

# *Progetto di metà corso*

## *Replica del layout della pagina avada/music*

Il progetto è realizzato in Vite - Vue.js3, con l'ausilio delle icone FontAwesome e del Google Font "ROBOTO". Per la parte stilistica si è fatto ricorso a SASS/SCSS.

Relativamente al trasferimento di dati tra i vari componenti del progetto (8 componenti + il componente principale "APP") sono stati preferiti props e \$emit allo store che è stato comunque utilizzato ma al cui interno trova posto solo una variabile.

Nonostante l'intenzione iniziale di strutturare il programma in modo da consentire alle varie selezioni (pulsanti, icone, voci di menù) di dar luogo ad azioni concrete di indirizzamento interno ed esterno ciò non è stato possibile per questioni temporali; se ne rimanda dunque l'eventuale implementazione ad un upgrade futuro del progetto stesso.

Tutto il layout del sito di riferimento è stato riprodotto piuttosto fedelmente, incluse le transizioni al passaggio del mouse sui vari elementi e l'interazione con il menù delle prenotazioni (sempre da un punto di vista del solo layout).

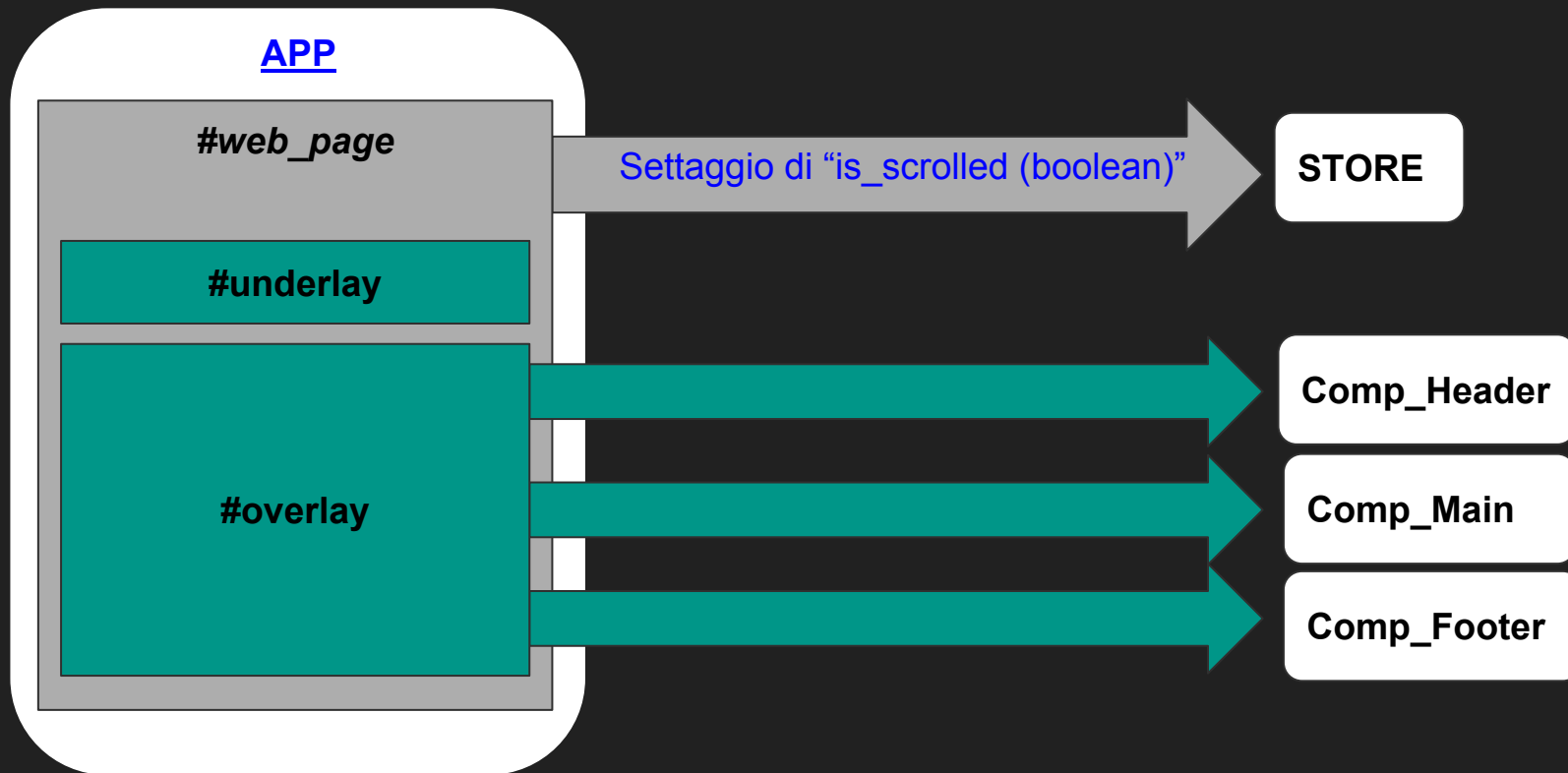
# App.vue

In **App** viene strutturata semanticamente la pagina, con il ricorso ai componenti principali.

La chiamata ai componenti ha luogo all'interno dell'id “**#overlay**”, il quale si contrappone all'id “**#underlay**”, dentro cui è presente l'immagine di sfondo sulla quale scorre tutta la pagina. Per il posizionamento dell'immagine in underlay non si è fatto ricorso all'attributo css “background-attachment”, bensì ad un “position:fixed”, cui si contrappone un “position:relative” dei componenti principali.

