Applicativo Mobile - Diagramma delle Classi di Design

StrutturaDAO
- context: Context
+ StrutturaDAO(context: Context)
+ getStrutturaPerId(idStruttura: int, callback: VolleyCallback)
+ getStrutturePerFiltri(filtriStruttura: Filtri, callback: VolleyCallback)
+ getRatingById(int idStruttura, final VolleyCallback<Float> callback)
com.google.android.gms
com.mapbox.mapboxsdk
CustomSupportMapFragment
- toolbar: Toolbar
+ CustomSupportMapFragment(toolbar: Toolbar)
MainActivity
- fusedLocationClient: FusedLocationProviderClient
- drawerLayout: DrawerLayout
~ toolbar: Toolbar
# yourPositionButton: FloatingActionButton
- navigationView: NavigationView
- menu: Menu
# instanceState: Bundle
- mapboxMap: com.mapbox.mapboxsdk.maps.MapboxMap
- localStyle: com.mapbox.mapboxsdk.maps.Style
+ latitudine: double
+ longitudine: double
~ initApiClient()
~ showLocationSettingsDialog()
- setUpBackPressed()
- setupButtons()
- moveCamera(latitude: double, longitude: double)
- setupYourPositionButton()
- setupFiltriButton()
- loadFiltriFragment()
- logout()
- configuraDrawer()
+ tornaAllaMainActivity()
- aggiornaDrawer()
- utenteLoggato(): boolean
- caricaRegistrazioneFragment()
- caricaLoginFragment()
- setYourPositionButtonVisible()
- setYourPositionButtonInvisible()
- controllaPermessiGPS()
- creaMapFragment(options: com.m-apbox.mapboxsdk.maps.MapboxMapOptions)
- getIdStrutturaFromFeature(feature: com.mapbox.geojson.Feature): int
# setMap()
<<xml>>
activity_main
<<xml>>
fragment_lista_strutture
FiltriFragment
- backButton: ImageButton
- container: ViewGroup
- cercaButton: Button
- nomeText: EditText
- cittaText: EditText
- distanzaText: EditText
- spinnerCategoria: Spinner
- spinnerPrezzo: Spinner
- spinnerValutazione: Spinner
- prossimitaSwitch: Switch
- latitudine: double
- longitudine: double
- fusedLocationClient: FusedLocationProviderClient
- setPosizione()
- initApiClient()
- showLocationSettingsDialog()
+ onActivityResult(requestCode: int, resultCode: int, data: Intent)
- initSpinnerCategoria(view: View, adapter: ArrayAdapter)
- initSpinnerPrezzo(view: View, adapter: ArrayAdapter)
- initSpinnerValutazione(view: View, adapter: ArrayAdapter)
- initSpinners(view: View, adapter: ArrayAdapter)
- initGUIElements(view: View)
- hasFineLocationAccess(): boolean
- hasGPSPermissions(): boolean
- configuraProssimitàSwitch()
- askForGPSPermissions()
~ disabilitaProssimita()
~ abilitaProssimita()
- cercaStrutture()
- nessunaStrutturaTrovata()
<<xml>>
fragment_connessione_assente
ConnessioneAssenteFragment
- tastoIndietro: ImageButton
NessunaStrutturaTrovataFragment
- tastoIndietro: ImageButton
<<xml>>
fragment_nessuna_struttura
<<xml>>
fragment_filtri
Filtri
- nome: String
- citta: String
- categoria: String
- prezzo: String
- valutazione: String
- distanzaMassima: String
- prossimita: boolean
+ getFiltriNonVuoti(longitudine: double, latitudine, double): String
ListaStruttureFragment
- backButton: ImageButton
~ listaStrutture: ArrayList<Struttura>
+ ListaStruttureFragment(strutture: ArrayList)
+ newInstance(strutture: ArrayList): Fragment
ListaStruttureRecyclerViewAdapter
- appActivity: AppCompatActivity
+ ListaStruttureRecyclerViewAdapter(listaStrutture: ArrayList<Struttura>, activity: MainActivity)
+ getItemCount(): int
+ onBindViewHolder(ViewHolder holder, final int position)
de.hdodenhof.circleimageview.CircleImageView
conn.squareup.picasso.Picasso
<<interface>>
VolleyCallback<T>
+ onSuccess(result: T)
+ onFail()
com.android.volley
RecensioneDAO
- context: Context
+ RecensioneDAO(context: Context)
+ getRecensionePerIdStruttura(idStruttura: int, callback: VolleyCallback<JSONArray>
+ aggiungiRecensione(nuovaRecensione: Recensione, callback: VolleyCallback<String>
PasswordUtils
- RANDOM: Random
- ALPHABET: String
- ITERATIONS: int
- KEY_LENGTH: int
+ getSalt(int length): String
+ hash(char[] password, byte[] salt): byte[]
+ generateSecurePassword(String password, String salt): String
+verifyUserPassword(String providedPassword, String securedPassword, String salt): boolean
UtenteDAO
- context: Context
+ UtenteDAO(context: Context)
- aggiungiUsernameAllaQuery(username: String): String
- controllaPassword(utenteJson: JSONObject, givenPassword: String): boolean
+ effettuaLogin(username: String, givenPassword: String, callback: VolleyCallback<Boolean>
- costruisciQueryInserimento(utente: Utente): String
+ registraUtente(utente: Utente, callback: VolleyCallback<String>)
+ isAccountEsistente(String username, final VolleyCallback<Boolean> callback)
<<xml>>
fragment_struttura
DettagliStrutturaFragment
