

ALMA MATER STUDIORUM – UNIVERSITÀ DI BOLOGNA
CAMPUS DI CESENA

Scuola di Scienze
Corso di Laurea in Ingegneria e Scienze Informatiche

BEACH VOLLEY SOCIETY: IL SOCIAL NETWORK DEI BEACHERS

Elaborato in
LABORATORIO DI BASI DI DATI

Relatrice
Prof.ssa ALESSANDRA
LUMINI

Presentata da
RICCARDO COFINI

Anno Accademico 2017 – 2018

INDICE

INTRODUZIONE.....	4
1 PROBLEMA AFFRONTATO	6
1.1 - OBIETTIVI	6
1.2 - ANALISI SISTEMI ESISTENTI.....	8
1.2.1 - FUBLES	8
1.2.2 - SPORT COMRADES.....	10
1.2.3 - ENJORE	11
1.3 - PROBLEMATICHE RELATIVE ALLO SVILUPPO ANDROID	13
2 ANALISI DEI REQUISITI	15
2.1 – REQUISITI DELLA APPLICAZIONE	16
2.2 – CARATTERISTICHE DEL SISTEMA	17
2.3 - UTILIZZO DEL SISTEMA	18
2.3.1 – CASI D’USO	18
2.3.2 - GESTISCI ACCOUNT	19
2.3.4 –GESTISCI LUOGHI.....	21
2.3.3 – GESTISCI PARTITA	22
2.3.5 – GESTISCI CERCHIA AMICI.....	23
2.4 - QUALITA’ DEL SOFTWARE.....	24
3 PROGETTAZIONE DEL SISTEMA.....	25
3.1 – ARCHITETTURA GENERALE.....	26
3.2 – BASE DATI.....	27
3.2.1 – UTENTI	27
3.2.2 – PARTITE	28
3.2.3 – CERCHIE	28
3.2.4 – LUOGHI	29
3.3 – FUNZIONI.....	30
3.3.1 - DIAGRAMMA DELLE ATTIVITÀ	30
3.4 - INTERFACCIA.....	31
3.4.1 – HOME PAGE	31
3.4.2 – LE MIE PARTITE.....	32
3.4.3 – IL MIO PROFILO	34
3.4.4 – LE MIE CERCHIE	35
3.4.5 – ELENCO CAMPI.....	36

4 IMPLEMENTAZIONE	37
4.1 – TECNOLOGIA UTILIZZATA	37
4.1.1 MY SQL.....	37
4.1.2 XML.....	37
4.1.3 FIREBASE.....	38
5 VALUTAZIONE DEI RISULTATI.....	39
5.1 – TEST	39
5.2 – USER EXPERIENCE.....	40
6 CONCLUSIONI	43
6.1 - SVILUPPI FUTURI	44
BIBLIOGRAFIA	45

INTRODUZIONE

Negli ultimi decenni il notevole sviluppo delle tecnologie mobile ha portato all'utilizzo delle stesse in ambiti che fino a qualche anno fa erano considerati preclusi a queste tipologie di sistemi, valutati troppo “deboli” per svolgere certe mansioni.

Questa evoluzione ha avuto un notevole impatto su tutti quei sistemi informatici che vengono utilizzati nel quotidiano per svolgere un vasto numero di funzioni, sia in ambito lavorativo e professionale che in ambito privato, si spazia infatti dalle applicazioni più comuni di intrattenimento, come la riproduzione di musica e video, o di videogioco, fino a quelle più complesse di Home Banking e di pagamento online. Non sono rare nemmeno le applicazioni di tipo gestionale che si interfacciano con un database, che permettono l'organizzazione e il controllo di attività più complesse.

L'obiettivo di questa tesi è quello di implementare funzionalità aggiuntive e migliorare l'interfaccia grafica e di navigazione di un'applicazione per sistemi mobile Android. Scopo dell'applicazione è quello di favorire l'organizzazione di partite di beach volley e di creare una community con la quale ritrovarsi per giocare in spiaggia e in strutture attrezzate. L'applicazione esistente permette ad ogni utente registrato di inserirsi in una partita o di crearne una nuova.

È possibile, inoltre, visualizzare le strutture vicine alla propria posizione tramite l'apposita mappa, o consultare l'elenco di tutti i campi da gioco disponibili nel sistema.

Ogni utente ha la possibilità infine di personalizzare il proprio profilo e di guadagnare punti, per scalare la classifica, in base al risultato delle partite giocate.

Nello specifico si andrà a riorganizzare la struttura del menù iniziale e di navigazione, in modo che sia più intuitivo e di facile utilizzo. Si cercherà inoltre di snellire la procedura per la creazione di una nuova partita e si implementerà la funzione di login e logout dell'applicazione, in modo che ogni utente possa collegarsi e scollegarsi a piacimento dal sistema.

La tesi è stata suddivisa in sei capitoli che riassumono le varie fasi del progetto.

Nel primo capitolo viene descritto il problema affrontato, si analizzano alcune delle applicazioni con funzionalità simili già presenti sul mercato e infine si fa una

panoramica su quelli che possono essere i problemi dello sviluppo di un'applicazione in Android.

Nel secondo capitolo viene presentata l'analisi dei requisiti software e hardware per lo sviluppo del progetto; verranno quindi presentati e discussi i contesti di utilizzo e le qualità che il software dovrebbe possedere, anche grazie all'uso di diagrammi UML.

Nel terzo capitolo viene descritta la fase di progettazione software del sistema, avvalendosi anche in questo caso di diagrammi UML per la definizione dei principali meccanismi di funzionamento, vengono inoltre mostrate le principali viste dell'applicazione e spiegate le loro funzionalità.

Il quarto capitolo è dedicato alle scelte implementative derivate dall'analisi precedente, nonché alle tecnologie utilizzate per l'implementazione delle principali sotto parti del sistema.

Il quinto capitolo descrive le fasi di test e valutazione delle prestazioni, e vengono inoltre analizzati i risultati ottenuti e l'usabilità del sistema.

Nel sesto ed ultimo capitolo sono presenti le conclusioni e possibili idee per gli sviluppi futuri.

CAPITOLO 1

PROBLEMA AFFRONTATO

1.1 - OBIETTIVI

Oggetto di questa tesi è l'implementazione di nuove funzionalità ad una applicazione per sistemi mobile Android che possa facilitare l'organizzazione di partite e tornei per appassionati di beach volley.

L'idea nasce dalla volontà di creare una community di beachers e sviluppare una piattaforma che permetta a tutti di conoscere in maniera rapida ed intuitiva quali sono i campi da gioco più frequentati.

Ogni utente registrato ha la possibilità di personalizzare il proprio profilo, aggiungendo la propria foto e inserendo dati relativi alle proprie caratteristiche fisiche e tecniche. Si hanno infatti informazioni sull'altezza, la data di nascita e la nazionalità, ma è possibile anche mettere in evidenza il proprio ruolo principale, la posizione preferita e il braccio "forte". Consultando il profilo di un giocatore è inoltre possibile visionare il numero di partite giocate e le statistiche degli ultimi match.

È possibile creare una nuova partita consultando l'elenco dei luoghi disponibili e selezionando il campo da gioco preferito. Oppure si può partecipare ad una partita gestita da un altro utente.

Ogni partita avrà una data e orario di inizio, entro la quale il giocatore si dovrà presentare al campo.

Per quanto un giocatore possa essere sportivo, a nessuno piace vincere troppo facilmente o perdere malamente, per questo è stato necessario introdurre un sistema di ranking, che associa ad ogni beacher un "punteggio" (simile all'ELO degli scacchi), in modo che tutti possano sapere in anticipo il livello degli altri partecipanti alla partita, e decidere di conseguenza se partecipare o meno. Questo porta alla creazione di una classifica.

Ogni giocatore guadagna o perde posizioni in classifica in base al risultato delle partite che gioca, creando una sana competizione tra i membri e stimolando l'uso dell'app.

L'applicazione infatti, al termine di ogni partita, deve offrire la possibilità di registrarne il risultato e renderlo consultabile dagli altri utenti, andando a creare uno storico delle partite giocate in ogni campo.

Per facilitarne la diffusione è necessario curare l'aspetto social permettendo di organizzare partite e tornei senza limiti geografici, permettendo di inserire nuovi campi da gioco.

L'utente deve inoltre avere la possibilità di crearsi una lista di giocatori "preferiti" con i quali si è trovato particolarmente bene e con cui vorrebbe rigiocare spesso. Per fare ciò l'applicazione deve permettere la creazione di una cerchia di amici e associarla al luogo di gioco scelto, in questo modo è possibile rimanere in contatto e organizzare match con giocatori con i quali si hanno avuto esperienze positive.

1.2 - ANALISI SISTEMI ESISTENTI

Non essendo impresa facile organizzare partite, a causa dell'incrocio di svariate variabili quali data, campo, orario, giocatori, negli ultimi anni sono state realizzate applicazioni con funzionalità simili orientate ad altri sport per estendere al di fuori della propria cerchia di amici la partecipazione a partite e tornei.

Nel seguito saranno presentati alcuni dei sistemi più noti.

1.2.1 - FUBLES



Figura 1 Home page FUBLES

Fubles (Figura 1) è un social network che serve principalmente ad organizzare partite di calcetto, mettendo in contatto tra loro i giocatori di una determinata città o zona. Teoricamente si possono organizzare anche partite di basket o pallavolo, ma quasi nessuno lo usa per questo.

