

**Documentazione per il caso di studio di**

**Sviluppo di Mobile Software**

**(SMS)**

**Applicazione per la Promozione Turistica**

Sommario

[Descrizione del Gruppo 3](#_Toc290019)

[Nome del Gruppo 3](#_Toc290020)

[Componenti 3](#_Toc290021)

[Introduzione al Progetto 4](#_Toc290022)

[Nome del Progetto 4](#_Toc290023)

[Descrizione 4](#_Toc290024)

[Logo 4](#_Toc290025)

[Funzionalità 5](#_Toc290026)

[Consultazione Meteo 5](#_Toc290027)

[Consultazione Attrazioni 5](#_Toc290028)

[Consultazione Locali Gastronomici 5](#_Toc290029)

[Consultazione Strutture per il Pernottamento 5](#_Toc290030)

[Consultazione dei principali luoghi di interesse Vicino Bari 5](#_Toc290031)

[Consultazione Degli Eventi nella Provincia di Bari 5](#_Toc290032)

[Informazioni Utili 5](#_Toc290033)

[Consultazione dei Coupon Disponibili 5](#_Toc290034)

[Requisiti per l’Utilizzo 5](#_Toc290035)

# Descrizione del Gruppo

## **Nome del Gruppo**

4Designers

## Componenti

|  |  |  |  |
| --- | --- | --- | --- |
| **Cognome** | **Nome** | **Matricola** | **Email** |
| Castellano | Graziano | 559926 | grazianocas90@gmail.com |
| Di Pierro | Davide | 661292 | ddipierro745@gmail.com |
| Gigante | Domenico | 661582 | dom.gigante0@gmail.com |
| Lisco | Federica | 664323 | fedelis1997@gmail.com |



# Introduzione al Progetto

## Nome del Progetto

Barintondo

## Descrizione

Barintondo è un’applicazione nata con uno scopo preciso: favorire il turismo nella città di Bari.

A tal punto, l’applicazione dovrà offrire un insieme di funzionalità che accompagni e supporti il turista in tutte le fasi di esplorazione: dalla ricerca dei punti d’interesse fino alla prenotazione di locali gastronomici, passando per la consultazione del meteo e l’utilizzo di coupon in determinati luoghi di interesse.

## Logo



# Funzionalità

## Consultazione Meteo

Barintondo permette di consultare in maniera rapida le informazioni principali relative al meteo di Bari in tempo reale.

In particolare, le informazioni mostrate sono:

* Temperatura
* Descrizione sintetica del meteo
* Immagine rappresentativa del meteo

Inoltre, è possibile consultare informazioni più dettagliate cliccando sull’apposito pulsante, il quale aprirà una pagina web con numerose informazioni aggiuntive (previsioni giornaliere e dei giorni a venire).

### Scelte Implementative

Le informazioni relative al meteo sono mostrate all’interno di un LinearLayout, composto da una “ImageView” per l’immagine rappresentativa e più textView per mostrare le singole informazioni. Il pulsante per accedere alle informazioni aggiuntive è un “Chip” di tipo “Action”.

Per ottenere le informazioni in tempo reale, si è deciso di utilizzare le API gratuite del sito OpenWeatherMap, per i seguenti punti di forza:

* Informazioni fornite gratuitamente per un massimo di 60 richieste al secondo, al contrario di altre API disponibili pubblicamente ma con notevoli limiti (60 richieste/ora, informazioni troppo limitate, …)
* Supporto alle informazioni multilingua
* Mette a disposizione una classe Java di nome OperWeatherMapHelper, utilizzabile tramite opportune dipendenze nel file build.gradle, che automatizza le richieste al server ed effettua automaticamente il parsing dei risultati, lasciando al programmatore solo l’onere di selezionare le informazioni desiderate
  + Tutte le richieste sono effettuate automaticamente in maniera asincrona

Le informazioni aggiuntive sono consultabili in una WebView, a cui si accede cliccando sull’apposito “Chip” all’interno del suddetto LinearLayout.

## Consultazione Attrazioni

Attraverso Barintondo è possibile consultare una lista aggiornata di tutti i punti di interesse presenti nella città di Bari tramite il pulsante “Attrazioni” nella schermata principale.

È possibile filtrare gli elementi della lista in base alla particolare categoria di appartenenza:

* Chiese
* Monumenti
* Teatri
* Musei
* Discoteche
* Lidi Balneari
* Luoghi adatti a Famiglie

Barintondo è anche in grado di adattarsi alle preferenze dell’utente: qualora il suo utilizzatore dovesse scegliere più frequentemente una determinata categoria di luoghi, Barintondo offrirà la possibilità di spostare tale categoria all’inizio della lista, così da renderne più agevole l’individuazione.

Inoltre, è possibile ricercare un particolare luogo all’interno della lista, indicandone (in parte o completamente) il nome.