In **App**, un “addEventListener(‘scroll’...)”, posizionato in mounted(), ha lo scopo di monitorare lo scroll verticale dell'intera pagina e di settare una specifica variabile all'interno dello “**store**” laddove la pagina sia a scroll zero.

# Struttura interna e chiamata ai componenti:



# Comp\_Header.vue

Il componente si struttura nel solo tag semantico `<header>` al cui interno sono posizionati il logo ed il tag semantico `<nav>`, dentro il quale è presente l'icona del menù principale (compresso).

Nel progetto non si è dato seguito alla dinamizzazione del menù, per il quale si potrebbe sviluppare il classico switch di espansione (al click sull'icona, il menù si espande in tutti i suoi elementi) e di ricompansione (al click su una specifica icona del menù espanso si rientra nella variante con menù compresso).

Il componente, mediante lettura del valore della variabile booleana `is_scrolled` presente nello “`store`”, modifica (nel caso) altezza e background dello `<header>`.

Il posizionamento sticky dello `<header>`, unitamente all'attributo `z-index:999` assicura allo stesso una ubicazione stabile ed un “primo piano” costante, durante l'intera sessione.

Per la visualizzazione dell'icona del menù compresso, viene chiamato il componente `Comp_Menu_Manager`, il quale riceve una serie di valori specifici per il settaggio dei propri “props”; tali valori vengono estrapolati dalla struttura dati `nav_icon{}`. In un eventuale sviluppo futuro dell'espansione del menù, il componente chiamato potrebbe restituire al chiamante un feedback di avvenuto click mediante un `$emit`.