Si possono cercare partite nelle vicinanze oppure organizzarne di private, restringendo le iscrizioni a invito e sfruttandolo di fatto come alternativa a Whatsapp o a una conversazione di Facebook. Se poi non arrivi a 10 si possono aprire le iscrizioni agli sconosciuti così da non far saltare la partita.

La facilità di Fubles ha lo stesso fascino immateriale delle cose

migliori della nostra epoca. Uno dei fondatori, tutti italiani, lo descrive così: «L'utente organizza la partita, può scegliere di proporla a un giro di amici o a tutta la community. Chi vuole giocare va sul sito, vede nella sua città se ci sono partite disponibili e partecipa. Oggi può fare tutto anche via smartphone con la nostra App». L'autentica esperienza Fubles sta nelle partite pubbliche, in cui chiunque può iscriversi fino ad un'ora prima dell'inizio, scegliendo anche la squadra (maglia

chiara vs scura) e ci si può rimuovere minimo quattro ore prima, oltre si può chiedere una sostituzione, ma se non la si ottiene la responsabilità rimane propria.

Quando si arriva a 10 la partita è chiusa; se non ci si presenta i giocatori a fine partita ti segnaleranno il “bidone”, che rimarrà impresso sulla tua scheda e funzionerà come il “campanello dei lebbrosi”. Così gli altri sono avvisati riguardo la tua scarsa affidabilità e il sistema prova a rendersi autoimmune.

A fine partita i giocatori possono darsi dei voti a vicenda e, sulla base di questi, verrà eletto anche un migliore in campo. I voti personali si accumuleranno nella propria scheda, andando a formare una media voto visibile pubblicamente, insieme alle partite disputate, il numero di bidoni, le volte che si è stati migliori in campo e il “rango”. [1]

Le funzionalità principale di Fubles sono:

1. **PROFILO GIOCATORE**

Crea il profilo con la tua figurina e seleziona una zona in cui vorresti giocare.

2. **TROVA E GIOCA UNA PARTITA.**

Fubles ti comunica quando ci sono partite disponibili nei tuoi dintorni o quando un tuo amico ti invita a giocare.

3. **PROPONI UNA PARTITA!**

Se non ci sono partite nella tua zona, puoi proporre una tu! Prenota il campo e crea la partita indicando il centro sportivo, lo sport e l'orario.

4. **CENTRI SPORTIVI:**

La lista dei centri sportivi è aggiornata costantemente dai membri della community.

[2]

1.2.2 - SPORT COMRADES

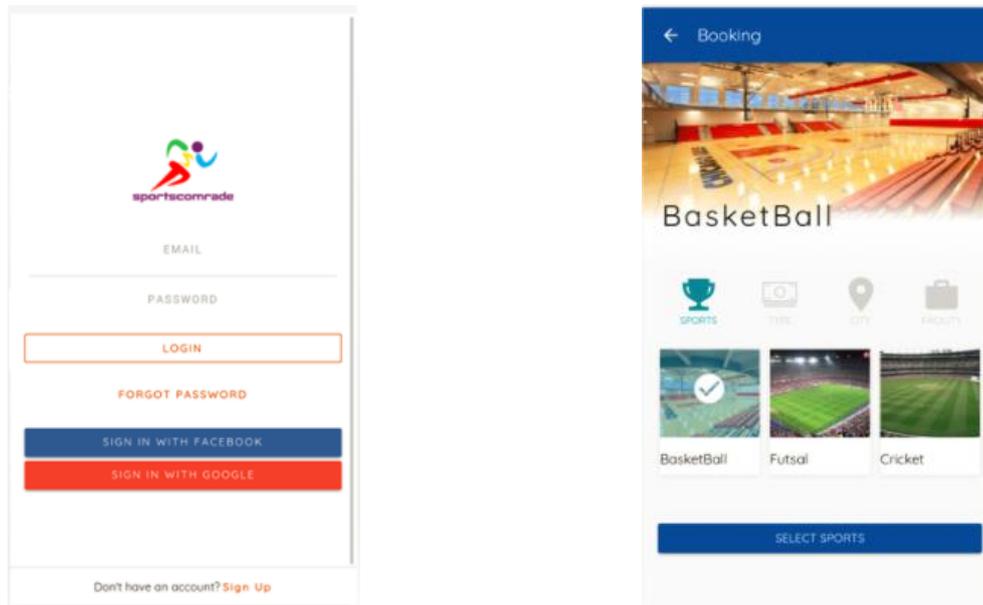


Figura 2 Login & Home Sport Comrades

Sport Comrades (Figura 2) è un'app che permette agli utenti di prenotare un campo da gioco nelle strutture disponibili nella propria zona. L'idea di questa app è quella di rendere più semplice l'organizzazione e la comunicazione tra gli organizzatori e le strutture ospitanti. L'utente ha la possibilità di monitorare le proprie prenotazioni, cronologia partite, squadre e amici. È possibile aggiungere altri utenti alla propria lista amici e chattare con loro per organizzare la prossima partita. Gli utenti possono anche darsi un voto a vicenda in base al loro comportamento, sportività e collaborazione. Gli amministratori delle strutture ospitanti possono monitorare le prenotazioni che gli utenti effettuano e confermarle tramite l'app o il sito internet. Hanno inoltre la possibilità di bloccare gli utenti indesiderati, impedendo loro di effettuare ulteriori prenotazioni, in caso di comportamento scorretto. [3]

1.2.3 - ENJORE



Figura 3 Home & Web Enjore

Enjore (Figura 3) nasce nel 2012 da un'idea di Nicola Taranto, oggi CEO della startup. L'idea di Nicola è stata quella di creare una piattaforma che permettesse agli utenti di gestire e vivere competizioni attraverso un sistema organizzativo molto semplice. Enjore, come qualsiasi social network, sfrutta le potenzialità legate alla condivisione di informazioni e contenuti multimediali, all'interno della propria community. È quindi un vero social network che, grazie alle sue caratteristiche, dà modo a organizzatori, giocatori e tifosi, di vivere on-line le loro passioni.

Enjore aumenta il coinvolgimento degli utenti e dei partecipanti grazie alle bacheche che consentono di pubblicare foto e video delle partite, di commentare gli incontri e votare le prestazioni dei singoli giocatori sia dal computer sia da smartphone.

I tornei possono essere organizzati secondo diverse tipologie: campionato all'italiana, a eliminazione diretta o a gironi + playoff. Le competizioni supportate, ognuna con regole e statistiche specifiche, spaziano dal calcio e calcetto alla pallavolo, dal rugby alla pallanuoto, dalla pallacanestro al tennis, e poi baseball, hockey, football americano, paddle tennis, softball, tennis tavolo e pallamano. Tra i videogame, Fifa, Dota 2, PES, Smite e League of Legend. E' possibile inoltre usare Enjore per competizioni di calcio-balilla e Subbuteo. Le squadre che partecipano al torneo possono essere personalizzate con foto, riconoscimenti, ruoli e staff. Ogni partita può essere dettagliata nei suoi risultati con tabellini che indicano marcatori, giocatori migliori, campi da gioco, condizioni meteo, sanzioni disciplinari, e così via. Enjore ha subito catturato l'interesse degli organizzatori di tornei e nel giro di

pochi giorni dal lancio, erano già un centinaio i tornei organizzati attraverso la piattaforma.

Pochi mesi dopo l'inizio dell'avventura, nel marzo del 2013 viene costituita la società e arriva il primo investimento seed di 150 mila euro, che permette a Enjore di migliorare la piattaforma e di far crescere il proprio team. [4]

Funzionalità dell'App:

- Refertazione partite
- Ricerca Tornei
- Consultazione Classifiche Squadre e Giocatori
- Consultazione Calendari
- Consultazione Partite
- Elenchi Squadre e Giocatori
- Commento Partite
- Votazione prestazioni Giocatori
- Caricamento foto Torneo
- Gallery foto Torneo

[5]

1.3 - PROBLEMATICHE RELATIVE ALLO SVILUPPO ANDROID

Version	Codename	API	Distribution
2.3.3 - 2.3.7	Gingerbread	10	0.3%
4.0.3 - 4.0.4	Ice Cream Sandwich	15	0.4%
4.1.x	Jelly Bean	16	1.7%
4.2.x		17	2.6%
4.3		18	0.7%
4.4	KitKat	19	12.0%
5.0	Lollipop	21	5.4%
5.1		22	19.2%
6.0	Marshmallow	23	28.1%
7.0	Nougat	24	22.3%
7.1		25	6.2%
8.0	Oreo	26	0.8%
8.1		27	0.3%

Figura 4 - Versioni di Android e rispettive percentuali di diffusione. (dati riferiti al 5 Febbraio 2018)

Una delle problematiche principali che un programmatore Android si trova ad affrontare riguarda la frammentazione delle versioni presenti nei diversi dispositivi. Nel tempo Google ha progettato e implementato molteplici versioni del sistema operativo. Ogni nuova versione era atta a migliorare, correggere e modernizzare quanto già era stato fatto con la versione antecedente.

Dal 2008 infatti, sono state 15 le versioni di Android create, molte delle quali presentano anche varie sotto-versioni che si differenziano ulteriormente da quella principale.

Come riporta il grafico in figura 4, le versioni utilizzate tutt'oggi da un

numero di dispositivi maggiori dello 0,1% del totale sono 8 (i dispositivi con uno share più basso non sono considerati nel grafico). [6]

Il problema principale deriva da quello che in altri contesti può essere considerato uno dei punti di forza dell'ecosistema Android: il vasto numero di dispositivi. Nel tempo infatti, tantissimi produttori di device hanno scelto Android come sistema operativo per i prodotti da loro creati; quasi tutti i produttori non utilizzano la versione stock di Android, ma personalizzano il sistema operativo aggiungendo funzionalità e modificando in parte l'interfaccia.

