrequest::URLDataConstants CLIPS::client::data::urlrequest::URLRequest CLIPS::client::data::Utility
R0F6.1.1	CLIPS::client::viewController::building::BuildingSearchActivity CLIPS::client::viewController::building::BuildingSearchView CLIPS::client::viewController::building::SearchCriteriaRow
R0F6.1.1.1	CLIPS::client::viewController::building::BuildingRow CLIPS::client::viewController::building::BuildingSearchView CLIPS::server::urlrequesthandler::BuildingsProvider CLIPS::server::urlrequesthandler::DBHandler CLIPS::server::urlrequesthandler::PathProvider CLIPS::server::urlrequesthandler::URLRequestHandler CLIPS::server::urlrequesthandler::URLRequestRouter CLIPS::server::urlrequesthandler::UserDataRequest

Requisito	Classi
R2F6.1.1.2	CLIPS::client::viewController::building::BuildingSearchActivity CLIPS::client::viewController::building::BuildingSearchView CLIPS::server::urlrequesthandler::DBHandler CLIPS::server::urlrequesthandler::URLRequestHandler CLIPS::server::urlrequesthandler::UserDataRequest
R0F6.2	CLIPS::client::viewController::building::BuildingActivity CLIPS::server::urlrequesthandler::BuildingsProvider CLIPS::client::viewController::building::BuildingView CLIPS::client::viewController::building::PathRow
R0F6.2.1	CLIPS::client::data::Building CLIPS::client::viewController::building::BuildingActivity CLIPS::server::urlrequesthandler::BuildingsProvider CLIPS::client::viewController::building::BuildingView
R0F6.2.1.1	CLIPS::client::data::Building CLIPS::client::viewController::building::BuildingActivity CLIPS::server::urlrequesthandler::BuildingsProvider CLIPS::client::viewController::building::BuildingView
R0F6.2.1.2	CLIPS::client::data::Building CLIPS::client::viewController::building::BuildingActivity CLIPS::server::urlrequesthandler::BuildingsProvider CLIPS::client::viewController::building::BuildingView
R0F6.2.1.3	CLIPS::client::data::Building CLIPS::client::viewController::building::BuildingActivity CLIPS::server::urlrequesthandler::BuildingsProvider CLIPS::client::viewController::building::BuildingView
R2F6.2.1.4	CLIPS::client::data::Building CLIPS::client::viewController::building::BuildingActivity CLIPS::client::viewController::building::BuildingView CLIPS::client::data::PathInfo
R0F6.2.2	CLIPS::client::data::Building CLIPS::client::viewController::building::BuildingActivity CLIPS::client::viewController::building::BuildingView

Requisito	Classi
R0F6.2.3	CLIPS::client::data::Building CLIPS::client::viewController::building::BuildingActivity CLIPS::client::viewController::building::BuildingView
R0F6.2.3.1	CLIPS::client::data::Building CLIPS::client::viewController::building::BuildingActivity CLIPS::client::viewController::building::BuildingView
R0F6.2.3.2	CLIPS::client::data::Building CLIPS::client::viewController::building::BuildingActivity CLIPS::client::viewController::building::BuildingView
R0F6.2.3.3	CLIPS::client::data::Building CLIPS::client::viewController::building::BuildingActivity CLIPS::client::viewController::building::BuildingView
R0F6.2.3.4	CLIPS::client::data::Building CLIPS::client::viewController::building::BuildingActivity CLIPS::client::viewController::building::BuildingView
R0F6.2.3.5	CLIPS::client::data::Building CLIPS::client::viewController::building::BuildingActivity CLIPS::client::viewController::building::BuildingView
R0F6.2.3.6	CLIPS::client::data::Building CLIPS::client::viewController::building::BuildingActivity CLIPS::client::viewController::building::BuildingView

Requisito	Classi
R0F7	CLIPS::client::data::datamanager::AbstractDataManagerListener CLIPS::client::data::urlrequest::AbstractUrlRequestListener CLIPS::client::viewController::authentication::ChangeProfileActivity CLIPS::client::data::urlrequest::ChangeProfileDataRequest CLIPS::server::urlrequesthandler::DBHandler CLIPS::server::urlrequesthandler::GetUserData CLIPS::client::data::LoggedInUser CLIPS::client::data::datamanager::LoginManager CLIPS::server::urlrequesthandler::PostUserData CLIPS::client::data::urlrequest::RequestMaker CLIPS::client::data::urlrequest::ServerError CLIPS::client::data::urlrequest::URLDataConstants CLIPS::client::data::urlrequest::URLRequest CLIPS::server::urlrequesthandler::URLRequestHandler CLIPS::server::urlrequesthandler::URLRequestRouter CLIPS::server::urlrequesthandler::UserDataRequest
R0F7.1	CLIPS::client::data::datamanager::AbstractDataManagerListener CLIPS::client::data::urlrequest::AbstractUrlRequestListener CLIPS::client::data::urlrequest::ChangeProfileDataRequest CLIPS::client::data::LoggedInUser CLIPS::client::data::datamanager::LoginManager CLIPS::client::data::urlrequest::RequestMaker CLIPS::client::data::urlrequest::ServerError CLIPS::client::viewController::authentication::UpdateUserInfoActivity CLIPS::client::viewController::authentication::UpdateUserInfoView CLIPS::client::data::urlrequest::URLDataConstants CLIPS::client::data::urlrequest::URLRequest
R0F7.1.1	CLIPS::server::urlrequesthandler::UsernameChecker