- tastoIndietro: ImageButton
- immagineStruttura: de.hdodenhf.circleImageView.CircleImageView
- ratingBarStruttura: RatingBar
- nomeStruttura: TextView
- descrizioneStruttura: TextView
- cittaStruttura: TextView
- indirizzoStruttura: TextView
- prezzoStruttura: TextView
- categoriaStruttura: TextView
- valutazioneRecensione: TextView
- numeroRecensioni: TextView
- fabButton: FloatingActionButton
- fabVisualizzaMappa: FloatingActionButton
- fabAggiungiRecensione: FloatingActionButton
- resetButton: ImageButton
- listaRecensioni: ArrayList<Recensione>
- recyclerView: RecyclerView
- FABAperta: boolean
+ DettagliStrutturaFragment(struttura: Struttura)
- creaUtenteDaQuery(result: JSONObject): Utente
- caricaRecensioniStruttura()
+ newInstance(struttura: Struttura): Fragment
- referenziaElementiUI(view: View)
- configuraFloatingActionButton()
- utenteLoggato(): boolean
- aggiungiListenerTastoIndietro()
- mostraDettagliStruttura()
- controllaNomeONicknameFiltro(recensione: Recensione, autoreFIltro: String): boolean
- controllaRatingFiltro(recensione: Recensione, ratingFiltro: int): boolean
- ordinaListaFiltrata(listaFiltrata: ArrayList<Struttura>, ordinaPerData: boolean)
- filtraListaRecensioni(filtri: Bundle): ArrayList<Recensione>
- initRecyclerViewConFiltri(recensioniFiltrate: ArrayLIst<Recensione>)
<<xml>>
fragment_aggiungi_recensione
AggiungiRecensioneFragment
- idStruttura: int
- titoloRecensione: TextView
- testoRecensione: TextView
- ratingBar: RatingBar
- aggiungiRecensioneButton: Button
- backButton: ImageButton
+ AggiungiRecensioneFragment(idStruttura: int)
+ newInstance(idStruttura: int): AggiungiRecensioneFragment
- referenziaElementiUI(view: View)
- controllaCampiNonVuoti(): boolean
- costruisciRecensioneDaInserire: Recensione
- tornaIndietroStruttura()
- getNickname(): String
<<xml>>
fragment_login
LoginFragment
- bottoneIndietro: ImageButton
- bottoneLogin: Button
- usernameLoginText: EditText
- passwordLoginText: EditText
- registratiTextLink: TextView
- progressBar: ProgressBar
- view: View
- referenziaElementiGUI(view: View)
- setupBackBtton(savedInstanceState: Bundle)
- checkCredenziali()
- updateNickname(nickname: String)
- changeUserStatus(newStatus: boolean)
RegistrazioneFragment
- tastoIndietroRegistrazione: ImageButton- dataDiNascita
- dataDiNascita: EditText
- confermaRegistrazioneButton: Button
- emailEditText: EditText
- passwordEditText: EditText
- nomeEditText: EditText
- cognomeEditText: EditText
- nicknameEditText: EditText
- mostraNicknameCheckBox: CheckBox
- referenziaElementiGUI(view: View)
- controllaCampiNonVuoti(): boolean
- controllaLunghezzaPassword(): boolean
- checkCampiNonVuoti(): boolean
- checkLunghezzaPassword(): boolean
- creaUtenteDaInserire(): Utente
- impostaTastoIndietroRegistrazione(savedInstanceState: Bundle)
<<xml>>
fragment_signup
ListaRecensioniRecyclerViewAdapter
+ ListaRecensioniRecyclerViewAdapter(listaRecensioni: ArrayList
VisualizzaRecensioneFragment
- autore: TextView
- data: TextView
- titolo: TextView
- testo: TextView
- ratingBar: RatingBar
- tastoIndietro: ImageButton
+ VisualizzaRecensioneFragment(recensione: Recensione)
+ newInstance(recensione: Recensione): VisualizzaRecensioneFragment
- referenziaElementiGUI(imetView: View)
- aggiungiListenerTastoIndietro()
- riempiCampiRecensione()
<<xml>>
fragment_visualizza_recensione
FiltriRecensioneDialog
- applicaFiltriButton: Button
- autoreText: EditText
- ratingRecensioni: RatingBar
- recentiSwitch: Switch
- view: View
- setDimensioni()
- referenziaElementiGUI()
<<xml>>
filtri_recensione_dialog
RecensioneComparator
+ COMPARATORBYDATA: Comparator<Recensione>
+ COMPARATORBYRATING: Comparator<Recensione>
<<xml>>
map_dialog
VisualizzaSuMappaDialog
+ VisualizzaSuMappaDialog(struttura: Struttura)
- setDimensioni()
- setMap()
VisualizzaSuMappaDialog uti
Struttura
- indirizzo: String
- nome: String
- descrizione: String
- città: String
- fasciaDiPrezzo: String
- categoria: String
- urlFoto: String
- idStruttura: int
- valutazioneMedia: double
- longitudine: double
- latitudine: double
/* Metodi getter e setter (?) */
Utente
- nome: String
- nickname: String
- email: String
- dataDiNascita: String
- password: String
- salt: String
- mostraNickname: boolean
/* Getter e Setter (?) */
Recensione
- testo: String
- dataRecensione: String
- titolo: String
- valutazione: int
- idRecensione: int
- statoRecensione: String
+ getDataInFormatoData(): Date
contiene
*
è scritta
1
si riferisce a
1
<<use>>

Viewer does not support full SVG 1.1

Applicativo Mobile - Diagramma delle Classi di Design