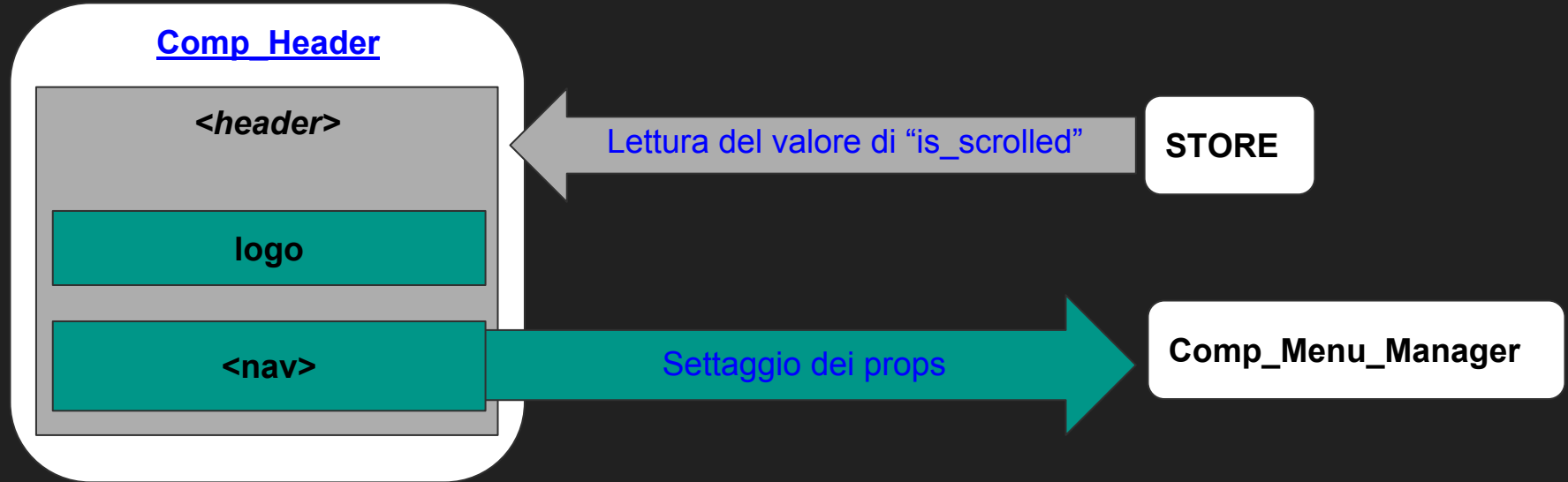
# *Dati utilizzati in Comp\_Header*

- `nav_icon : {}` (oggetto)

consta delle seguenti coppie chiave / valore:

- `menu_class : 0` (servirà al componente specifico per settare la classe di stile)
- `is_horizontal : true` (stabilisce che il menù è a sviluppo orizzontale - parametro che verrebbe utilizzato esclusivamente in un'eventuale espansione del menù)
- `category : 0` (il valore "0" identifica tutti gli elementi di tipo icona)
- `menu_items : []` (array con un unico oggetto, le cui coppie chiave / valore sono ....)
  - `text` (classi font awesome per l'icona specifica)
  - `is_active` (identifica la voce di menù in quanto attiva o meno)

# Struttura interna e chiamata ai componenti:



# Comp\_Main.vue - pagina 1

Il componente si struttura all'interno del tag semantico `<main>`.

Si tratta del componente fulcro dell'applicazione, da cui si diramano le chiamate ai vari sotto-componenti e la distribuzione dei dati

Le sezioni interne al main organizzano e potrebbero gestire (in un eventuale sviluppo futuro dell'applicazione) tutto il layout ed il flusso di informazioni interno al progetto.

Dettaglio delle sezioni:

- `#heading_section`, in cui sono posizionati del testo (tra cui il titolo `<h1>`) ed un piccolo menù costituito da due pulsanti. Il menù con i pulsanti viene posizionato (ed eventualmente gestito in sviluppi futuri) mediante chiamata al componente `Comp_Menu_Manager`.
- `#card_set_section`, in cui sono visualizzate sei cards (immagine + testo), tali che, al passaggio del mouse presentino degli effetti di transizione. Le cards sono state posizionate in griglia e, in un eventuale upgrade, potrebbero, al click, indirizzare verso altre pagine. Ciascuna card viene posizionata singolarmente mediante chiamata al sotto-componente `Comp_Single_Card`.

# Comp\_Main.vue - pagina 2

- `#poster_section`. In questa sezione è presente un'immagine di background, cui sono sovrapposti un paragrafo ed un pulsante "play" (non attivo).
- `#live_dates_section` in cui è presente un menù a sviluppo verticale con dei pulsanti che, al click, mostrano, affiancati, una mappa (statica), del testo, con annesso pulsante di prenotazione (non attivo) ed una foto del gruppo musicale cui fa riferimento la voce del menù selezionata.
- Le varie sezioni sono, quasi sempre, separate da un componente `Comp_Banner` che, a seconda dei casi, può consistere in un testo opportunamente formattato o in un pulsante cliccabile (non attivo) a tutta larghezza.

Le sezioni scambiano informazioni e dati con i sotto-componenti mediante settaggio degli opportuni props o ricezione di eventi di tipo `@emit`.

Segue il dettaglio...