Nel momento in cui Google crea una nuova versione del sistema operativo però, non tutti i device che utilizzano la versione precedente verranno aggiornati. Alcuni infatti non saranno idonei, per problemi legati all'hardware, a utilizzare la nuova versione, ma la gran parte dei mancati aggiornamenti è dovuta al fatto che i diversi produttori

non sempre vogliono investire tempo e risorse per aggiornare device non più recentissimi.

Tutto questa frammentazione si traduce in possibili problematiche nell'utilizzo di particolari librerie o framework, che spesso sono supportati da una certa versione in poi; bisognerà quindi scegliere attentamente quale versione minima supportare nel progetto, per non rischiare di escludere alcune funzionalità indispensabili. In questo lavoro si è scelto di utilizzare la versione 16.

Un'altra problematica da considerare sempre in riferimento al vasto numero di device in circolazione riguarda il reparto hardware degli stessi. Oltre ad una frammentazione del sistema operativo infatti, lavorando con Android si è chiamati a dover provvedere alla gestione di tutta una serie di dispositivi con display che variano per dimensione e risoluzione.

Un'altra problematica da considerare nello sviluppare un'applicazione Android riguarda il dispendio in termini di energia utilizzata e potenza di calcolo richiesta. Oggigiorno, nella maggior parte dei casi, questo non rappresenta più un problema, vista l'evoluzione che i dispositivi hanno raggiunto, ma quando si ha a che fare con sistemi mobile, si può facilmente incappare in situazioni in cui il sistema, pur operando correttamente e ottenendo alla fine risultati positivi, richieda un tempo di calcolo troppo lungo per poter definire l'applicazione utilizzabile. Altro aspetto da considerare è il consumo energetico, in quanto le batterie degli smartphone hanno una portata limitata e un'applicazione deve riuscire a svolgere le sue funzioni gravando sulla batteria il meno possibile.

Una possibile problematica quindi potrà essere quella di operare attraverso una serie di processi ben progettati, che permettano di raggiungere il risultato prestabilito avendo comunque riguardo per le risorse e senza che questi richiedano un quantitativo di tempo che rendano l'applicazione poco funzionale.

CAPITOLO 2

ANALISI DEI REQUISITI

Un requisito è una scrittura più o meno formale e rigorosa di una caratteristica del sistema, fatta dallo specialista. La gestione dei requisiti (acquisizione, analisi, negoziazione, specifica, validazione) è il primo passo del processo di sviluppo e una delle fasi più critiche dello sviluppo software, perché influenza in modo diretto il successo o il fallimento dei progetti.

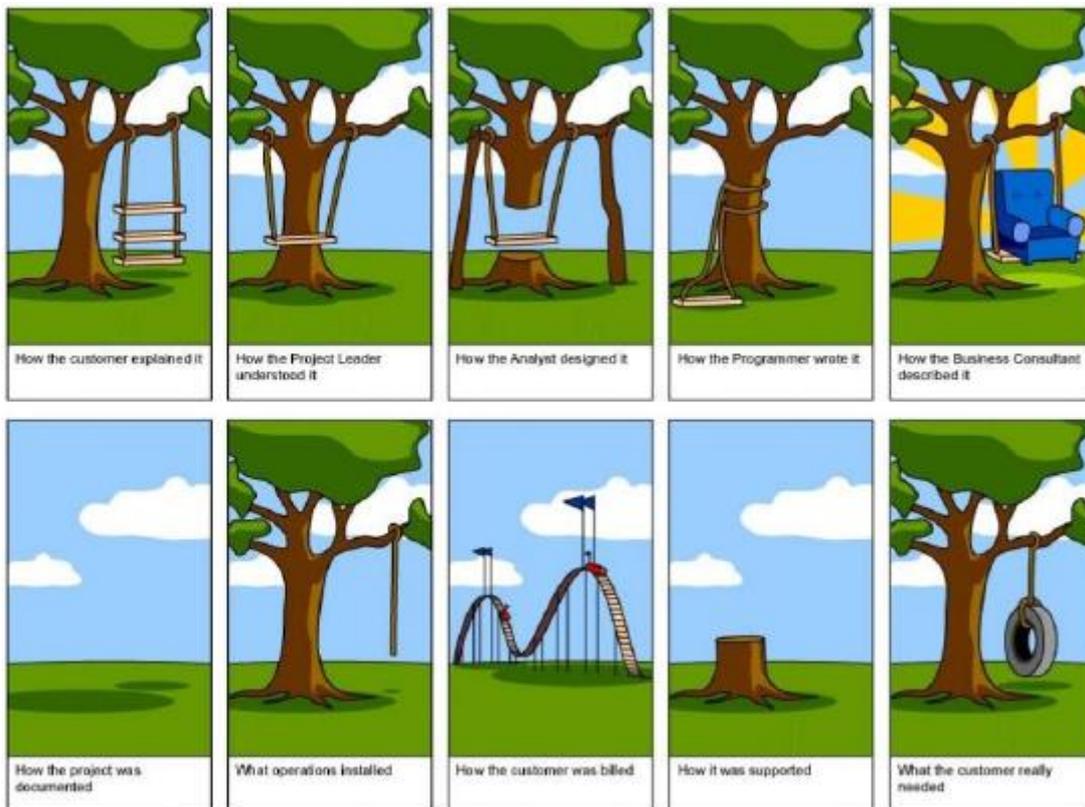


Figura 5 Metafora dell'altalena

L'altalena rappresentata in figura 5 è la metafora più comune per la gestione dei requisiti nei progetti software. Lo scopo che si prefigge questo documento è quello di formalizzare ciò che dovrà essere sviluppato e implementato nelle fasi successive del progetto. Per le specifiche di progettazione verranno utilizzati diagrammi UML, in quanto sono uno standard ed evitano le possibili ambiguità.

2.1 – REQUISITI DELLA APPLICAZIONE

Come già specificato precedentemente questa applicazione ha lo scopo di creare comunità di appassionati di beach volley, e facilitare l'organizzazione e la condivisione di partite e tornei.

Le principali entità gestite dell'applicazione sono:

- **GESTIONE UTENTI\ACCOUNT PERSONALE**

Al primo accesso, ad ogni utente verrà richiesta la creazione di un account personale.

Il giocatore avrà la possibilità di personalizzare il proprio profilo inserendo i propri dati anagrafici e caratteristiche di beacher. Potrà anche caricare una immagine profilo, per essere riconosciuto dagli altri utenti. All'interno del proprio account personale si potrà creare una lista amici e salvare i campi da gioco più frequentati, in modo da poterli avere sempre a disposizione da qualsiasi dispositivo si effettui l'accesso.

- **INSERIMENTO\ELIMINAZIONE LUOGHI DI GIOCO**

I luoghi di gioco sono inseribili da ogni giocatore, in modo da poter facilitare la diffusione dell'applicazione, possono invece essere eliminati solo dall'utente che li ha inseriti. Qualora un utente volesse inserire un nuovo luogo dovrà specificare le sue caratteristiche, ovvero la posizione fisica, il numero di campi da gioco disponibili e se è libero o a pagamento. Un luogo può ospitare una o più partite simultaneamente, in base al numero di campi da gioco disponibili. Viene mantenuta traccia delle partite giocate in ogni luogo, così da creare uno storico partite consultabile dai giocatori, per avere a disposizione, oltre all'archivio dei match una classifica aggiornata per tutti i membri dei gruppi.

- **CREAZIONE\PARTECIPAZIONE A PARTITE**

Cuore del sistema è la gestione delle partite. Ogni nuova partita creata dovrà essere associata ad un campo da gioco, svolgersi in una data e in un orario stabilito ed avere una durata, in modo da evitare sovrapposizioni con altre partite. Le partite possono essere aperte, con restrizioni o private. Una partita con restrizioni è accessibile solo se si soddisfano le condizioni per partecipare alla partita (sesso, elo, ruolo, ecc). Nel caso di partite private, sarà l'utente creatore a dover invitare i giocatori, o permettere l'accesso solo alla propria lista amici. Al termine di ogni partita viene registrato l'esito e segnato il punteggio

- **CERCHIE DI AMICI**

Ogni utente ha la possibilità di creare ed essere inserito in delle cerchie, in modo da poter rimanere in contatto più facilmente con i propri amici e giocatori con cui interagisce più spesso. Ogni cerchia è associata ad un luogo di gioco, ha un nome che la contraddistingue ed il proprio elenco di partecipanti, è visibile da tutti gli utenti.

2.2 – CARATTERISTICHE DEL SISTEMA

Il sistema deve possedere delle caratteristiche di base imprescindibili ad ogni software

- **UTILIZZO MIRATO DELLE RISORSE**

Trattandosi di un progetto in ambito mobile, le risorse sono un aspetto delicato da considerare. Nell'adempimento dei suoi compiti il sistema deve operare in maniera tale da poter garantire un utilizzo intelligente delle risorse, di norma limitate.

- **REATTIVITÀ**

Il sistema potrà essere utilizzato anche in situazioni nelle quali il tempo è un fattore rilevante. Pertanto, tutto il complesso dovrà essere reattivo nei diversi passaggi in modo da non gravare sull'esecuzione del processo generale.

- **FACILITÀ DI UTILIZZO**

Il sistema deve essere agevolmente utilizzabile ed “user friendly”. Per garantire questo, vi deve essere una standardizzazione dei comandi in modo che siano riconosciuti da tutti e come tali associabili facilmente alla loro funzione.