Ogni elemento della lista mostrerà delle informazioni sintetiche per ciascun punto di interesse:

* Nome
* Valutazione, sulla base delle recensioni precedentemente aggiunte dai turisti
* Categoria
* Informazioni sull’apertura (ove possibile)

Cliccando su di un elemento della lista, sarà possibile accedere alla relativa schermata di dettaglio, all’interno della quale possiamo consultare numerose informazioni sul luogo scelto:

* Indirizzo
* Indicazioni stradali su come raggiungerlo (tramite apposito pulsante)
* Altri punti di interesse presenti nei paraggi (sia luoghi turistici che locali gastronomici e strutture per il pernottamento)
* Recensioni sul luogo; è anche possibile inserire una propria recensione

All’interno di questa schermata è anche possibile aggiungere il luogo mostrato alla lista dei propri interessi, a cui si potrà accedere tramite opportuna sezione all’interno del menu laterale.

### Scelte Implementative

Per realizzare la UI relativa alla schermata contenente la lista dei punti di interesse sono stati utilizzati i seguenti componenti:

* Toolbar, contenente una “Action” di tipo “SearchView”, sempre visibile e che permette all’utente di cercare un determinato elemento nella lista in base a nome o categoria. Il meccanismo di ricerca è implementato tramite i metodi “getFilter()” e “publishResults()” nella classe **LuogoAdapter.java** del package **utils**. Tale classe implementa l’interfaccia “Filterable”, che richiede una propria implementazione dei suddetti metodi.
* LinearLayout contenente un ChipGroup, con la logica di selezione mutuamente esclusiva (al più un solo chip attivo in ogni istante). Ogni Chip è di tipo “Action” e gestisce la selezione tramite un’opportuna implementazione dell’interfaccia onClickListener.

Il meccanismo di personalizzazione dell’ordine delle categorie è implementato **CONTINUA DAVIDE**

* La lista è implementata tramite l’utilizzo di una “Recycler View”, con le relative componenti:
  + **Luogo.java**: classe che descrive gli elementi del modello
  + **LuogoAdapter.java**: classe che rappresenta l’adapter utilizzato dalla “Recycler View”. Tale classe ottiene i dati dal server remoto tramite l’utilizzo dei metodi della classe **ControllerRemoteDB.java**, la quale contiene i metodi in cui è implementata la logica di comunicazione con il DB Remoto tramite framework “Volley”.
  + **MyViewHolder** (classe interna di **LuogoAdapter.java**): implementazione del componente ViewHolder, utilizzato dalla Recycler View
* Per gli elementi della “Recycler View” è stato definito un apposito layout, descritto nel file **luogo\_list\_content.xml**. Tale layout prevede una “ImageView” per mostrare un’immagine rimpicciolita del luogo, un particolare layout per mostrare il voto complessivo del luogo **CONTINUA GRAZIANO** ed un insieme di “TextView” per mostrare le informazioni del luogo

Per realizzare la UI relativa alla schermata contenente il dettaglio di un punto di interesse sono stati utilizzati i seguenti componenti:

* “FAB” per l’aggiunta di un luogo alla propria lista di interessi; implementata tramite apposita implementazione dell’interfaccia onClickListener del “FAB”. Il salvataggio dell’interesse nel DB remoto è effettuato tramite opportuno metodo della classe **ControllerRemoteDB.java**
* Il layout precedentemente descritto per mostrare il voto complessivo del luogo
* “ImageView” per mostrare una foto del luogo selezionato
* Tre “Button”, cliccando i quali verrà mostrato un particolare fragment, ciascuno dei quali contiene un particolare sottoinsieme di informazioni:
  + Il primo (partendo da sinistra), selezionato di default, contenente informazioni generiche sul luogo
  + Il secondo, che mostrerà una lista di tutti i luoghi entro il raggio di 3km dal luogo selezionato. La lista non è altro che una “Recycler View” che utilizza gli stessi componenti sopra descritti; unica differenza è il layout di riga: sebbene sia sempre lo stesso, in questa “Recycler View” viene mostrata anche una icona a destra, che mostra la categoria di appartenenza del luogo (Attrazione, Mangiare o Dormire”)
  + Il terzo, che mostrerà le recensioni relative al luogo selezionato; anche in questo caso sarà utilizzata una “Recycler View”, supportata da **Review.java**, **ReviewAdapter.java** e relativa classe interna **MyViewHolder**. Il layout di riga utilizzato è descritto nel file **recensione\_list\_content.xml**

In basso è presente un “FAB” che, quando cliccato, sostituirà la lista delle recensioni con **CONTINUA GRAZIANO**

## Consultazione Locali Gastronomici

Così come le attrazioni, Barintondo offre la possibilità di consultare un elenco di tutti i locali gastronomici di Bari. Il meccanismo di accesso è sempre lo stesso: basterà cliccare sul pulsante “Mangiare” nella schermata principale.

L’implementazione di tale funzionalità segue la stessa logica di quella spiegata a proposito delle attrazi

## Consultazione Strutture per il Pernottamento

## Consultazione dei principali luoghi di interesse Vicino Bari

Mostrati in una lista e tramite lo slider nella home

## Consultazione Degli Eventi nella Provincia di Bari

## Informazioni Utili

## Creazione e consultazione di luoghi preferiti

## Consultazione dei Coupon Utilizzabili

Consultazione lista e dettaglio, utilizzo tramite BT

## Requisiti per l’Utilizzo

Registrazione con dati sintetici, login, logout

# Scelte Implementative Generali

Classe per parsing immagini, volley per le richieste al server, listener per gestire gli eventi in seguito a richieste al server, nav drawer, componenti material design, invio feedback tramite intent ad applicazione di posta, modalità offline, un utente per volta