## Interazioni di #heading\_section

La sezione chiama il sotto-componente “Comp\_Menu\_Manager” e setta i relativi props passando i dati presenti nella struttura “heading\_menu”...

- heading\_menu : {} (oggetto)

consta delle seguenti coppie chiave / valore:

- menu\_class : 10 (servirà al componente specifico per settare la classe di stile)
- is\_horizontal : true (stabilisce che il menù è a sviluppo orizzontale)
- category : 1 (il valore “1” identifica i pulsanti con testo)
- menu\_items : [] (array con due oggetti, le cui coppie chiave / valore sono ....)
  - text (testo sul pulsante)
  - is\_active (identifica la voce di menù in quanto attiva o meno)

## Interazioni di #card\_set\_section - pagina 1

La sezione chiama due sotto-componenti: “Comp\_Banner” e “Comp\_Single\_Card” e setta i relativi props passando i dati presenti nelle strutture “banner\_array” e “cards\_array”...

- banner\_array : [] (array di oggetti)

le coppie chiave / valore degli elementi di banner\_array sono:

- id (valore numerico utilizzato per il \$emit solo nel caso di banner “pulsante cliccabile”)
- bg\_color (stabilisce il colore di sfondo del banner)
- bar\_color (colore della barra presente in alto al centro del banner (visibile solo nel primo banner))
- line\_bool (booleano che identifica la presenza di una sottile linea (presente nei banner “solo testo”))
- title (stringa con funzione di titolo nel banner “solo testo” o di testo sul banner “pulsante”)
- text (testo del banner non cliccabile)
- clickable (booleano che identifica il tipo di banner (solo testo o pulsante cliccabile))

## *Interazioni di #card\_set\_section - pagina 2*

- `cards_array` : [] (array di oggetti)

le coppie chiave / valore degli elementi di `cards_array` sono:

- `image` (url dell'immagine della card)
- `title` (stringa che identifica il titolo della card)
- `text` (testo secondario della card)

## *Interazioni di #poster\_section*

La sezione chiama il sotto-componente “**Comp\_Menu\_Manager**” e setta i relativi props passando i dati presenti nella struttura “poster\_menu”

L’oggetto poster\_menu non differisce dagli altri oggetti usati per settare i props di **Comp\_Menu\_Manager**, se non nel valore di alcune delle chiavi, quali **menu\_class**, **category** e **menu\_items.text**

## Interazioni di #live\_dates\_section - pagina 1

Analogamente a #card\_set\_section, anche #live\_dates\_section chiama il sotto-componente Comp\_Banner cui passa i valori di banner\_array mediante settaggio dei props, dopodichè chiama anche il componente Comp\_Menu\_Manager per generare un menù a sviluppo verticale con una complessità maggiore relativamente agli altri menù del progetto; per tale finalità, i props del componente chiamato vengono settati con i valori dell'oggetto live\_dates\_menu

- live\_dates\_menu : {} (oggetto)

le coppie chiave / valore sono le medesime (salvo i valori) degli altri oggetti usati su Comp\_Menu\_Manager, tranne che per gli elementi di .menu\_items e sono:

- text (nome della band musicale)
- dates (stringa con data e località del concerto)
- is\_active (booleano che identifica la voce di menù attiva al momento)
- map\_details (url della mappa (statica) online)
- img\_details (url della foto (su disco) della band)
- sub\_title (stringa che apre la parte testuale (in funzione di titolo) tra mappa e immagine)
- paragraph (paragrafo visualizzato nella parte testuale)