- **PREVENZIONE DEGLI ERRORI**

Il sistema dovrà essere robusto e saper prevenire ed eventualmente gestire tutti quei casi in cui si potrebbero riscontrare degli errori, anche quelli causati da un errato utilizzo del software.

2.3 - UTILIZZO DEL SISTEMA

Un caso d'uso specifica cosa ci si aspetta da un sistema (“*what?*”) ma nasconde il suo comportamento (“*how?*”). E' una sequenza di azioni, con varianti, che producono un risultato osservabile da un attore e rappresenta un requisito funzionale. Ogni sequenza (detta scenario) rappresenta l'interazione di entità esterne al sistema (attori) con il sistema stesso o sue componenti. Il flusso principale degli eventi viene separato dalle varianti alternative.

L'attore rappresenta un soggetto o un'entità che non fa parte del sistema, ma interagisce in qualche modo con esso. Individua un ruolo che l'utente ricopre nell'interagire con il sistema. Gli attori eseguono i casi d'uso.

Nel nostro caso l'attore del sistema è l'utente (giocatore di beachvolley), i casi d'uso sono: Registra account, crea/gestisci partita, visualizza luoghi, crea/gestisci cerchia.

2.3.1 – CASI D’USO

Il diagramma dei casi d'uso è il modello che descrive i requisiti del sistema in termini di funzionalità (Casi d'Uso) e ambiente circostante (Attori). Mostra cosa deve fare il sistema.

Di seguito vengono descritte le specifiche dei casi d'uso con scenari principali e alternativi. Uno scenario è una sequenza di passi che descrivono l'interazione tra un sistema e un attore (che dovrebbe trarre vantaggio dall'interazione). Nello specifico, il mio contributo all'applicazione è stato quello di migliorare la gestione delle partite, permettendo di creare una nuova partita visualizzando a video l'elenco dei luoghi disponibili, con la possibilità di una ricerca tramite città\nome campo e non dover necessariamente ricercarlo tramite la mappa.

2.3.2 - GESTISCI ACCOUNT

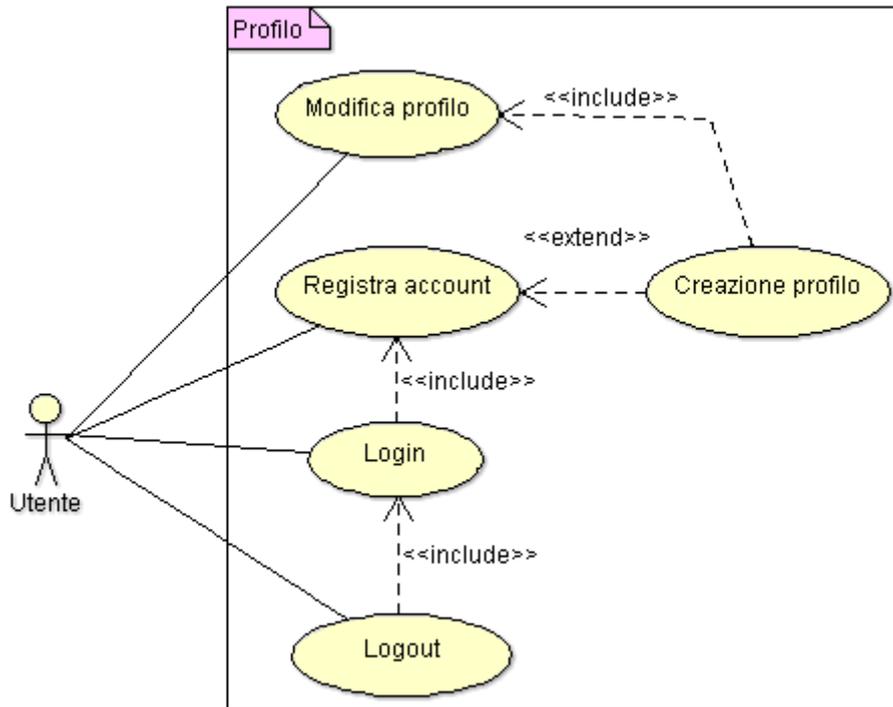


Figura 6 - Caso d'uso "Gestisci account"

Nome caso d'uso: Gestisci account (Figura 6)

Attori: Utente

Precondizioni:

1. l'utente non è registrato nel sistema

Scenario principale:

1. l'utente accede per la prima volta al sistema
2. il sistema visualizza la scelta tra Login e Registra account
3. l'utente sceglie Registra account
4. il sistema visualizza i campi da compilare per registrare l'account
5. l'utente inserisce i dati di registrazione
6. il sistema verifica la correttezza dei dati, se i dati non sono corretti il sistema evidenzia i campi da modificare e ritorna al punto 4
7. se i dati sono corretti il sistema conclude la fase di registrazione
8. l'utente può effettuare il login

Postcondizioni:

1. L'utente è registrato nel sistema e può modificare il suo profilo

Scenario secondario:

1. l'utente accede al caso d'uso registra account
2. il sistema visualizza i campi da compilare per la registrazione dell'account
3. l'utente inserisce i dati
4. il sistema verifica che i dati sono già presenti nel database
5. il sistema avvisa l'utente che i dati inseriti sono già stati utilizzati

Postcondizioni:

1. L'utente effettua il login

2.3.4 –GESTISCI LUOGHI

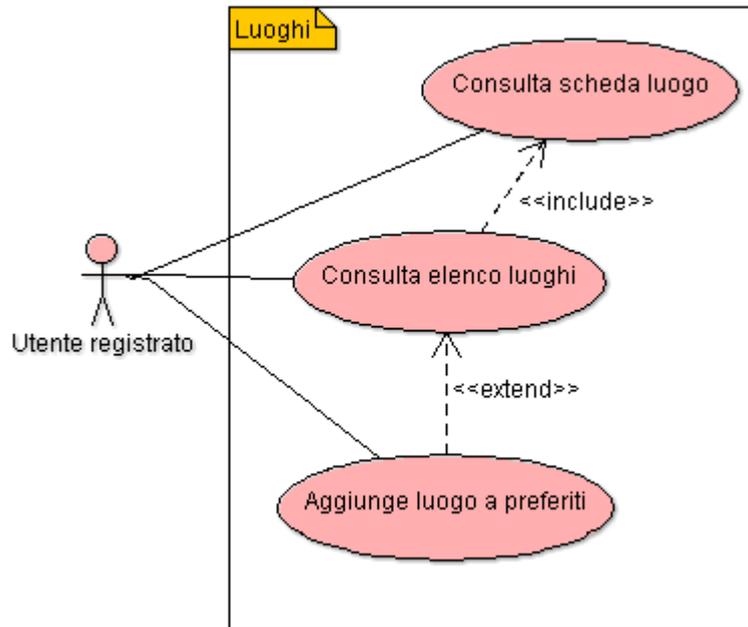


Figura 7 - Caso d'uso "Gestisci luoghi"

Nome caso d'uso: Gestisci luoghi (Figura 7)

Attori: Utente registrato

Precondizioni:

1. L'utente ha effettuato il login nel sistema

Scenario principale:

1. il caso d'uso comincia quando l'utente vuole visualizzare i luoghi inseriti nel sistema
2. il sistema visualizza i luoghi di gioco registrati
3. l'utente sceglie un luogo
4. il sistema visualizza la scheda con le info del luogo

Postcondizioni:

1. L'utente può inserire il luogo selezionato nella propria lista dei preferiti
2. L'utente può iniziare il caso d'uso crea partita

2.3.3 – GESTISCI PARTITA

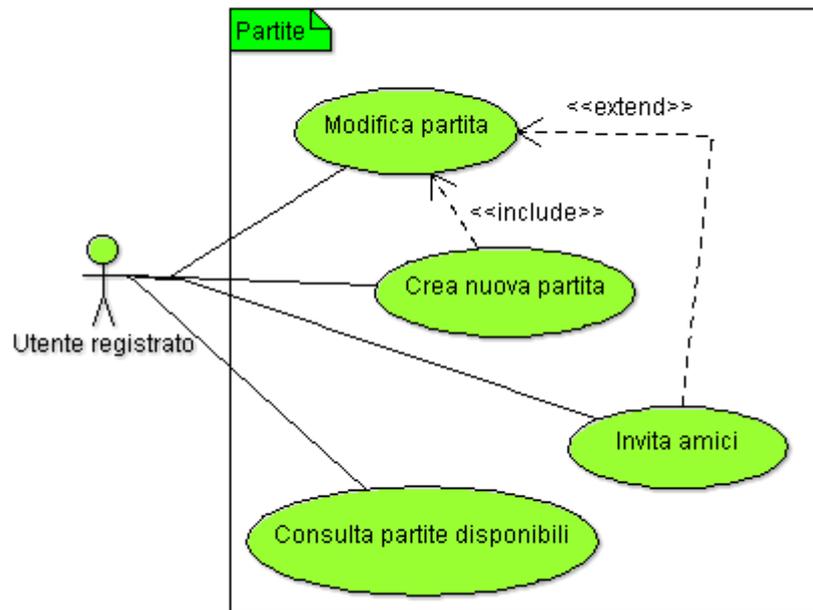


Figura 8 - Caso d'uso "Gestisci partita"

Nome caso d'uso: Gestisci partita (Figura 8)

Attori: Utente registrato

Precondizioni:

1. L'utente ha selezionato un luogo di gioco

Scenario principale:

1. il sistema visualizza i campi da compilare per creare la partita
2. l'utente inserisce i dati
3. il sistema verifica la correttezza dei dati
 - (a) se i dati non sono corretti il sistema evidenzia i campi da modificare
4. se i dati sono corretti il sistema aggiunge la partita