Requisito	Classi
R0F7.2	CLIPS::client::data::datamanager::AbstractDataManagerListener CLIPS::client::data::urlrequest::AbstractUrlRequestListener CLIPS::client::viewController::authentication::ChangeProfileActivity CLIPS::client::data::urlrequest::ChangeProfileDataRequest CLIPS::client::data::datamanager::LoginManager CLIPS::client::data::urlrequest::RequestMaker CLIPS::client::data::urlrequest::ServerError CLIPS::client::data::urlrequest::URLDataConstants CLIPS::client::data::urlrequest::URLRequest
R0F7.2.1	CLIPS::client::viewController::authentication::ChangeProfileActivity CLIPS::server::urlrequesthandler::PasswordChecker
R0F7.3	CLIPS::client::data::datamanager::AbstractDataManagerListener CLIPS::client::data::urlrequest::AbstractUrlRequestListener CLIPS::client::viewController::authentication::ChangeProfileActivity CLIPS::client::data::datamanager::LoginManager CLIPS::server::urlrequesthandler::PostUserData CLIPS::client::data::urlrequest::ServerError CLIPS::client::data::urlrequest::URLRequest

Tabella 4: Tracciamento requisiti-classi

10 Algoritmo di assegnazione del punteggio

L'assegnazione del punteggio di ciascuna prova viene fatta dalla classe `LinearScoringAlgorithm`. Questa classe viene configurata all'inizio di una prova, con dei dati forniti dal server in modo che ricevendo in input:

- il numero di risposte corrette,
- il numero di risposte totali,
- la durata della prova

sappia fornire un punteggio appropriato.

Non è necessario che siano per forza risposte a domande testuali: se la prova consiste di un gioco le risposte corrette possono essere il numero di successi e le risposte totali il totale dei tentativi.

La classe `LinearScoringAlgorithm` viene configurata per ciascuna prova con un JSON contenente le seguenti chiavi:

`maxScore` indica il punteggio massimo assegnabile ad ogni prova

`minScore` indica il punteggio minimo assegnato ad ogni prova

`maxTime` indica il tempo massimo in cui completare la prova: se un giocatore impiega più di questo tempo viene penalizzato come se ci avesse messo questo tempo

`minTime` il tempo massimo in cui l'utente può finire la prova ottenendo il massimo punteggio per il tempo

`timeWeight` indica il peso con cui conta il tempo nel computo del punteggio finale

`accuracyWeight` indica il peso con cui conta l'esattezza delle risposte nel computo del punteggio finale

Nello specifico, per l'assegnazione dei punteggi, `LinearScoringAlgorithm` assegna linearmente un punteggio tra `minScore` e `maxScore` alla prova confrontando le risposte corrette con quelle totali, assegna un punteggio tra `minScore` e `maxScore` al tempo in cui viene risolta la prova (dove un tempo di `minTime` o inferiore aggiudica il massimo dei punti, cioè `maxScore`, e un tempo di `maxTime` o superiore aggiudica il minimo dei punti, cioè `minScore`) e infine effettua la media tra i due valori, pesata secondo i pesi `accuracyWeight` e `timeWeight`.

La formula per il conteggio dei punti risulta quindi, definendo:

$$\begin{aligned}\Delta_{score} &= \text{maxScore} - \text{minScore} \\ \text{weight} &= \text{weight}_a + \text{weight}_t \\ \tilde{t} &= \min(\text{time}_{max}, \min(\text{time}_{min}, \text{time}))\end{aligned}$$

dove weight_t e weight_a sono `timeWeight` e `accuracyWeight`, otteniamo la funzione di per l'assegnazione del punteggio come:

$$\text{score} = \frac{\Delta_{score}}{\text{weight}} \cdot \left(\frac{\text{correct}}{\text{total}} \cdot \text{weight}_a + \left(1 - \frac{\tilde{t} - \text{time}_{min}}{\text{time}_{max} - \text{time}_{min}} \right) \cdot \text{weight}_t \right)$$

Ad esempio se si vuole un algoritmo che non tenga in considerazione del tempo impiegato dal giocatore per completare una prova si può creare un `LinearScoringAlgorithm` passando il `timeWeight` = 0. Invece se si vuole che ad ogni prova, vengano assegnati sempre almeno 10 punti, per non demoralizzare gli utenti meno abili sarà sufficiente assegnare un `minScore` di 10.

11 Specifica JSON client-server

In questa sezione definiamo come sono strutturati i JSON utilizzati per la trasmissione di dati tra client e server.

11.1 AppInfo

```

1 {
2   description   : String (testo di descrizione delle funzioni dell'
3     App)
4   supportemail  : String (email per il supporto)
5   websiteURL    : String (url del sito di supporto)
6   discoveryUUID : String (UUID dei beacon usati dall'App)
7 }

```

11.2 Path

id: String (l'id del percorso) algorithm: [Algorithm](#) (l'algoritmo per ...)

```

1 {"menu": {
2   "id": "file",
3   "value": "File",
4   "popup": {
5     "menuitem": [
6       {"value": "New", "onclick": "CreateNewDoc()"},
7       {"value": "Open", "onclick": "OpenDoc()"},
8       {"value": "Close", "onclick": "CloseDoc()"}
9     ]
10  }
11 }}
12 0123456789

```

```

1 {
2   id           : int (l'id del percorso),
3   title        : String (il titolo del percorso),
4   startingMessage : String (il messaggio mostrato all'inizio del
5     percorso),
6   rewardMessage  : String (il messaggio mostrato alla fine del
7     percorso),
8   steps         : [
9     \hyperref[sub:Step]{Step},
10    ...
11  ] (l'array che contiene le tappe del percorso)
12 }

```

```

1 {
2   stopBeacon    : \hyperref[sub:Beacon]{Beacon} (il beacon che
3     segnala la posizione della tappa)
4   proximities  : [
5     \hyperref[sub:Proximity]{Proximity}
6     ...
7   ]
8 }

```

```

6   ] (l'array che contiene gli oggetti in cui sono salvati i dati
    per segnalare all'utente la distanza che manca per
    raggiungere questa stazione)
7   proof      : \hyperref[sub:Proof]{Proof} (la prova che l'utente
    deve affrontare quando raggiunge la stazione)
8   }

```

```

1   {
2     id      : int (l'id del beacon)
3     UUID   : String (la stringa che identifica i beacon dell'
    applicazione o i beacon dell'edificio[U+FFFD]1 primo campo che
    permette di identificare il beacon)
4     Major  : int (il secondo campo che permette di identificare il
    beacon)
5     Minor  : int (il terzo campo che permette di identificare il
    beacon)
6   }

```

```

1   {
2     beacon : \hyperref[sub:Beacon]{Beacon} (il beacon che viene usato
    per comunicare all'utente quant[U+FFFD]icino alla stazione
    cercata)
3     percentage: double (il valore percentuale della distanza
    percorsa, utile per una rappresentazione grafica della
    distanza che manca)
4     textToDisplay: String (il testo da far visualizzare all'
    applicazione per informare l'utente se si sta avvicinando e
    quant[U+FFFD]istante)
5   }

```

11.3 Algorithm

```

1   {
2     minScore      : int
3     maxScore      : int
4
5     minTime       : double
6     maxTime       : double
7
8     timeWeight    : double
9     accuracyWeight: double
10  }
11 per la spiegazione delle chiavi si veda la \hyperref[sec:Algoritmo
    di assegnazione del punteggio]{sezione corrispondente}.

```

11.4 Proof

Le prove possono essere di tipi diversi: il primo sono domande a scelta multipla, con eventuali immagini. *indichiamo le chiavi opzionali con un punto di domanda*

11.4.1 Scelta Multipla

```
1 {
2   testType: "standardQuiz" (indica il tipo di test)
3   testTypeVersion: 0.1 (indica la versione, per compatibilit [U+FFFD]
   con futuri aggiornamenti)
4   data : (dati delle domande) {
5     shuffleAnswers: Bool (se vero il client deve presentare le
       domande in ordine casuale)
6     shuffleQuestions: Bool (se vero il client deve mescolare le
       risposte per ogni domanda)
7     questions : (domande) [
8       {
9         question: String (testo della domanda)
10        answers : [String] (array di risposte tra cui scegliere
            )
11        image   : String? (url dell'immagine da mostrare,
            opzionale)
12        correctIndex : 0 (indice della risposta corretta)
13      },
14      . . .
15    ]
16  }
17 }
```

11.4.2 Vero o Falso

```
1 {
2   testType: "VeroFalso"
3   testTypeVersion: 0.1
4   data : {
5     shuffleQuestions: Bool (vero se le domande vanno mescolate)
6     questions : [
7       {
8         question: String (domanda)
9         correctAnswer: Bool (risposta)
10      },
11      . . .
12    ]
13  }
14 }
```