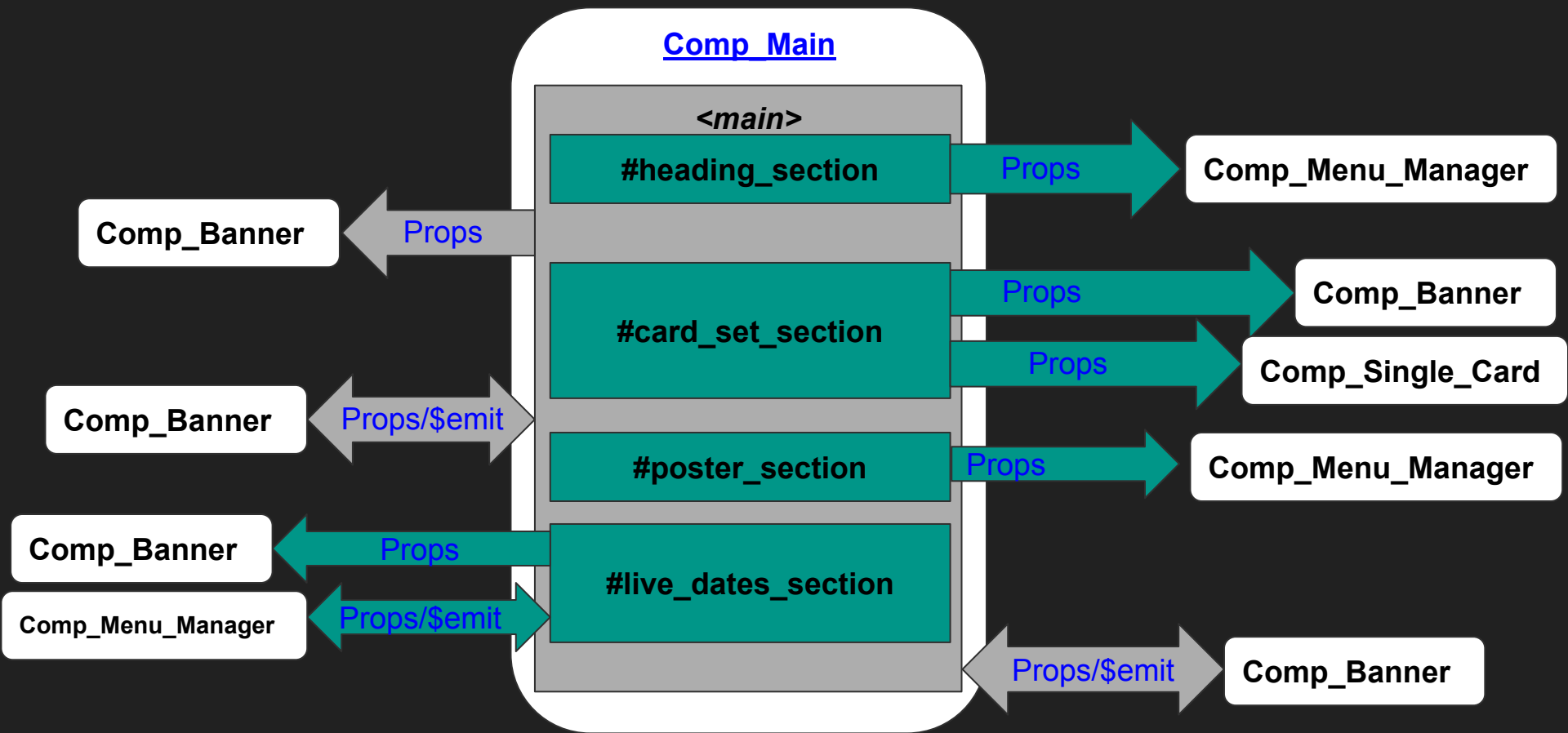
## Interazioni di #live\_dates\_section - pagina 2

La comunicazione tra #live\_dates\_section ed il sotto-componente Comp\_Menu\_Manager, oltre al settaggio dei props del chiamato, si compone anche di un feedback dallo stesso, sotto forma di \$emit.

Il feedback ha origine nel sotto-componente Comp\_Special\_Item, utilizzato da Comp\_Menu\_Manager per generare i pulsanti (voci di menù) della sola sezione in oggetto (#live\_dates\_section), essendo essi strutturalmente di un livello di complessità leggermente superiore agli altri menù del progetto.

In Comp\_Special\_Item viene generato il `this.$emit("click_on_special", this.cs_index)` al click ed esso viene acquisito in Comp\_Menu\_Manager mediante l'assegnazione dinamica `@click_on_special = "rebound_click"`, per poi essere inviato al livello superiore (Comp\_Main / #live\_dates\_section) dalla "funzione di rimbalzo" rebound\_click mediante l'ulteriore `$emit("rebounded", index_to_rebound)`. Al termine del percorso del feedback, sarà compito del metodo click\_from\_special(special\_index) rendere attivo l'elemento cliccato, disattivando tutti gli altri (poichè in questo menù non è possibile avere più elementi attivi in contemporanea).