Postcondizioni:

1. L'utente può modificare la partita e invitare amici

2.3.5 – GESTISCI CERCHIA AMICI

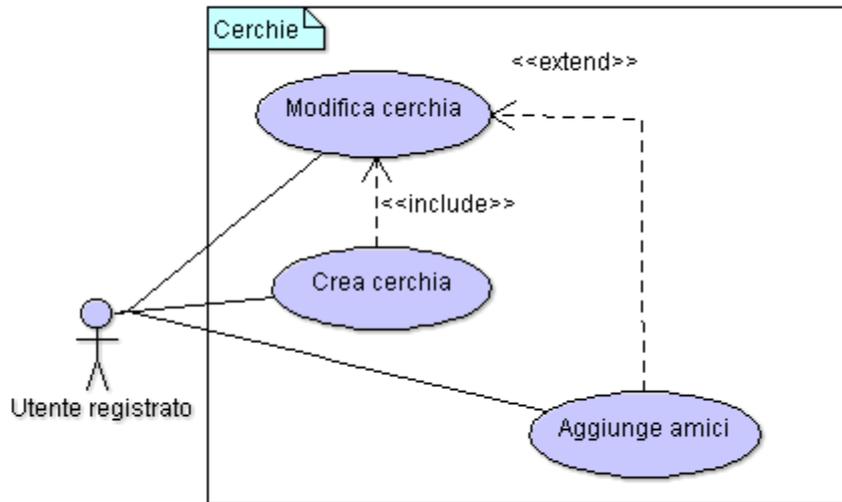


Figura 9 - Caso d'uso "Gestisci cerchia amici"

Nome caso d'uso: Gestisci cerchia amici (Figura 9)

Attori: Utente registrato

Precondizioni:

1. L'utente ha effettuato il login nel sistema

Scenario principale:

1. il caso d'uso comincia quando l'utente vuole creare una nuova cerchia di amici.
2. il sistema visualizza la scheda con le informazioni da inserire
3. l'utente inserisce i dati
4. l'utente sceglie quali amici inserire nella cerchia
5. il sistema aggiunge la nuova cerchia

Postcondizioni:

1. L'utente può visualizzare e modificare le proprie cerchie

2.4 - QUALITA' DEL SOFTWARE

Alla luce delle analisi effettuate e delle specifiche del progetto, è possibile individuare delle qualità sulle quali dovrà essere posta una particolare attenzione, che sono qui di seguito descritte:

- Correttezza. Un software si può definire corretto se rispetta le specifiche di progetto, ovvero se è in linea con quanto previsto dalla fase di analisi.
- Affidabilità. Un software è affidabile se l'utente può dipendere da esso, ovvero se riesce a gestire i malfunzionamenti senza affliggere l'utilizzatore.
- Efficienza. Un software è definito efficiente se utilizza intelligentemente le risorse disponibili indipendentemente dal carico di lavoro richiesto.
- Usabilità. Un software presenta tale caratteristica se presenta un'interfaccia utente immediatamente comprensibile e di facile utilizzo.
- Interoperabilità. Un software si può definire tale se non risulta essere un sistema chiuso, ma permette di gestire in maniera agile i risultati ottenuti tramite il suo utilizzo; inoltre lo stesso deve poter operare con altri sistemi.
- Robustezza. Un software si può definire robusto se riesce a funzionare in maniera controllata anche in corrispondenza di circostanze non previste dalle specifiche.

CAPITOLO 3

PROGETTAZIONE DEL SISTEMA

Nel seguente capitolo verranno descritte le scelte progettuali derivate dall'analisi precedentemente effettuata; verranno quindi definite le caratteristiche dell'architettura generale e delle sotto parti del sistema, corredate da diagrammi UML.

Le diverse fasi progettuali saranno strutturate in modo tale da risultare più astratte possibile rispetto al restante sistema, per facilitare il riutilizzo delle logiche di progetto.

Nella seguente immagine (Figura 10) possiamo vedere riportata la struttura del database nel quale vengono archiviati ed organizzati tutti i dati dell'applicazione.

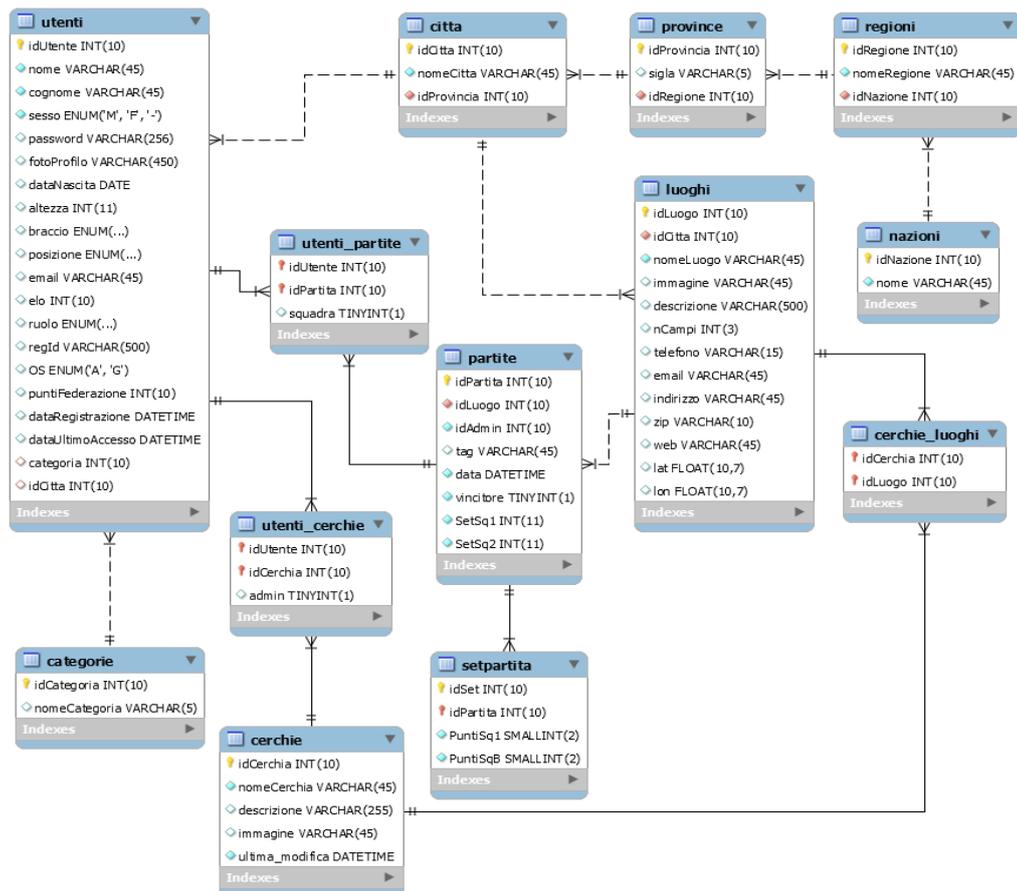


Figura 10 – Struttura database

3.1 – ARCHITETTURA GENERALE

Sulla base dei requisiti progettuali che sono stati analizzati precedentemente (vedi Sezione 2.1), possiamo ora definire quella che sarà la struttura generale del sistema; queste specifiche saranno poi utilizzate nella fase di implementazione.

Possiamo definire a livello progettuale quattro principali entità dell'architettura del sistema; Utenti, luoghi, partite, cerchie.

Gli attributi e le associazioni che intercorrono tra esse sono rappresentate nel seguente schema Entity/Relationship (Figura 11).

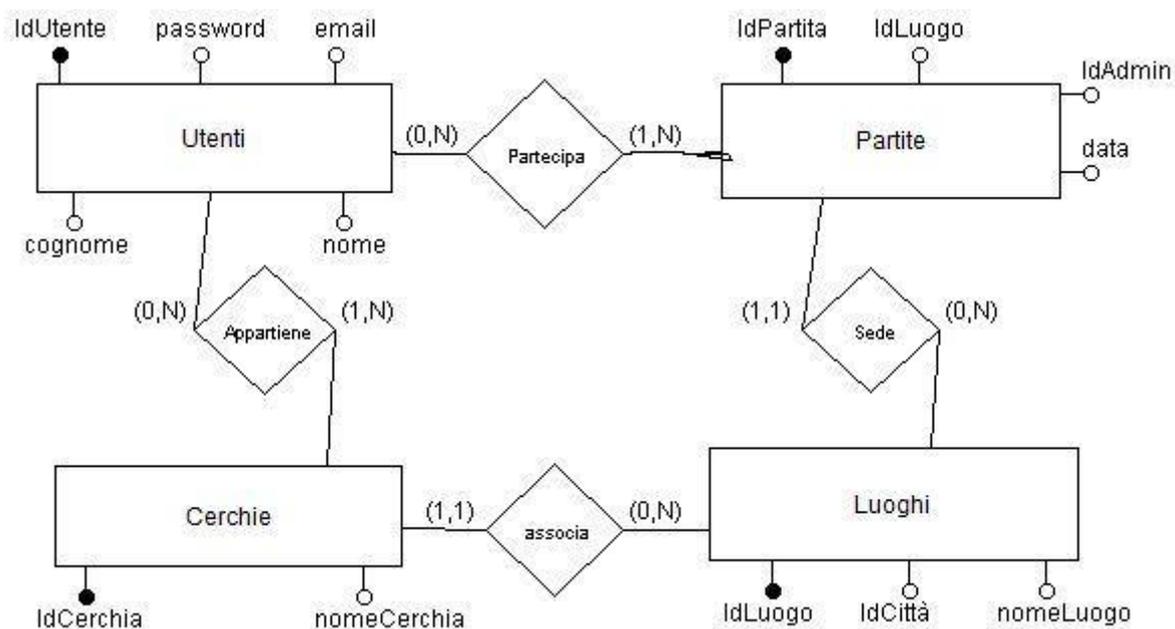


Figura 11 - Schema Entity/Relationship

3.2 – BASE DATI

Lo schema logico ottenuto è il seguente:

UTENTI (IdUtente, nome, cognome, email, password)

PARTITE (IdParita, IdLuogo, IdAdmin, data)

CERCHIE (IdCerchia, nomeCerchia)

LUOGHI (IdLuogo, IdCittà, nome)

UTENTI_PARTECIPA_PARTITE (IdUtente, IdPartita)

UTENTI_APPARTIENE_CERCHIE (IdUtente, IdCerchia)

PARTITE_SEDE_LUOGHI (IdPartita)

CERCHIE_ASSOCIA_LUOGHI (IdCerchia)

Nelle sezioni seguenti verrà approfondita l'analisi delle entità, precisando il ruolo di ciascuna.