# Struttura interna e chiamata ai componenti:



# Comp\_Footer.vue

Il componente si struttura all'interno del tag semantico `<footer>`.

Il footer si articola in due sotto sezioni: `#footer_upper` e `#footer_lower`.

`#footer_upper` presenta un layout molto simile a quello dello `<header>` salvo il fatto che il menù, in questo caso, non è in configurazione compressa, bensì espansa. Anche in questa fattispecie, si potrebbe pensare di attuare lo switch della configurazione del menù e di dinamicizzare il click sui menu items, indirizzando verso i punti semantici della pagina. In questa sezione, come nel resto del lavoro, per la rappresentazione del menù si fa ricorso al componente `Comp_Menu_Manager`.

`#footer_lower` presenta un testo con crediti (cliccabili ma non attivati) ed un menù con alcune icone social per le quali non è stata tuttavia implementata la funzione di indirizzamento ai social stessi. Ancora una volta, per la realizzazione del menù si ricorre al solito componente, avendo cura di settarne opportunamente i parametri via props.



# *Dati utilizzati in Comp\_Footer*

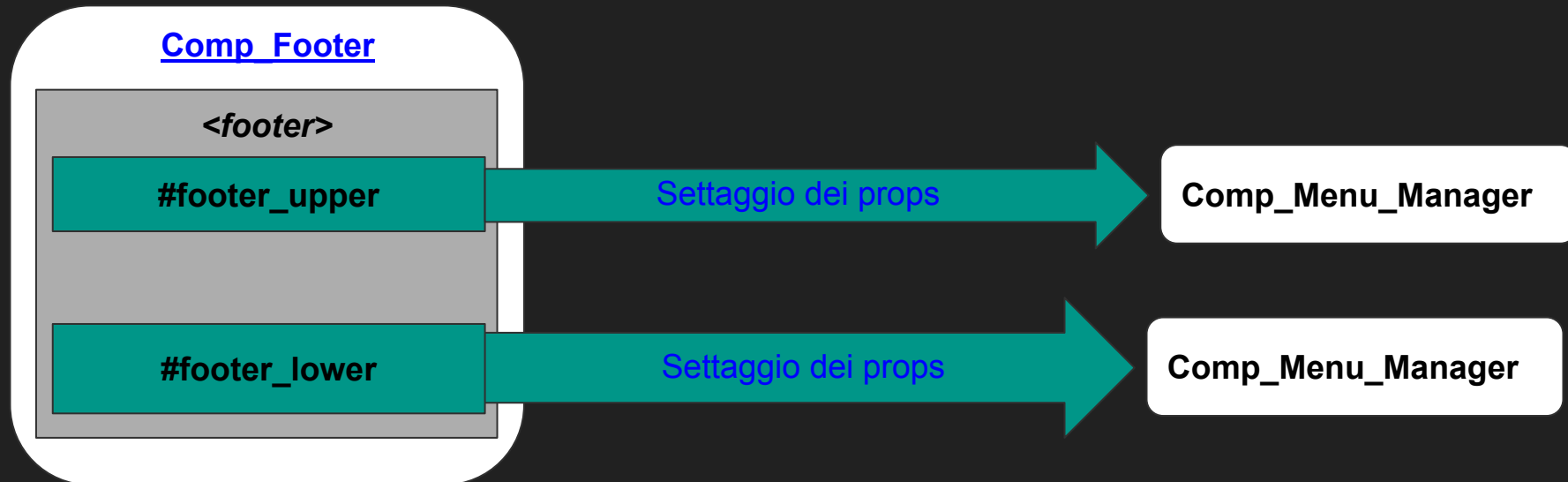
Le due chiamate a `Comp_Menu_Manager` avvengono mediante il settaggio dei props con i valori degli oggetti:

- `footer_menu` per la chiamata effettuata da `#footer_upper` e
- `social_menu` per la chiamata effettuata da `#footer_lower`

La struttura base dei due oggetti è la medesima ed è la stessa riscontrabile in tutte le altre chiamate al componente, salvo, ovviamente il valore delle chiavi.