3.2.1 – UTENTI

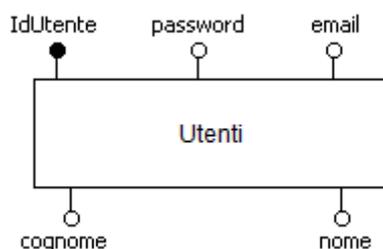


Figura 11.1 – Entità UTENTI

L'entità UTENTE (Figura 11.1) serve per tenere traccia di tutti gli utilizzatori del sistema (beachers). Il compito principale di UTENTE è la gestione delle credenziali per poter effettuare il login tramite e-mail e password scelti dall'utilizzatore in fase di registrazione.

Gli attributi di UTENTE sono i seguenti:

IdUtente: serve per identificare l'utente in maniera univoca all'interno del sistema l'utente (Primary key)

Email: indirizzo email associato all'account al momento della registrazione.

Password: utilizzata in fase di login insieme all' e-mail per autenticare l'utente nel sistema. Questo campo per motivi di sicurezza viene salvato nel database criptato.

Nome: serve per visualizzare il nome dell'utente all'interno dell'applicazione

Cognome: serve per visualizzare il cognome dell'utente all'interno dell'applicazione

3.2.2 – PARTITE

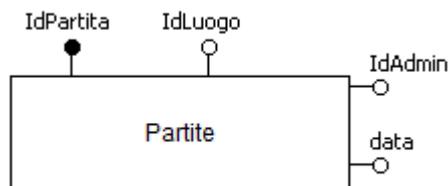


Figura 11.2 – Entità PARTITE

L'entità PARTITE (Figura 11.2) serve a tenere traccia di tutte le partite create dai giocatori.

Gli attributi di PARTITE sono:

IdPartita: serve per identificare la partita in maniera univoca all'interno del sistema (Primary key)

IdLuogo; serve per identificare il luogo associato alla partita (Foreign key)

IdAdmin: serve per identificare l'utente che ha creato la partita (Foreign key)

Data: serve per definire la data in cui si svolgerà la partita

3.2.3 – CERCHIE



Figura 11.3 – Entità CERCHIE

L'entità CERCHIE (Figura 11.3) serve per gestire i gruppi di gioco degli utenti.

Gli attributi di CERCHIE sono:

IdCerchia: serve per identificare la cerchia in maniera univoca all'interno del sistema (Primary key)

NomeCerchia: serve per visualizzare il nome della cerchia all'interno dell'applicazione

3.2.4 – LUOGHI

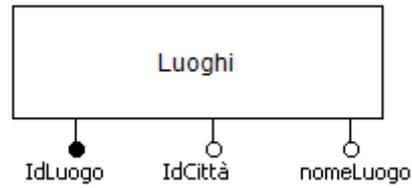


Figura 11.4 – Entità LUOGHI

L'entità LUOGHI (Figura 11.4) serve per gestire i vari luoghi di gioco inseriti dagli utenti.

Gli attributi di LUOGHI sono:

IdLuogo: serve per identificare il luogo in maniera univoca all'interno del sistema (Primary key)

IdCittà: serve per associare ogni luogo ad una città (Foreign key)

Nome Luogo: serve per visualizzare il nome del luogo all'interno dell'applicazione

3.3 – FUNZIONI

3.3.1 - DIAGRAMMA DELLE ATTIVITÀ

Il diagramma delle attività, in UML, è un diagramma di Flusso (con alcuni elementi aggiuntivi) che mostra una sequenza di attività. Viene utilizzato per rappresentare i passi (le transazioni) che compongono il flusso di un caso d'uso, descrivono quindi il comportamento dinamico di un sistema. La rappresentazione delle attività è comoda in quanto consente di rappresentare sinteticamente flusso principale e flussi alternativi. Il diagramma in figura 12 mostra il flusso delle attività dell'utente per visualizzare, creare o modificare partite.

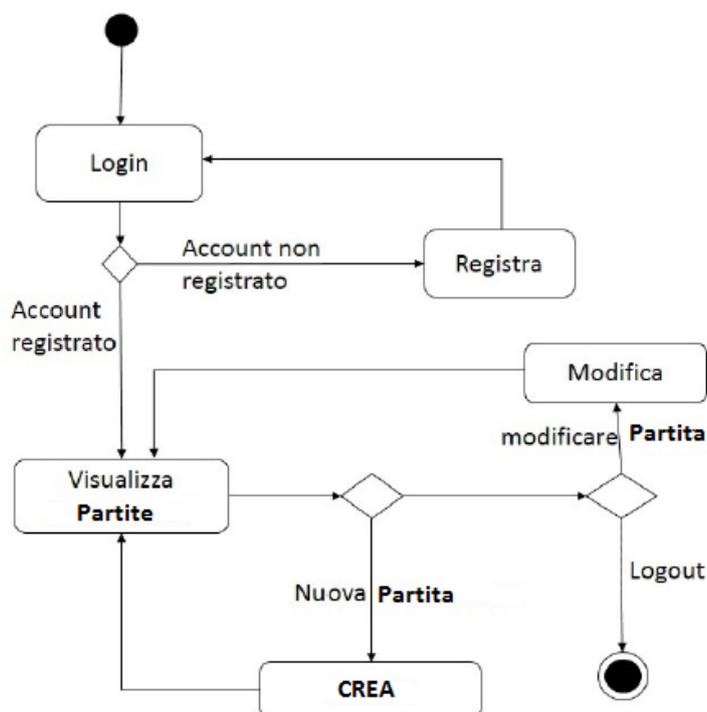


Figura 12 – Diagramma delle attività

3.4 - INTERFACCIA

L'interfaccia grafica in questo tipo di applicazioni è fondamentale che sia intuitiva e di facile utilizzo.

L'utente infatti deve essere invogliato all'utilizzo dell'applicazione e deve capire a colpo d'occhio dove trovare le funzionalità che cerca.

Per fare ciò è stato adottato il layout tipico delle applicazioni Android, che prevede una schermata iniziale dove sono presenti i collegamenti per richiamare le funzionalità principali e più utilizzate. È inoltre presente un menù laterale, richiamabile da ogni schermata dell'applicazione, per poter navigare velocemente tra le funzionalità, accedere al profilo personale ed effettuare il logout.

Di seguito si andrà ad illustrare il funzionamento del prodotto finito, con gli screenshot delle schermate più significative e brevi commenti.

3.4.1 – HOME PAGE

Trattandosi di una applicazione gestionale, una delle principali caratteristiche dovrà essere l'immediatezza e la facilità di utilizzo.

La schermata principale sarà quindi una Home page (Figura 13) contenente quattro pulsanti per la gestione delle quattro funzionalità principali dell'applicazione, ovvero la visualizzazione delle partite a cui si è registrati e la relativa gestione, il riepilogo del proprio profilo utente, l'elenco delle cerchie di amici a cui si è iscritti e l'elenco dei campi di gioco disponibili.



Figura 13 – HOME PAGE

Vi è inoltre un menù laterale (Figura 13.1) che permette una rapida navigazione tra tutte le funzionalità dell'applicazione e la possibilità di effettuare il logout. Il menù laterale è richiamabile da ogni schermata e in ogni momento.

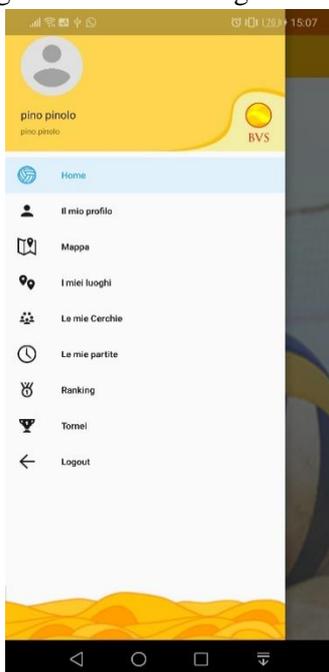


Figura 13.1 – Menù laterale

3.4.2 – LE MIE PARTITE

Questa schermata (Figura 13.2) presenta il riepilogo delle partite a cui il giocatore è attualmente iscritto, a quelle a cui ha partecipato e gli inviti a match futuri che ha ricevuto.