La chiave `.menu_items.text`, anche in questo caso, assume il valore di testo (elemento del menù) o di icona, a seconda del valore della chiave `.category`.

# Struttura interna e chiamata ai componenti:



# Comp\_Banner.vue

Il componente in oggetto si occupa della visualizzazione di un layout statico o interattivo (a seconda del settaggio del suo unico props “**banner\_data**”).

Nel caso di settaggio statico (**banner\_data.clickable** = false) esso si limita a visualizzare del testo centrato, con eventuale background (width:100%) ed eventuali elementi decorativi, quali una barra colorata ed una linea anch’essa colorata (anche i colori dipendono dal settaggio del props).

Nel caso invece in cui risulti **banner\_data.clickable** = true, il banner si trasforma in un pulsante (width:100%) cliccabile e soggetto a transizione di stile all’hover e restituisce, mediante \$emit, in caso di click, un indice identificativo, utilizzato poi, dal chiamante per dare seguito all’operazione richiesta (funzione non implementata). A ragion veduta, si sarebbe potuto utilizzare il banner solo nella sua funzione statica, trasferendo la funzionalità cliccabile ad altri componenti già adibiti alla visualizzazione di pulsanti o elementi di menù. In eventuali sviluppi futuri ci si potrebbe orientare in questo senso.

**Comp\_Banner** non utilizza alcun sotto-componente e non presenta alcuna struttura dati specifica, ad esclusione del props.

# Comp\_Menu\_Manager.vue

Il componente `Comp_Menu_Manager` è uno dei più utilizzati nel progetto.

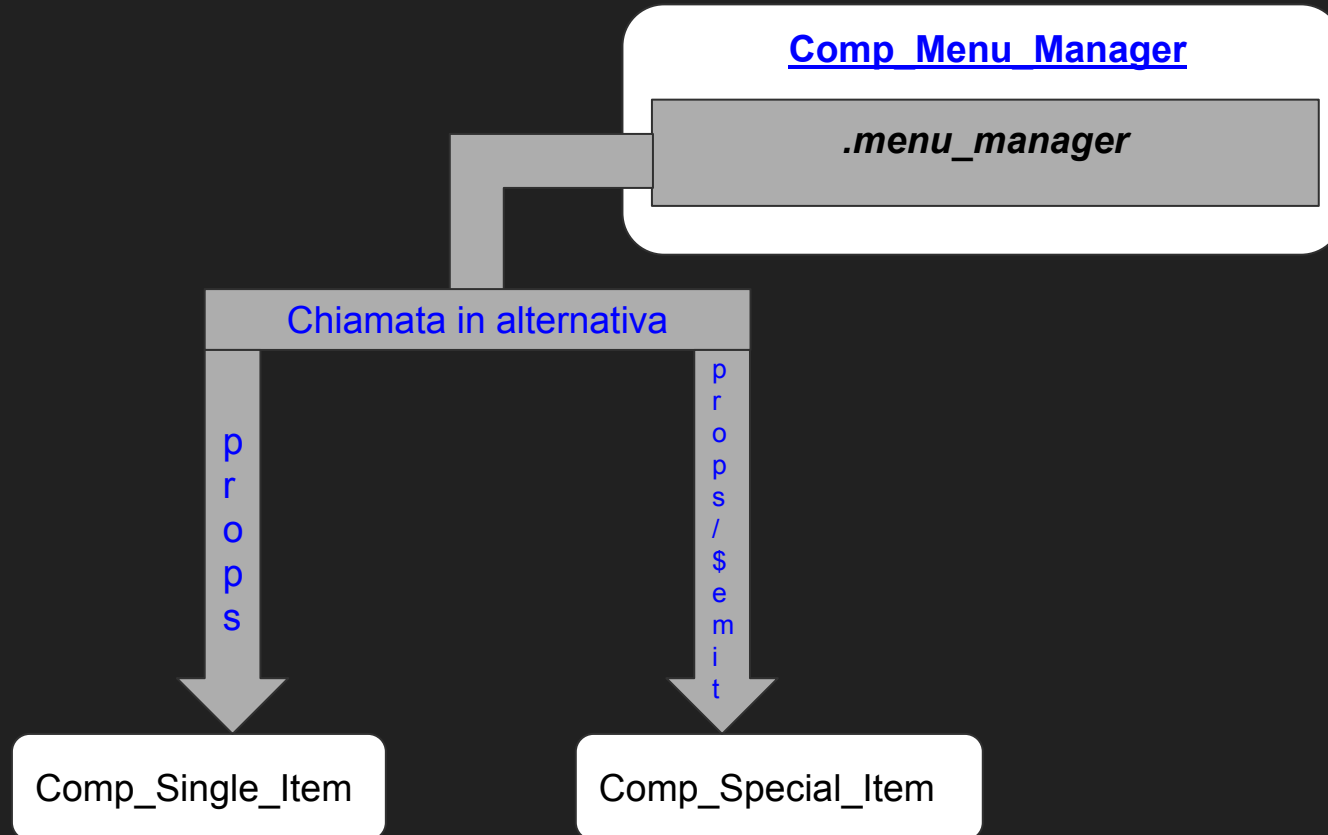
La sua funzione è quella di generare un qualsiasi menù (con orientamento orizzontale o verticale a seconda del valore assegnato al suo props booleano `is_horizontal`) e di invocare sotto-componenti per la generazione dei pulsanti o delle voci del menù stesso.