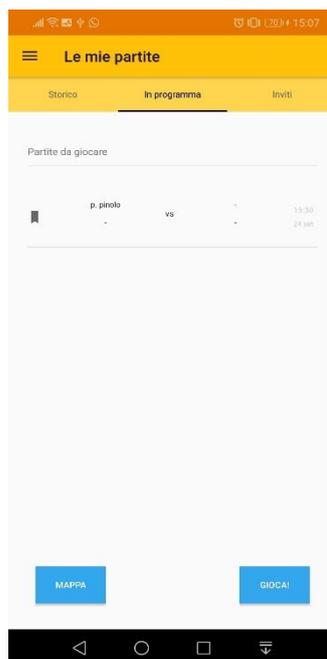


Figura 13.2 – Le mie partite

Sono presenti anche due bottoni, “Gioca!” e “Mappa”, il primo permette la creazione di una nuova partita, il secondo apre la mappa (Figura 13.3), dando la possibilità di visualizzare quali sono i campi da gioco registrati nell’applicazione vicini alla propria posizione.



Figura 13.3 – Mappa

Al tocco dell’utente sulla partita desiderata, verrà aperta la schermata di riepilogo (Figura 13.4) dove sarà possibile effettuare modifiche o invitare giocatori qualora si fosse l’organizzatore dell’incontro.



Figura 13.4 – Dettaglio partita

3.4.3 – IL MIO PROFILO

In questa schermata (Figura 13.5) sarà possibile visionare e modificare le proprie caratteristiche fisiche e di giocatore inserite al momento della registrazione. È inoltre presente un bottone per caricare una immagine dalla galleria del telefono da utilizzare come immagine profilo.

Nel tab “statistiche” è presente un riepilogo delle proprie statistiche delle partite giocate.

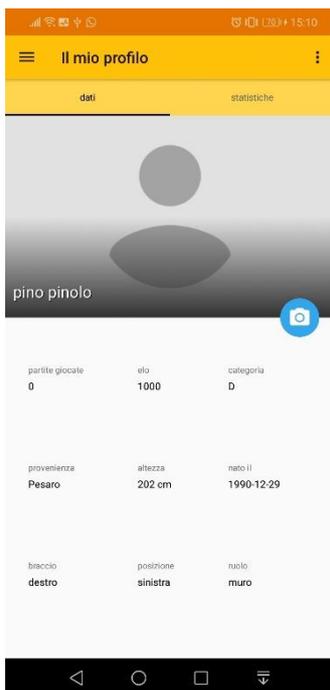


Figura 13.5 – Profilo

3.4.4 – LE MIE CERCHIE

Alla pressione del bottone “Le mie cerchie”, si aprirà la relativa schermata (Figura 13.6), presentando tutte le cerchie a cui l’utente è iscritto, e un bottone che permette la creazione di una nuova cerchia. Alla pressione di una cerchia si aprirà l’elenco dei giocatori partecipanti, e qualora si fosse il creatore, si avrà la possibilità di modificarla e espellere o invitare altri utenti.

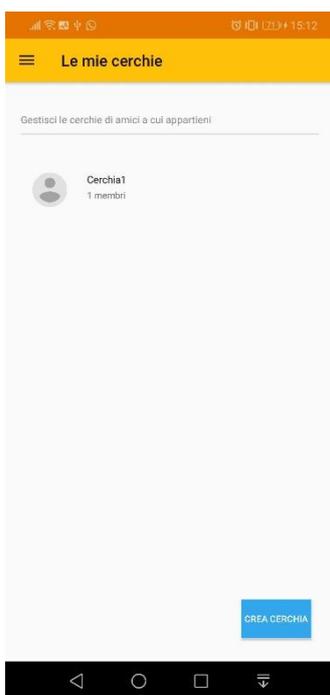


Figura 13.6 – Cerchie

3.4.5 – ELENCO CAMPI

La schermata (Figura 13.7) permette di visualizzare tutti i campi da gioco, con relativa immagine di anteprima se presente, registrati nell'applicazione. È inoltre presente la possibilità di filtrare la ricerca digitando il nome del campo o della città desiderata.

Alla pressione del campo si aprirà la schermata di riepilogo con tutte le informazioni ad esso relative. Sarà inoltre presente un bottone per creare una nuova partita su quel campo da gioco.



Figura 13.7 – Campi da gioco

CAPITOLO 4

IMPLEMENTAZIONE

4.1 – TECNOLOGIA UTILIZZATA

In questo capitolo saranno esplorate quelle che sono le principali tecnologie adottate per la creazione e la gestione di rilevanti funzionalità dell'applicazione.

4.1.1 MY SQL

Una base di dati (database) è un insieme di dati logicamente correlati fra loro. I Database Management System (DBMS) sono prodotti software in grado di gestire database che hanno grandi quantità di dati, che condividono i dati fra più utenti e applicazioni e che utilizzano dei sistemi di protezione e autorizzazione per l'accesso ai dati. Esistono diversi tipi di DBMS: gerarchico, reticolare, relazionale, ad oggetti; il modello che più si adatta alle mie esigenze è il modello relazionale che organizza i dati in tabelle, basandosi sulle relazioni fra essi.

MySQL è un sistema di gestione di basi di dati relazionali multi-piattaforma MySQL si occupa della strutturazione e della gestione a basso livello dei dati stessi, in modo da velocizzarne l'accesso, la modifica e l'inserimento di nuovi elementi.

In questo progetto è stato utilizzato un DB per mantenere organizzati ed aggiornati i dati, permettendo di ridurre di molto lo spazio fisico occupato dall'applicazione nel dispositivo e di avere sempre dati aggiornati in tempo reale.

4.1.2 XML

XML, (“*eXtensible Markup Language*”), è un linguaggio estensibile realizzato per poter utilizzare in modo semplice i documenti strutturati, studiato per il Web e per superare i limiti di HTML (“*HyperText Markup Language*”), ma con possibilità di utilizzo in ambienti differenti.

Sviluppato dal W3C, il *World Wide Web Consortium*, XML è un sottoinsieme di SGML (“*Standard Generalized Markup Language*”), uno standard internazionale che definisce le regole per scrivere “*markup language*”, volutamente non comprende alcune funzionalità complesse di SGML difficilmente implementabili su Web.

Un linguaggio di markup è composto di istruzioni, è composto di istruzioni, definite tag o marcatori, che descrivono la struttura e la forma di un documento. Ogni marcatore (o coppia di marcatori) identifica un elemento o componente del documento. I marcatori vengono inseriti all'interno del documento e sia il testo, sia i

marcatori sono memorizzati in formato ASCII. XML usa la codifica dei caratteri UNICODE, un documento XML è un documento XML è “leggibile” da un utente umano da un utente umano senza la mediazione di software specifico.

È stato utilizzato per costruire l’interfaccia grafica dell’applicazione, a partire dal layout delle varie schermate fino all’inserimento delle immagini per lo sfondo ed i bottoni.

4.1.3 FIREBASE

Firebase è un potente servizio on line che permette di salvare e sincronizzare i dati elaborati da applicazioni web e mobile. Si tratta di un “*database NoSQL*” dalle grandissime risorse, ad alta disponibilità ed integrabile in tempi rapidissimi in altri progetti software, semplicemente sottoscrivendo un account al servizio. Firebase rientra in una categoria di servizi on-line in rapidissima diffusione e noti come “backend”. Le caratteristiche principali di Firebase sono: la capacità di sincronizzazione dei dati oltre che di storage, infatti Firebase è in grado di aggiornare i dati istantaneamente, sia se integrato in app web che mobile; la disponibilità di librerie client per integrare Firebase in ogni app. Android, Javascript e framework con esso realizzati, Java e sistemi Apple; sicurezza, infatti i dati immagazzinati in Firebase sono replicati e sottoposti a backup continuamente. La comunicazione con i client avviene sempre in modalità crittografata tramite SSL con certificati a 2048-bit.

Per lo sviluppo di questo progetto è stato scelto di utilizzare anche Firebase per la gestione di tutti i dati dinamici che avevano bisogno di un DB più “reattivo”, infatti Firebase si occupa della gestione delle partite in corso e dell’invio delle notifiche ai vari utenti interessati.

CAPITOLO 5

VALUTAZIONE DEI RISULTATI

Questo capitolo presenta le varie modalità di testing ed analisi che sono state utilizzate per valutare l'applicazione.

Seguiranno poi i feedback di alcuni utenti ai quali è stato chiesto di provare l'applicazione, per avere un riscontro sull'usabilità della stessa.

5.1 – TEST

Il testing rappresenta una delle attività più importanti per assicurare la qualità del software. Dopo aver generato il codice sorgente, si deve collaudare il software per scoprire (e correggere) quanti più errori possibili prima di rilasciare il prodotto.

Quando il software è stato realizzato, spesso è necessario apportare delle modifiche, sia perché il software non risulta rispettare completamente le specifiche fissate, sia perché si ritiene necessario perfezionare alcune funzionalità e caratteristiche. Si possono fare piccole modifiche oppure, se siamo molto lontani dai requisiti, si può scegliere di fare la reingegnerizzazione del sistema, in pratica ripartire da capo.

Finita con successo la fase di manutenzione si può eseguire il rilascio del sistema.

I test sono stati condotti simulando tanti possibili scenari d'uso propri e impropri al fine di testarne la robustezza funzionale.

Gli aspetti dell'applicazione su cui sono stati effettuati il maggior numero di test riguardano principalmente:

- Database: sono stati effettuati tutti i test di inserimento, interrogazione e cancellazione sul database (query) per verificarne il corretto funzionamento;
- Sicurezza: inserimento di dati validi nei rispettivi campi di input, gestione in sicurezza dei dati memorizzati nel database;
- Usabilità: sono stati effettuati test sull'intuitività e sulla facilità di utilizzo dell'applicazione, facendola utilizzare da utenti completamente inesperti.

5.2 – USER EXPERIENCE

Data la natura del progetto, il livello di usabilità dell'applicazione rappresenta un aspetto di estrema rilevanza. La facilità di utilizzo e l'immediata comprensione da parte dell'utente delle principali attività sono aspetti che concorrono a determinare la buona riuscita del progetto al pari delle qualità prestazionali.

Per riuscire a valutare correttamente l'usabilità dell'intero sistema, è stata svolta un'attività di testing; è stata quindi sottoposta l'applicazione all'utilizzo da parte di 10 utenti finali con diversi livelli di conoscenza del sistema Android, ai quali è stato anche proposto un breve questionario organizzato per punti, presentato in Figura 14

In questo modo è stato possibile analizzare nello specifico quali sono i principali passaggi che possono risultare di non facile comprensione all'utilizzatore finale, e avere delle indicazioni su quali problematiche si possono incontrare nel normale utilizzo dell'applicazione.

I risultati dei questionari sono stati elaborati e le considerazioni sono presentate in base alle sezioni del questionario.

- *Generale*: nei suoi aspetti più generali l'applicazione è stata ben valutata, in quanto presenta sempre nelle diverse schermate titoli esplicativi, e i contenuti sono strutturati secondo una logica semplice, ordinata e di facile utilizzo.
- *Navigazione/interfaccia*: anche questi due aspetti hanno ricevuto pieni voti. Grazie in particolar modo all'uso di bottoni semplici ed intuitivi, l'utente non si trova mai spiazzato o disorientato, riuscendo così ad effettuare in maniera rapida le operazioni desiderate.
È stato particolarmente apprezzato il menù laterale che permette una navigazione rapida tra le funzionalità.
- *Funzionalità*: le funzionalità offerte dall'applicazione sono state valutate positivamente, anche il livello di personalizzazione del proprio profilo è risultato adeguato.

Gli utenti meno esperti hanno trovato qualche difficoltà nell'uso della mappa per la selezione del campo da gioco per creare una nuova partita o visualizzare le partite disponibili.

Un utente ha inoltre valutato i tempi di esecuzione leggermente troppo lunghi.

- *Aspetti migliorabili:* da questa sezione è emerso che l'aspetto che più necessita di miglioramenti è l'aggiunta di nuove funzionalità, nonché la miglioria di quelle presenti. Uno degli aspetti più apprezzati è risultato essere l'interfaccia e la facilità di utilizzo e di navigazione.

**QUESTIONARIO DI VALUTAZIONE APPLICAZIONE
THE BEACH VOLLEY SOCIETY**

Valuta le seguenti domande con un punteggio da 0 a 3, dove :

0	Assolutamente no
1	Più no che si
2	Più si che no
3	Assolutamente si

Valutazione dell'applicazione				
Generale				
1. I contenuti sono strutturati in maniera semplice ed intuitiva?	0	1	2	3
2. Il testo è facilmente leggibile?	0	1	2	3
3. Le diverse sezioni sono dotate di un titolo significativo che illustra il tipo di informazione visualizzata?	0	1	2	3
Navigazione				
1. Tutte le pagine presentano opzioni di navigazione?	0	1	2	3
2. Sono presenti funzionalità di annullamento delle operazioni?	0	1	2	3
3. Sono presenti funzionalità per ritornare alla home page?	0	1	2	3
4. Il menù di navigazione è efficace?	0	1	2	3
Interfaccia				
1. L'applicazione ha un aspetto gradevole?	0	1	2	3
2. Lo stile grafico è adeguato a rappresentare le funzionalità delle diverse schermate?	0	1	2	3
3. Ci sono situazioni in cui la navigazione è facilitata dalla grafica?	0	1	2	3
4. Ci sono situazioni in cui la navigazione è compromessa dalla grafica?	0	1	2	3
Funzionalità				
1. I tempi di esecuzione delle funzionalità sono ragionevoli?	0	1	2	3
2. Il livello di personalizzazione del proprio profilo giocatore è adeguato?	0	1	2	3
3. Il sistema di ranking è equo?	0	1	2	3
4. Le partite sono di facile gestione?	0	1	2	3
Cosa pensi sia possibile o necessario migliorare?				
1. Intefaccia grafica o layout	0	1	2	3
2. Menù di navigazione	0	1	2	3
3. Funzionalità dell'applicazione	0	1	2	3
4. Organizzazione dei contenuti	0	1	2	3
Di seguito puoi scrivere le tue considerazioni personali				

Figura 14 – Questionario di valutazione

Analizzando infine le diverse considerazioni personali è possibile concludere la fase di test dell'usabilità affermando che in generale l'applicazione è ben strutturata e di facile utilizzo, secondo la quasi totalità dei tester.

CAPITOLO 6

CONCLUSIONI

L'obiettivo di questa tesi è stato l'implementazione di funzionalità aggiuntive e la migloria dell'interfaccia grafica e di navigazione di un'applicazione per sistemi mobile Android. Scopo dell'applicazione è quello di favorire l'organizzazione di partite di beach volley e di creare una community con la quale ritrovarsi per giocare in spiaggia e in strutture attrezzate.

Per poter sviluppare al meglio quanto necessario, è stata indispensabile una prima parte di scoperta delle funzionalità dell'applicazione per poterne comprendere il funzionamento e le finalità. Successivamente si è passati ad una analisi più approfondita del codice e della base dati, in modo da capire quali fossero le parti su cui intervenire e come poter integrare in modo coerente il codice aggiuntivo.

È stata implementata la possibilità di effettuare il logout dall'applicazione, indispensabile per utilizzare più account sullo stesso dispositivo (es. dispositivi condivisi da più utenti o singolo utente con un account professionale e uno privato), è stato inoltre aggiunto un menù dal quale è possibile visualizzare (e filtrare) un elenco di tutti i campi da gioco disponibili, mentre prima era possibile cercarli solo tramite la mappa. Questo ha snellito l'operazione di creazione di una nuova partita e di ricerca dei campi da gioco disponibili, rendendo l'applicazione di più facile utilizzo anche per utenti che hanno meno dimestichezza con l'uso della mappa su Android.

Nelle fasi di progettazione ed implementazione è stata posta una particolare attenzione alla semplicità d'uso dell'applicazione e alla sua interfaccia grafica, in modo da renderla il più intuitiva possibile e che fosse stilisticamente coerente tra le varie schermate. L'obiettivo era quello di renderla il più possibile *user friendly* e che potesse essere di facile utilizzo anche da utenti meno esperti.

Le principali difficoltà che si sono presentate e che hanno poi condizionato la realizzazione del progetto, sono state relative all'integrazione con il codice già presente. Non avendo appunto la possibilità di modificare in maniera significativa quanto già scritto, senza che questo impattasse sull'integrità del sistema, è stato necessario adattare il nuovo codice, a volte stravolgendo completamente quanto pensato nella fase di analisi.

La realizzazione di questo progetto è stata un'esperienza interessante, in quanto, seppur in scala ridotta, mi ha dato la possibilità di scoprire un linguaggio di programmazione a me nuovo e di prendere dimestichezza con un ambiente di sviluppo molto utilizzato per lo sviluppo di applicazioni Android. Mi ha inoltre dato l'occasione di testare le mie capacità di analisi e di risoluzione dei problemi, competenze fondamentali per il lavoro di programmatore che attualmente svolgo.

6.1 - SVILUPPI FUTURI

Come emerso dalla fase di analisi dell'usabilità, questo progetto potrebbe essere ampliato e modificato in diverse sue parti.

Innanzitutto, in futuro sarebbe utile rivedere il codice, ottimizzandolo in modo da evitare codice duplicato o utilizzando funzioni ad hoc che permettano di snellirlo e renderlo più leggibile e performante.

Inoltre, anche la base dati potrebbe essere rivista, ottimizzare le tabelle e i collegamenti, in modo da avere una gestione dati più solida e di rapida consultazione. Oltre alla gestione dei dati e alla riorganizzazione del codice, possibili sviluppi futuri comprendono anche l'implementazione di nuove funzionalità.

In particolare, sarebbe utile poter introdurre uno *spogliatoio*, o una sorta di lobby prepartita, in cui i giocatori che partecipano hanno la possibilità di chattare senza doversi scambiare i numeri di telefono o essere in lista amici. Questo favorirebbe l'organizzazione e il confronto con gli altri giocatori (es. per avvisare un ritardo dovuto da un imprevisto o proporre una modifica alle modalità della partita).

Interessante potrebbe essere la possibilità di poter dare una valutazione agli altri giocatori al termine della partita, in modo da avere, oltre al ranking, un altro metro di giudizio che sia più personale piuttosto che sulle competenze sportive (es. non sono un bravissimo giocatore, ma ho valutazioni positive in sportività e spirito di squadra).

Sarebbe inoltre utile aggiungere l'integrazione con google calendar, in modo da avere sempre sott'occhio le proprie partite pianificate e non perdersi mai un match tra la frenesia della vita moderna.

BIBLIOGRAFIA

- [1] [<http://www.ultimouomo.com/la-vita-con-fubles/>]
- [2] [<https://play.google.com/store/apps/details?id=it.android.fubles>]
- [3] [<https://play.google.com/store/apps/details?id=io.ionic.sportscomrade>]
- [4] [<http://www.agoravox.it/Enjore-la-startup-dei-tornei.html>]
- [5] [<https://play.google.com/store/apps/details?id=com.enjore>]
- [6] [<https://developer.android.com/about/dashboards/index.html>]