Nella maggior parte dei casi, il sotto-componente chiamato è `Comp_Single_Item` al quale vengono trasferiti i valori dei props ricevuti con l'aggiunta di un identificativo utilizzato poi dal sotto-componente per riconoscere se il menù in costruzione sia relativo alle icone social.

In un solo caso `Comp_Menu_Manager` invoca il sotto-componente `Comp_Special_Item`, ovvero quando dovrà generare il menù per “`Comp_Main / #live_dates_section`”, trattandosi di un menù con complessità più alta degli altri, poichè presenta un pulsante switchable sulla voce del menù e, se attivo, apre una sotto-sezione con elementi misti di tipo <immagine / testo / ulteriore pulsante cliccabile>.

Il props `menu_class`, oltre ad essere trasferito ai livelli inferiori, viene anche utilizzato da `Comp_Menu_Manager` per creare dei `:key` differenziati (a seconda del menù) nel v-for comune. Nel componente viene, nel caso, anche emesso il già visto `$emit(“rebounded”)` di rimbalzo tra `Comp_Special_Item` e `Comp_Main`.

# Struttura interna e chiamata ai componenti:



# *Comp\_Single\_Card.vue*

Il componente `Comp_Single_Card`, chiamato da “`Comp_Main / #card_set_section`” si occupa del posizionamento delle cards, singolarmente, della relativa formattazione e stile (anche nelle transizioni all’hover).

I suoi unici props “`current_card`” e “`index`” indicizzano la card e la posizionano opportunamente all’interno della griglia preesistente.

# Comp\_Single\_Item.vue

In `Comp_Single_Item` vengono generati e visualizzati gli elementi del menù chiamante.

In questo componente, a seconda del valore dei props, si determina se un elemento deve essere un `<button>` o un `<a>`, se si tratterà di un'icona o di normale testo e, ancora, se, come nel caso del menu social icons, ci debba essere del testo pop\_up sull'icona selezionata. In questo componente si determinano anche le classi di stile del singolo elemento, nel caso di elemento attivo, non attivo o in hover.

Non è stato implementato alcun `$emit` per dar seguito al click sulle voci ma lo si potrebbe fare in un eventuale upgrade futuro del progetto.

# Comp\_Special\_Item.vue

Questo componente viene chiamato solo in fase di generazione del menù relativo a “Comp\_Main / #live\_dates\_section”.

Nella generazione dei singoli menù items, il componente si occupa del posizionamento dei vari sotto-elementi, invoca il suo concorrente Comp\_Single\_Item per posizionare un ulteriore pulsante al suo interno e restituisce, mediante \$emit(“click\_on\_special”) il valore del menu item appena attivato / disattivato.



# Struttura interna e chiamata ai componenti:

