

ALMA MATER STUDIORUM · UNIVERSITÀ DI BOLOGNA

---

SCUOLA DI SCIENZE  
Corso di Laurea in Informatica per il management

**SVILUPPO DI UN'APPLICAZIONE  
IN AMBITO FIERISTICO :  
TECNOLOGIE DI MAPPATURA  
E NAVIGAZIONE INDOOR**

**Relatore:**  
Chiar.mo Prof.  
Luciano Bononi

**Presentata da:**  
Giovanni Londei

**Correlatore Aziendale :**  
Dott.ssa  
Yvonne Schmidt

**I Sessione**  
**Anno Accademico 2016/2017**

*Ai miei Genitori,  
per avermi permesso di realizzare tutto questo  
e per aver sempre creduto in me ...*



# Introduzione

Questa tesi descrive la mia partecipazione al progetto "Blick's Eye" commissionato dall'azienda "Blick Desing S.R.L.". Il progetto è stato realizzato in collaborazione con lo studente Gabriele Carlucci durante lo svolgimento delle attività di tirocinio curricolare e tirocinio per tesi in azienda.

Lo scopo del progetto è la realizzazione di una applicazione in grado di gestire le informazioni relative ad un evento fieristico e che fornisca all'utente funzionalità utili a migliorare l'esperienza in fiera. Lo scopo finale di "Blick's Eye" è la realizzazione di una applicazione di base che non sia vincolata ad un singolo evento fieristico ma che definisca la struttura per la realizzazione di applicazioni specifiche per le singole fiere. Tali app devono poter fornire all'utente tutte le informazioni di base riguardanti la manifestazione di interesse come data di inizio e fine , orari di apertura e chiusura , luogo dell'evento e prezzi dei biglietti, devono essere inoltre in grado di gestire le informazioni riguardanti gli stand , gli espositori , i settori merceologici e gli eventi rendendole facilmente accessibili agli utenti .

Spesso soprattutto in eventi fieristici di grandi dimensioni è difficile per l'utente orientarsi e trovare facilmente ciò che gli interessa , il progetto "Blick's Eye" vuole implementare dunque un sistema di navigazione che permetta di muoversi facilmente all'interno dell'ambiente che ospita la manifestazione. Quest'ultima funzione rappresenta l'innovazione principale che il progetto vuole realizzare e allo stesso tempo l'ostacolo più arduo da superare . L'utilizzo della tecnologia GPS per la geolocalizzazione e la navigazione in ambienti outdoor è uno standard ed esistono svariate applicazione differenti tra

loro che permettono agli utenti di sfruttare tali servizi, tuttavia la maggior parte degli eventi fieristici si svolge all'interno di ambienti chiusi dove questa tecnologia non funziona ed occorre dunque utilizzare altre metodologie per raggiungere tale scopo.

Sono tre le problematiche principali da risolvere per creare un sistema che simuli il funzionamento di un navigatore GPS ma in ambienti indoor :

**Geolocalizzazione dell'Utente** Come detto in precedenza non è possibile calcolare la posizione dell'utente in un ambiente chiuso attraverso l'utilizzo della tecnologia GPS , occorre dunque utilizzare altri tipi di tecnologia per risolvere tale problema.

**Mappatura dell'ambiente** Se per le applicazioni outdoor esistono numerose versioni di mappe fornite dalle immagini satellitari stesso non vale per gli ambienti indoor. Per permettere agli utenti di muoversi all'interno di un ambiente chiuso occorre realizzare una mappa che ne rappresenti le caratteristiche fornendo informazioni sulla suddivisione dei locali , sui punti di ingresso ed uscita , sui punti di accesso ai piani superiori o inferiori e su eventuali punti di interesse.

**Sistema di Navigazione** E' necessario definire un sistema che permetta di dare all'utente finale informazioni su come muoversi nell'ambiente , questo sistema deve tener conto dei punti di ingresso e di uscita , dei punti di accesso ai piani inferiori e superiori e di eventuali ostacoli o impedimenti

Sono state analizzate le principali tecnologie adottabili per risolvere tali problematiche, io mi sono occupato principalmente dell'aspetto relativo alla mappatura dell'ambiente e alla navigazione valutando diverse opzioni di realizzazione in base alle limitazioni imposte dal nostro caso d'uso. Dopo aver valutato diverse alternative è stato progettato un sistema di mappatura e navigazione adatto alle caratteristiche degli ambienti fieristici e alle richieste del committente. E' stata quindi realizzata una prima versione dell'app con le principali funzionalità richieste .

## Capitoli della tesi

Il primo capitolo della tesi è lo stato dell'arte nel quale vengono descritti i principali servizi di Geolocalizzazione e Navigazione sia in ambito outdoor che indoor.

Nel secondo capitolo viene descritto nel dettaglio l'obiettivo del progetto "Blick's Eye". Vengono elencate le specifiche tecniche e di funzionamento fornite dall'azienda "Blick Design S.R.L" e analizzate le principali problematiche da affrontare.

Il terzo capitolo dà una panoramica sulle principali tecniche per la mappatura di ambienti indoor analizzandole in base alle limitazioni imposte dal progetto. Viene descritto poi nel dettaglio il sistema che si è scelto di utilizzare per effettuare la mappatura in ambienti fieristici e vengono elencate le motivazioni che hanno portato ad effettuare tali scelte.

Nel quarto capitolo vengono elencate ed analizzate diverse metodologie per la realizzazione di sistemi di navigazione indoor. Vengono poi valutate tutte le alternative in base ai vincoli di realizzazione imposti dal caso d'uso. Si descrive inoltre nel dettaglio il sistema di navigazione progettato per la realizzazione dell'app analizzando le motivazioni che hanno portato ad effettuare tali scelte.

Nel quinto capitolo si descrive lo sviluppo della prima versione dell'applicazione. Vengono inizialmente elencate le principali tecnologie utilizzate per poi descrivere nel dettaglio le varie funzionalità implementate concentrandosi sulle classi e i metodi utilizzati.

Il sesto ed ultimo capitolo sono le conclusioni. In questo capitolo si riassume in breve il lavoro svolto per la realizzazione di questo progetto, vengono analizzate le alternative valutate per la risoluzione delle diverse problematiche e motivate le scelte effettuate. Si parla inoltre dei possibili sviluppi futuri del progetto elencando le principali funzionalità da integrare.



# Indice

<b>Introduzione</b>	<b>i</b>
<b>1 Stato dell'Arte</b>	<b>3</b>
1.1 Servizi di Navigazione e Geolocalizzazione Outdoor per Dispositivi Mobili . . . . .	3
1.1.1 Google Maps . . . . .	3
1.1.2 Apple Maps . . . . .	4
1.1.3 Here WeGo . . . . .	5
1.2 Servizi di Navigazione e Geolocalizzazione Indoor per Dispositivi Mobili . . . . .	5
1.2.1 Indoor Navigation Indoor Positioning by infsoft . . . . .	5
1.2.2 MapsPeople . . . . .	6
1.2.3 Inmaps . . . . .	7
<b>2 Progetto "Blick Eye"</b>	<b>9</b>
2.1 Obiettivo . . . . .	9
2.2 Specifiche . . . . .	10
2.2.1 Specifiche Tecniche . . . . .	10
2.2.2 Specifiche di Funzionamento . . . . .	12
2.3 Problematiche e Limiti di Realizzazione . . . . .	15
<b>3 Mappatura di Ambienti Indoor</b>	<b>17</b>
3.1 Problema . . . . .	17
3.2 Alternative . . . . .	18



---

3.2.1	IndoorAtlas . . . . .	18
3.2.2	Google Tango . . . . .	23
3.3	Scelte di Implementazione . . . . .	28
3.3.1	Soluzione . . . . .	28
3.3.2	Georeferenziare una Immagine con JOSM (Java Open- StreetMap Editor) . . . . .	29
3.3.3	Sovrapporre una Immagine ad una Mappa . . . . .	32
3.4	Conclusioni Capitolo . . . . .	35
<b>4</b>	<b>Tecnologie di Navigazione Indoor</b>	<b>39</b>
4.1	Problema . . . . .	39
4.2	Alternative . . . . .	40
4.2.1	Navigazione Inerziale . . . . .	40
4.2.2	Navigazione tramite Database Fotografico . . . . .	43
4.2.3	Navigazione tramite realtà Aumentata . . . . .	45
4.3	Scelte di Implementazione . . . . .	47
4.3.1	Soluzione . . . . .	47
4.3.2	Progettazione e Creazione della Griglia di Navigazione	48
4.3.3	Algoritmo di Navigazione Indoor . . . . .	50
4.4	Navigazione tra diversi Padiglioni . . . . .	52
4.5	Conclusioni Capitolo . . . . .	52
<b>5</b>	<b>Sviluppo dell'App</b>	<b>55</b>
5.1	Tecnologie Utilizzate . . . . .	55
5.2	Struttura dell'App . . . . .	57
5.2.1	Struttura Generale . . . . .	57
5.2.2	Home . . . . .	58
5.2.3	Navigazione Outdoor . . . . .	59
5.2.4	Navigazione Indoor . . . . .	61
5.2.5	Espositori . . . . .	68
5.2.6	Settori Merceologici . . . . .	70
5.2.7	Eventi . . . . .	71

<b>6 Conclusioni</b>	<b>73</b>
6.1 Sviluppi Futuri . . . . .	75
6.2 Opinioni Personali . . . . .	76
<b>Bibliografia</b>	<b>77</b>



# Elenco delle figure

2.1	Homepage da Specifiche . . . . .	13
2.2	Mappa Indoor da Specifiche . . . . .	15
3.1	Logo IndoorAtlas . . . . .	19
3.2	Funzionamento IndoorAtlas . . . . .	19
3.3	Esempio di utilizzo MapCreator . . . . .	21
3.4	Esempio di Calibrazione MapCreator . . . . .	21
3.5	Logo Project Tango . . . . .	24
3.6	Funzionamento Motion Tracking,Area Learning e Depth Precision . . . . .	26
3.7	Funzionamento Videocamera Lenovo Phab 2 pro . . . . .	27
3.8	Mappatura degli Stand con Josm . . . . .	32
4.1	Mappatura DirectionPoints con Josm . . . . .	50
4.2	Mappatura DoorPoints con JOSM . . . . .	50
5.1	Main.Storyboard App . . . . .	57
5.2	EventsTableView . . . . .	59
5.3	EventDetailView . . . . .	59
5.4	Funzione "Come Arrivare" . . . . .	60
5.5	Percorso rappresentato su Maps di Apple . . . . .	60
5.6	Esempio mappa degli stand . . . . .	62
5.7	Esempio di Funzionamento del metodo CalcPath . . . . .	66
5.8	Esempio di Funzionamento del metodo CalculateDirectionPath . . . . .	66

5.9	Seleziona Punto di Partenza . . . . .	67
5.10	Mappatura DoorPoints con JOSM . . . . .	67
5.11	ExpoTableView . . . . .	69
5.12	ExpoDetailView . . . . .	69
5.13	SectorTableView . . . . .	71
5.14	EventsTableView . . . . .	72
5.15	EventDetailView . . . . .	72



# Capitolo 1

## Stato dell'Arte

Si analizzano nello stato dell'arte i principali servizi per la geolocalizzazione e la navigazione sia in ambienti outdoor che indoor.

### 1.1 Servizi di Navigazione e Geolocalizzazione Outdoor per Dispositivi Mobili

#### 1.1.1 Google Maps

Servizio Web creato da Google che consente la visualizzazione e la ricerca di mappe geografiche della quasi totalità della Terra. Tale servizio offre la possibilità di ricercare qualsiasi località si desideri inoltre consente la ricerca di punti di interesse quali ristoranti, negozi, ospedali eccetera. Le mappe possono essere visualizzate anche attraverso immagini satellitari fino ad un certo livello di zoom, tramite questa modalità si riescono a riconoscere dettagli specifici di ogni luogo. Google maps permette anche di calcolare percorsi di navigazione per collegare 2 località differenti, consentendo all'utente anche di scegliere il mezzo di trasporto. Le diverse possibilità di scelta sono: muoversi a piedi, tramite automobile, tramite i servizi pubblici, tramite taxi o anche in aereo. In caso di navigazione tramite automobile Google Maps è dotato di un servizio di valutazione del traffico molto preciso che

permette di visualizzare in anticipo l'eventuale presenza di code o incidenti stradali. Google Maps offre anche diversi servizi collegati con esso come ad esempio Google StreetView servizio che permette di visualizzare fotografie a 360 di diverse località nel mondo , o come Google Places che permette di indicare un'attività commerciale in una determinata locazione. Questo Servizio utilizza la tecnologia GPS per la geolocalizzazione dell'utente e per la navigazione, necessita di una connessione internet per il corretto funzionamento . Questo servizio è disponibile su tutti i dispositivi mobili tramite l'apposita app , inoltre sono disponibili pacchetti di sviluppi per i principali linguaggi di programmazione che permettono di integrare le mappe di Google in qualsiasi tipo di programma o applicazione . Google Maps è sicuramente il servizio mappe più utilizzato in particolare nei dispositivi mobili.

### 1.1.2 Apple Maps

Apple Maps è un servizio Mappe creato da Apple per essere il sistema di navigazione e geolocalizzazione di default dei dispositivi iOS, macOS, e watchOS. Questo servizio permette la visualizzazione di mappe geografiche comprensive di quasi la totalità della Terra, possono essere visualizzate anche in formato satellitare e in formato 3D. Apple Maps permette la ricerca di punti di interesse quali ristoranti, bar, negozi eccetera , inoltre permette di calcolare percorsi di navigazione scegliendo il mezzo di trasporto che si vuole utilizzare. Il servizio offre alcune alternative per il calcolo del percorso che comprendono la possibilità di muoversi a piedi, in automobile o tramite servizi pubblici , tuttavia le informazioni sui servizi pubblici sono limitati ad un numero ristretto di città. Il servizio utilizza la tecnologia gps per la geolocalizzazione dell'utente e per la navigazione, necessita di una connessione internet per il corretto funzionamento. Questo servizio non è disponibile per dispositivi mobili che non sono Apple.



### 1.1.3 Here WeGo

Here WeGo [1] è una piattaforma di geolocalizzazione e navigazione outdoor che permette l'utilizzo di tali servizi anche offline, è disponibile un app per utilizzare questa piattaforma attraverso dispositivi mobili sia iOS che Android. Questa app permette di scaricare delle mappe relative ad una determinata area geografica ed utilizzarle anche quando si è offline. Here WeGo permette la ricerca di punti di interesse quali ristoranti, centri commerciali, negozi, permette inoltre di calcolare percorsi di navigazione tra diversi luoghi scegliendo il mezzo di trasporto tra: automobile, mezzi pubblici o a piedi. In caso di navigazione tramite automobile tale piattaforma mette a disposizione un servizio di valutazione del traffico.

## 1.2 Servizi di Navigazione e Geolocalizzazione Indoor per Dispositivi Mobili

### 1.2.1 Indoor Navigation Indoor Positioning by infsoft

L'azienda Infsoft mette a disposizione una piattaforma per la geolocalizzazione e la navigazione in ambienti indoor.

#### Indoor Positioning [2]

Per quanto riguarda il sistema di posizionamento questa piattaforma mette a disposizione 2 possibilità:

**Client-Based Positioning** Con sistema di posizionamento client-based si intende che la geolocalizzazione avviene direttamente tramite il dispositivo utilizzato dall'utente, per fare questo la piattaforma prevede la possibilità di utilizzare metodi basati sulla triangolazione Wi-Fi o sull'utilizzo di dispositivi Bluetooth Low Energy ovvero di Beacon

**Server-Based Positioning** Con sistema di posizionamento server-based si intende che la posizione di ogni utente viene calcolata tramite un server

. Per fare questo si utilizzando metodologie basate sulla triangolazione Wi-fi, sull'utilizzo di Beacon , sull'utilizzo della tecnologia UltraWideBand o sull'utilizzo di radiofrequenze. Per questo tipo di sistema occorrono necessariamente installazioni hardware.

Queste diverse tecnologie possono anche coesistere per migliorare la precisione del metodo di posizionamento, sia se si utilizza un sistema client-base sia che si utilizzi un sistema server-based.

### **Indoor Navigation [3]**

Per quanto riguarda la navigazione Infsoft mette a disposizione un sistema basato sul tipo di tecnologia di posizionamento scelto. Solitamente per applicazione che necessitano di un navigatore in ambienti indoor si utilizza una tecnologia client-base basata sull'utilizzo di Beacon, Triangolazione Wifi o tramite l'utilizzo di hardware aggiuntivi che permettono di utilizzare la tecnologia UltraWideBand.

#### **Come si utilizza?**

Infsoft mette a disposizione un SDK sia per Android che per iOS , quindi è possibile utilizzare questa piattaforma sulla maggior parte dei dispositivi mobili.

### **1.2.2 MapsPeople**

MapsPeople [4] è una piattaforma per la navigazione in ambienti indoor che sfrutta le mappe e tutte le funzionalità di Google Maps. Per quanto riguarda il sistema di posizionamento può sfruttare le principali tecnologie di geolocalizzazione indoor come Beacon , triangolazione Wi-Fi o sistemi basati sull'utilizzo di radiofrequenze. Il sistema di navigazione di MapsPeople equivale al sistema di navigazione fornito da Google Maps in ambienti outdoor, è in grado dunque di rappresentare i percorsi sulla mappa e di stimare il tempo di percorrenza. MapsPeople offre la possibilità di utilizzare

un programma chiamato MapsIndoor CMS (Content Management System) che permette all'utente di gestire graficamente la mappatura dell'ambiente indoor sul quale viene poi sviluppato il sistema di localizzazione e navigazione, andando ad inserire punti di interesse con le relative informazioni o indicazioni sui percorsi che possono essere utilizzati. Maps People fornisce un SDK per applicazioni Android, iOS e per servizi Web.

### 1.2.3 Inmaps

Inmaps [5] è una start-up italiana che offre servizi di geolocalizzazione e navigazione in ambienti indoor. La tecnologia di posizionamento utilizzata si basa sull'utilizzo di dispositivi "Bluetooth Low Energy" ovvero Beacon. Inmaps inoltre offre anche la possibilità di gestire ulteriori funzionalità come le notifiche di prossimità, ovvero la possibilità di avere informazioni su ciò che è vicino all'utente, o di fare analisi basandosi sulle informazioni fornite dai Beacon. Questo sistema è adatto per la creazione di applicazioni di navigazione all'interno di edifici che possono essere: centri commerciali, musei, ospedali o anche eventi fieristici. L'azienda inmaps offre diversi pacchetti a pagamento per sviluppare applicazioni per dispositivi mobili comprensivi di SDK per Android ed iOS.



# Capitolo 2

## Progetto "Blick Eye"

### 2.1 Obiettivo

L'Idea di base del progetto "Blick Eye" é la realizzazione di una serie di applicazioni per dispositivi mobili (iOS e Android) in ambito fieristico. Tale applicazioni sono rivolte in particolare al pubblico delle singole fiere , le funzioni devono dunque migliorare l'esperienza dell'utente nell'ambito dell'evento fieristico . L'App deve essere in grado di fornire tutte le informazioni principali riguardanti una fiera come : settori merceologici , espositori, punti di interesse ed eventi; inoltre deve mettere a disposizione un sistema di navigazione che permetta all'utente di muoversi agevolmente all'interno degli ambienti che ospitano la manifestazione. Tale sistema di navigazione indoor ha lo scopo di rendere estremamente facile per il visitatore della fiera raggiungere i punti di interesse e agevolare quindi la ricerca di settori, espositori o eventi . La richiesta che l'azienda "BlickDesign S.R.L." ha fornito é stata quella di realizzare una applicazione di base attraverso la quale andare a modellare successivamente applicazioni specifiche che riguardino singoli eventi fieristici , questo ha richiesto implicitamente la creazione di un sistema efficiente per modellare di volta in volta le app senza andare incontro a costi e tempi eccessivi.

## 2.2 Specifiche

### 2.2.1 Specifiche Tecniche

L'azienda "BlickDesign S.R.L." ha richiesto a me ed al mio collega Gabriele Carlucci la realizzazione di un'applicazione in ambiente Android ed iOS che fornisca all'utente finale tutte le informazioni riguardanti un evento fieristico e che permetta l'utilizzo di diverse funzionalità utili alla visita di tale manifestazione. Le informazioni richieste all'interno dell'app sono : data di inizio e di fine fiera, orari di apertura e chiusura, luogo in cui si svolge , costo dei biglietti, mappa degli stand, informazioni sugli espositori, informazioni sui settori merceologici ed informazioni sugli eventi. In qualunque evento fieristico ogni espositore può trattare diversi settori merceologici e possiede un determinato stand. La fiera può svolgersi all'interno di diversi padiglioni, ognuno di questi può contenere diversi stand. L'app deve essere in grado di gestire tutte le informazioni appena elencate e renderle facilmente accessibili all'Utente. Le funzionalità richieste all'interno dell'app sono le seguenti :

#### **Visualizzare le informazioni principali**

Funzionalità che permette all'utente finale di visualizzare tutte le informazioni principali della fiera come la data di inizio e di fine , gli orari di apertura e chiusura e il luogo nel quale si svolge la fiera.

#### **Come Arrivare**

Funzionalità che permette all'utente di sapere come raggiungere il luogo della fiera attraverso diversi mezzi di trasporto come : automobile, mezzi pubblici, bicicletta o eventualmente a piedi.

#### **Lista dei Settori**

Funzionalità che permette all'utente di visualizzare tutti i settori merceologici presenti alla fiera , essendo questi collegati con gli espositori è necessaria

inoltre una funzione che permetta di visualizzare una lista degli espositori che trattano un determinato settore merceologico.

### **Lista degli Espositori**

Funzionalità che permette all'utente di visualizzare tutti gli espositori presenti alla fiera e i relativi dettagli come : nome, stand , padiglione , numero di telefono , email e indirizzo della sede.

### **Lista degli Eventi**

Funzionalità che permette all'utente di visualizzare tutti gli eventi della fiera e i relativi dettagli come : titolo ,orario di inizio e fine e dove questo ha luogo.

### **Preferiti**

Funzionalità che permette all'utente di poter selezionare gli espositori preferiti, questi devono essere memorizzati e poter essere visualizzati in una lista accessibile rapidamente da qualsiasi schermata dell'app-

### **Mappa degli Stand**

Funzionalità che permette all'utente di visualizzare una mappa contenente i locali in cui si svolge l'evento fieristico con i relativi Stand mappati nel modo corretto . Ogni espositore possiede uno stand é quindi necessario poter ricavare dalla mappa le informazioni riguardanti quest'ultimi.

### **Navigazione Indoor**

Funzionalità che permette all'utente di calcolare un percorso per arrivare ad un punto di interesse partendo dalla sua posizione attuale o da una posizione selezionata. Il Punto di interesse deve poter essere scelto o come punto casuale all'interno della mappa o come stand cliccabile su di essa o andando a sceglierlo tramite la lista degli espositori.

### 2.2.2 Specifiche di Funzionamento

Il progetto "Blick's Eye" é pensato per creare una serie di applicazioni per dispositivi mobili realizzate appositamente per singoli eventi fieristici. Ogni singola app deve quindi trattare le informazioni e le funzionalità riguardanti una singola Fiera. L'applicazione da realizzare inizialmente deve essere un prototipo tramite il quale inserendo i dati riguardanti una specifica manifestazione fieristica se ne possa creare una specifica per il singolo evento. L'idea del progetto é che la creazione delle app specifiche non sia strettamente collegata al codice, ma che possa avvenire anche senza l'intervento di un programmatore o comunque andando a modificare una minima parte di codice. Si analizza ora come le diverse funzionalità principali devono funzionare secondo le richieste del committente :

#### Home

Tramite la homepage dell'applicazione l'utente deve poter facilmente accedere a tutte le funzionalità principali dell'app tramite un menú di navigazione posto ai piedi dello schermo. Nella Homepage deve essere presente un menú che permetta all'utilizzatore di accedere alle funzionalità di base cioè quelle che forniscono le informazioni principali sulla fiera e su come arrivare è prevista inoltre la presenza di banner pubblicitari.

#### Come Arrivare

La funzionalità "come arrivare" deve permettere all'utente di conoscere le diverse modalità per raggiungere il luogo della fiera. Deve essere rappresentato un percorso che dalla locazione dell'utente o da un punto da esso scelto arrivi fino alla locazione dell'evento fieristico, deve essere integrata la possibilità di visualizzare tale percorso e le relative indicazioni stradali tramite i principali servizi di navigazione per dispositivi mobili come ad esempio Google Maps o Maps di iOS, in modo da permettere all'utente di selezio-





Figura 2.1: Homepage da Specifiche

nare anche il mezzo di trasporto che preferisce e di utilizzare il GPS per raggiungere la meta.

### Gestione settori merceologici ed espositori

Settori merceologici ed espositori devono essere rappresentati all'interno dell'app attraverso delle liste. Essendo ogni settore merceologico collegato a diversi espositori, deve essere possibile attraverso la lista dei settori, cliccando su uno di essi, accedere ad una nuova lista di espositori che trattano il settore merceologico selezionato. Dalla lista degli espositori, cliccando su uno di essi deve essere visualizzata una nuova schermata con tutte le informazioni che lo riguardano cioè: nome, stand, padiglione, numero di telefono, email e indirizzo della sede. Nella pagina che rappresenta i dettagli degli espositori deve essere presente un pulsante che permetta di visualizzare l'espositore selezionato nella mappa degli stand.

### Gestione degli Eventi

Gli eventi devono essere rappresentati in una lista, cliccando su uno di essi deve essere visualizzata una nuova pagina con i dettagli dell'evento cioè: titolo, orario di inizio e fine e luogo in cui si svolge.

### **Mappa degli Stand e Navigazione Indoor**

La mappa degli stand deve fornire una rappresentazione veritiera della disposizione che essi hanno all'interno dei padiglioni , deve inoltre indicare dove si trovano i principali punti di interesse come servizi , punti ristoro , sale conferenze eccetera e mostrare precisamente quali sono i punti di ingresso e uscita di ogni padiglione. La mappa deve poter rappresentare eventualmente i diversi piani dei padiglioni in cui l'evento si svolge. Il sistema di navigazione interno alla fiera deve conoscere la locazione dell'utente , quest'ultimo deve poter selezionare il punto di destinazione in diversi modi :

- Selezionando un punto specifico sulla mappa
- Selezionando uno stand sulla mappa
- Selezionando un'espositore dalla lista degli espositori

Una volta selezionato il punto di destinazione, l'app deve calcolare il percorso piú conveniente per l'utente e rappresentarlo nella mappa degli stand. Il sistema di navigazione deve essere in grado di riconoscere l'area degli stand e creare un percorso che eviti di oltrepassarne i bordi, deve inoltre riconoscere quando il punto di partenza e di arrivo si trovano su padiglioni differenti ed essere in grado di costruire un percorso che tenga in considerazione i punti di ingresso ai vari ambienti.



Figura 2.2: Mappa Indoor da Specifiche

## 2.3 Problematiche e Limiti di Realizzazione

Dalla analisi delle specifiche è emerso che il problema principale da affrontare è la realizzazione della funzionalità di navigazione che permetta all'utente di muoversi facilmente tra i vari stand all'interno dei padiglioni. I Sistemi di navigazione e geolocalizzazione Indoor sono oggi giorno oggetto di ricerca e dibattito scientifico, non esiste infatti una tecnologia standard che permetta la realizzazione di un sistema del genere senza avere dei limiti o dei vincoli nella realizzazione, se per la geolocalizzazione outdoor esistono già mappe satellitari precise e l'utilizzo di gps combinato alla triangolazione dei segnali Wi-fi fornisce un sistema efficiente, stesso non vale per gli ambienti Indoor. Le problematiche principali da affrontare per la realizzazione di un sistema del genere sono in particolare : Come rilevare la posizione dell'utente in un ambiente indoor ? Come mappare un ambiente indoor? Quale tipo di sistema di navigazione utilizzare? Si analizzano ora le problematiche appena descritte correlate alle caratteristiche dell'app e ai limiti di realizzazione del progetto.

### **Posizione Dell'Utente : Geolocalizzazione Indoor**

Il problema da risolvere é conoscere la locazione dell'utente dell'app all'interno dell'ambiente che ospita la fiera , tenendo conto della sua posizione rispetto agli stand degli espositori o ad eventuali punti di interesse. Non é possibile calcolare la posizione di un utente in un ambiente Indoor utilizzando il GPS, infatti tale tecnologia non funziona correttamente all'interno di ambienti chiusi. Sono diverse le tecnologie alternative che si possono utilizzare per la geolocalizzazione Indoor, la maggiorparte delle soluzioni prevede l'utilizzo di hardware aggiuntivi da applicare in loco attraverso i quali si può conoscer la locazione dell'utente.

### **Mappatura digitale dell'ambiente Indoor**

Per creare un sistema di navigazione indoor é necessario creare una mappa digitale che rappresenti fedelmente l'ambiente Indoor, nel ambito di questo progetto con riferimenti agli Stand, i punti di interesse e punti di ingresso e uscita. Esistono diverse tecnologie attraverso le quali poter realizzare la mappatura del genere , la maggiorparte di queste prevedono la presenza fisica del dispositivo mobile nell'area di interesse. Questo argomento verrà trattato più specificatamente nel capitolo 3 della tesi. Navigazione Indoor Per fornire agli utilizzatori dell'app una funzionalità di Navigazione che agevoli l'utente nel visitare la fiera , c'è bisogno che quest'ultimmo possa facilmente trovare la strada per arrivare ad un determinato punto o stand. Le Modalità per creare un sistema di Navigazione Indoor sono molteplici, verranno analizzate nello specifico nel capitolo 4 della tesi.

# Capitolo 3

## Mappatura di Ambienti Indoor

### 3.1 Problema

Per mappatura di ambienti indoor si intende la creazione di una mappa digitale che rappresenti fedelmente l'interno di un edificio, tenendo conto dei punti di ingresso, della suddivisione dei locali, dei punti di interesse principali e della eventuale presenza di punti di accesso a piani superiori o inferiori come scale o ascensori. Nell'ambito del progetto "Blick's Eye" la mappatura deve comprendere anche la presenza degli stand in cui vengono ospitati gli espositori della fiera, questi devono essere rappresentati fedelmente all'interno della mappa cioè con le giuste dimensioni e nella giusta locazione, inoltre deve essere prevista la possibilità di collegare tali stand ai relativi espositori, così da poter accedere direttamente dalla mappa alle informazioni riguardanti quest'ultimi. Per poter decidere quali tecnologie utilizzare per la mappatura, occorre innanzitutto analizzare le limitazioni poste dalla natura del progetto. Innanzitutto essendo l'obiettivo quello di creare diverse applicazioni per diversi eventi fieristici, occorre progettare un sistema di mappatura che sia possibile utilizzare più volte in diverse condizioni e in edifici che possono essere anche diversi tra loro, è importante considerare che tali eventi possono essere dislocati in differenti città del mondo anche nello stesso periodo. Nonostante la disposizione degli stand nella fiera sia progettata solitamente

con largo anticipo dall'inizio dell'evento tramite delle planimetrie , spesso questi vengono allestiti solo poche ore prima dell'inizio della manifestazione, per questo risulta molto complicato pensare di effettuare una mappatura una volta che tutti gli stand sono stati preparati. Per ovvi motivi creare la mappa digitale a sole poche ore dall'inizio dell'evento é altamente rischioso infatti se per qualche ragione il sistema non dovesse funzionare nel modo corretto, ci sarebbe il rischio che l'intera applicazione non riesca a funzionare. Alla luce di tale considerazioni sono state valutate diverse tecnologie con la quale poter risolvere tale problematica.

## 3.2 Alternative

### 3.2.1 IndoorAtlas

Indoor Atlas [6] é una piattaforma per la geolocalizzazione e navigazione Indoor che sfrutta il campo geomagnetico terrestre, rilevabile tramite gli smartphone attraverso il sensore magnetico , cioè la bussola . Tale tecnologia si fonda sul principio per il quale ogni edificio, in virtù delle sue dimensioni, forma e materiali di costruzione, interagisce in maniera unica e distintiva con il campo magnetico terrestre. Il Software di IndoorAtlas è in grado di utilizzare questa traccia come mappa per rilevare la posizione dell'utente e calcolare eventuali percorsi di navigazione all'interno di esso. La rilevazione delle variazioni del campo magnetico terrestre sta alla base di tale tecnologia che può essere però integrata attraverso l'utilizzo altri strumenti come wi-fi o dispositivi Bluetooth per garantire una maggiore precisione al sistema.



Figura 3.1: Logo IndoorAtlas

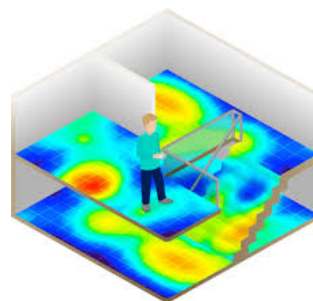


Figura 3.2: Funzionamento IndoorAtlas

### Mappare con IndoorAtlas

Collegandosi al sito "www.indooratlas.com" è possibile provare gratuitamente il sistema e si può subito creare la mappa digitale di un piano di un qualsiasi edificio a scelta ed utilizzarla all'interno di un'applicazione per dispositivi mobili. L'intero processo è diviso in 4 parti : scelta della locazione e aggiunta del piano , mappatura con Mapcreator , gestione dei dati della mappa ed infine creazione dell'app.

### Scelta della locazione dell'edificio e aggiunta del piano

Per prima cosa il sistema richiede di selezionare il punto esatto in cui si trova l'edificio di cui si voglia mappare il piano questo è selezionabile attraverso una mappa satellitare, oppure si può inserire per iscritto l'indirizzo esatto o le coordinate relative al luogo di interesse . Inoltre è possibile dare all'edificio che si intende mappare un nome ed una descrizione facoltativa. Una volta che la scelta è stata effettuata si passa all'inserimento dei piani, ogni nuovo piano inserito necessita di un nome obbligatorio , del numero di piano che rappresenta e della sua altitudine , inoltre occorre caricare anche l'immagine di una planimetria che rappresenti la sua suddivisione interna . Per fare questo occorre selezionare l'immagine adatta , dopodiché la piattaforma permette di allineare graficamente l'immagine scelta alla mappa digitale

in modo da poter ottenere una rappresentazione che riproduca fedelmente l'interno dell'ambiente.

### **Mappare con MapCreator**

Una volta inserito l'edificio e il piano desiderato con la pianimetria correttamente allineata, si passa alla parte vera e propria di mappatura, per fare questo occorre l'applicazione "IndoorAtlas MapCreator", inoltre bisogna essere dotati di un cellulare che rilevi il wi-fi e che possieda sia giroscopio che accelerometro, l'app infatti sfrutta la tecnologia basata sul campo magnetico terrestre per effettuare una vera e propria mappatura degli ambienti Indoor. Per mappare con IndoorAtlas MapCreator è innanzitutto necessario essere presenti fisicamente all'interno dell'ambiente di interesse, nella schermata principale dell'app viene visualizzata l'immagine relativa al piano che è stata caricata in precedenza. Per prima cosa occorre selezionare una serie di waypoint in punti facilmente riconoscibili all'interno della stanza, è necessario all'incirca selezionare un waypoint ogni 10 - 20 metri, è importante che questi siano distanti almeno mezzo metro dalle pareti. L'applicazione permette di calcolare la distanza tra 2 punti cliccando prima su uno di essi e poi su un'altro, questo permette all'utente di capire se la mappa è nella giusta scala e di rendersi conto di dove sono i luoghi corrispondenti ai waypoint. Dopo aver selezionato tutti i punti è necessario calibrare il telefono, per fare questo occorre lasciare il dispositivo fermo appoggiato con lo schermo rivolto verso l'alto finché la prima parte di tale processo non termina, dopodiché occorre roteare il telefono tenendo il centro del cellulare fisso finché il processo di calibrazione non è terminato. Una volta calibrato il telefono si può passare alla mappatura vera e propria dell'ambiente, per fare questo occorre innanzitutto selezionare un punto di partenza tra i vari waypoints creati, dopodiché bisogna posizionarsi esattamente nella posizione corrispondente al punto selezionato, effettuate queste operazioni si può iniziare a mappare cliccando sul waypoint di partenza e premendo il tasto avvia. Una volta avviata la modalità di mappatura occorre camminare fino al waypoint



piú vicino a quello di partenza, la camminata deve essere il piú naturale e rettilinea possibile , inoltre lo smartphone deve essere tenuto sempre vicino al corpo e devono essere evitati movimenti bruschi con esso, una volta sicuri di essere nel secondo punto basta selezionarlo e l'applicazione mostrerá la strada percorsa, per mappare l'intera area é necessario ripetere l'operazione con tutti i waypoint presenti, fino che non si torna in quello di partenza. Una volta mappati tutti i waypoints ' possibile generare e salvare la mappa per poi riutilizzarla in un'applicazione.

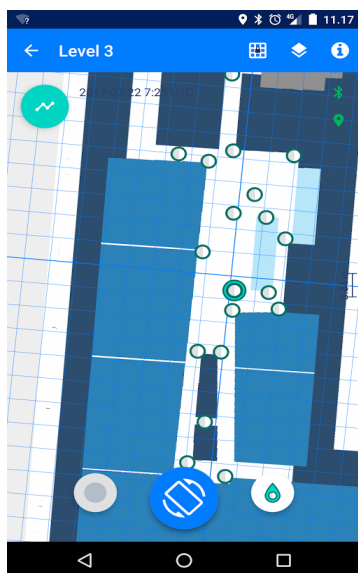


Figura 3.3: Esempio di utilizzo MapCreator

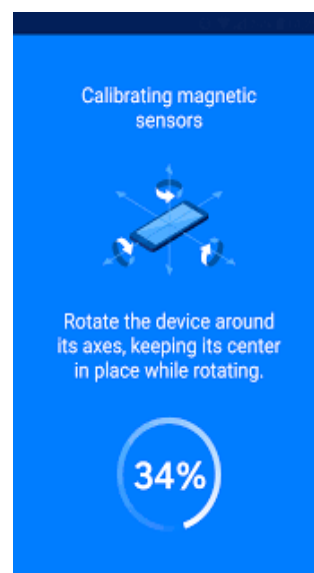


Figura 3.4: Esempio di Calibrazione MapCreator

### Gestione della Mappa e Creazione dell'App

Una volta creata la mappa IndoorAtlas permette di gestirla a proprio piacimento andando inserire eventuali informazioni extra riguardanti i waypoints ed andando ad inserire tutti i vari punti di interessi necessari. IndoorAtlas può essere utilizzato per creare applicazione per la maggiorparte dei dispositivi mobili, infatti la piattaforma mette a disposizione un SDK sia per dispositivi Android che iOS.

### Problematiche e Limitazioni

Nonostante questa tecnologia sia molto interessante ed innovativa in particolare perchè non richiede l'integrazione di alcun dispositivo hardware , ci sono diverse problematiche che riguardano il suo utilizzo in particolare considerando il problema di mappatura che ci si pone all'interno di questo progetto. Il primo problema riguarda il funzionamento della tecnologia , essendo infatti piuttosto recente non ci sono ancora informazioni molto dettagliate riguardo al suo effettivo funzionamento. In particolare uno dei dubbi maggiori riguarda la sua veridicità in presenza di un alto numero di persone, infatti ogni essere umano essendo composto per gran parte di fluidi , comporterà una piccola variazione del campo geomagnetico terrestre in un determinato punto, se questo risulta ininfluenza per una singola persona non si può sapere se il funzionamento sia lo stesso in presenza di un'alto numero di individui. Una soluzione a questo problema potrebbe essere mappare l'edificio interessato quando all'interno si trovano già un discreto numero di persone, tuttavia questo risulta molto complicato da applicare ad un evento fieristico, infatti i visitatori si presenteranno alla fiera solo dalla data di inizio in poi , quando l'applicazione dovrebbe già essere funzionante e quindi anche la mappatura dovrebbe essere già pronta. Oltre alle persone , anche gli oggetti possono provocare variazioni sullo spettro magnetico , per questo motivo per effettuare una mappatura corretta occorrerebbe procedere a tale operazioni quando l'intera Fiera é già stata allestita e anche la merce é già stata esposta, altrimenti si rischia di avere dei grandi errori di precisione. Tuttavia effettuare

la mappatura a fiera completata risulta piuttosto complicato , infatti spesso gli stand vengono allestiti la sera prima del giorno di apertura o addirittura a volte anche la mattina prima dell'orario di inizio della manifestazione, questo significa che il tempo a disposizione per tale operazione è eccessivamente limitato , ed eventuali problematiche durante questa fase potrebbe portare al malfunzionamento generale dell'app. Per tali motivi è stata scartata l'ipotesi di utilizzo della piattaforma "IndoorAtlas"

### 3.2.2 Google Tango

Tango [7] é una piattaforma creata da Google , questa tecnologia utilizza la visione del computer per dare ai dispositivi mobili la possibilità di rilevare la propria posizione rispetto al mondo che gli circonda senza utilizzare GPS o altri segnali esterni. Questo consente agli sviluppatori di creare diversi tipi di applicazione ed esperienze per l'utente quali : navigazione Indoor , mappatura 3D, riconoscimento ambientale ed esperienze di realtà virtuale.

#### Come funziona ?

Per capire il concetto che sta alla base di questa tecnologia si può pensare a come tutti i giorni utilizziamo la nostra vista per orientarci ad esempio per trovare la strada adatta ad arrivare in una stanza e poi per sapere dove ci troviamo nella stanza o dove si trovano le pareti e gli oggetti che ci circondano . Le relazioni fisiche appunto determinano il modo in cui ci muoviamo e analizziamo l'ambiente dove viviamo tutti i giorni . La tecnologia Tango dá al dispositivo mobile la possibilità di avere questo tipo di conoscenza basandosi su tre concetti fondamentali : Motion Tracking (Tracciamento del movimento) , Area Learning (Studio dell'Area) e Depth Perception (Percezione di profondità) .



Figura 3.5: Logo Project Tango

### **Motion Tracking**

Un dispositivo mobile dotato della tecnologia Tango é in grado di tenere traccia del suo movimento attraverso lo spazio 3D , quando una persona si sposta con il dispositivo questo può capire dove si trova e come si sta muovendo, il sistema di tracciamento é simile a quello di un mouse con la differenza che non si muove su un piano ma su 3 dimensioni . Tango implementa il motion tracking attraverso l'odometria visiva inerziale o VIO , questa utilizza le immagini della telecamera per determinare una modifica della posizione , ad esempio se si scatta la foto ad un edificio da lontano e poi ci si avvicina e se ne scatta un'altra , si può calcolare la distanza della macchina fotografica in base alle posizione e alla dimensione dell'edificio sulla foto, questo viene integrato ai sensori di movimento presenti nel dispositivo mobile per valutare la rotazione e l'accelerazione.

### **Area Learning**

La tecnologia Tango fornisce a un dispositivo mobile la stessa abilità delle persone nel riconoscere dove si trovano in un ambiente osservando le

caratteristiche che lo circondano. In particolare l'Area Learning permette anche di salvare le caratteristiche uniche e relative a una determinata Area in modo da poterle riconoscere in futuro . Per fare questa operazione le caratteristiche visive di una zona vengono memorizzate matematicamente in un file ADF che verrà inserito in un indice consultabile dal dispositivo , tramite la funzione di localizzazione il dispositivo una volta memorizzata una zona sarà in grado di riconoscerla nel caso ci si ritrovi di nuovo, questo sistema é utile anche per migliorare il tracciamento del movimento . Il "Motion Tracking" se utilizzato autonomamente perde la sua precisione col tempo, ovvero piú viene utilizzato piú il percorso calcolato dal dispositivo mobile sarà variabile rispetto a quello veramente compiuto, tramite l' Area Learning si possono effettuare delle correzioni al percorso quando ci si accorge di essere in una determinata area salvata in memoria. Questo meccanismo serve a correggere gli errori di deriva , quando un dispositivo che sta compiendo un percorso si accorge di essere tornato all'origine cercherà di correggere gli errori di deriva per aver una rappresentazione piú realistica del percorso, stessa cosa vale per tutte le altre zone riconosciute, cosí da rendere la deriva sempre meno influente e il percorso calcolato sempre piú simile al percorso reale.

### Depth Precision

Questa tecnologia fornisce al dispositivo mobile la capacità di comprendere la distanza da un oggetto. Per fare questo la piattaforma Tango permette agli sviluppatori di utilizzare diverse metodologie tra le piú comuni come : infrarossi o onde stereo . I dispositivi attuali che utilizzano la tecnologia Tango sono progettati per funzionare meglio in ambienti Indoor su distanze minime da 0,5 a 4 metri, queste sono le condizioni ideali per l'utilizzo di un sistema a infrarossi per la rilevazione di profondità ; tuttavia tali metodi hanno delle limitazioni , infatti non funzionano correttamente quando gli ambienti sono fortemente illuminati dalla luce del sole, da lampadine a incandescenza o quando sono presenti oggetti che non riflettono la luce a infrarossi.

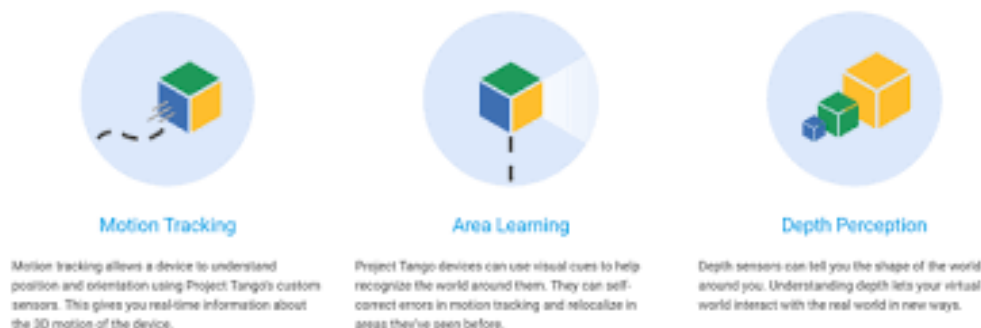


Figura 3.6: Funzionamento Motion Tracking, Area Learning e Depth Precision

### Google Virtual Positioning Service

Combinando le tecnologie messe a disposizione da Tango é possibile creare un sistema per mappare ambienti interni tramite un dispositivo mobile e di creare applicazioni basate sulla posizione senza dover utilizzare il GPS o altre tecnologie di geolocalizzazione. Purtroppo però tale servizio non é ancora stato del tutto sviluppato ed esiste solamente una beta a numero chiuso fornita da Google.

### Problematiche e Limitazioni

L'utilizzo di Tango é attualmente disponibile solo attraverso lo smartphone "Lenovo Phan 2 Pro" che é l'unico dispositivo mobile in commercio ad utilizzare questa tecnologia. Tale limitazione rende impensabile l'utilizzo di questo sistema per gli scopi che si pone il progetto "Blick Eye's", infatti per utilizzarla in questo momento sarebbe necessario che tutti gli utenti degli eventi fieristici fossero a disposizione di tale dispositivo mobile. Anche nel caso che la tecnologia si sviluppi rapidamente in futuro tale metodologia utilizzata per la mappatura di ambienti indoor presenterebbe comunque delle problematiche nel caso d'uso in questione. Utilizzando le tecnologie di Area Learning e Depth Precision si potrebbe infatti creare una mappatura virtuale dell'ambiente in cui si svolge l'evento fieristico, tuttavia per fare questo



Figura 3.7: Funzionamento Videocamera Lenovo Phab 2 pro

occorrerebbe muoversi fisicamente con il dispositivo mobile per tutta l'area interessata, questa operazione inoltre andrebbe fatta una volta che la fiera sia stata completamente allestita con tutti gli stand nelle posizioni corrette senza possibilità che avvengano ulteriori modifiche dopo la mappatura . Il tempo necessario ad effettuare tale processo è quindi variabile in base alle dimensioni dell'ambiente in cui si svolge la fiera. Il problema principale è che spesso gli eventi fieristici vengono allestiti con pochi giorni o ore d'anticipo dalla data di inizio e questo comporterebbe un tempo eccessivamente limitato per effettuare la mappatura e di conseguenza un rischio molto alto in caso di malfunzionamenti in questa fase. Prendendo in considerazione tali problematiche è stata scartata l'ipotesi di utilizzare la tecnologia "Google Tango".

## 3.3 Scelte di Implementazione

### 3.3.1 Soluzione

Analizzando le possibili soluzioni per la mappatura di ambienti indoor é stato subito chiaro il fatto che la maggior parte di queste tecnologie richiedano sia la presenza fisica nell'area di interesse, sia una certa quantità di tempo variabile in base alla dimensione di esso. Riferendosi al caso d'uso in questione l'utilizzo di queste metodologie dovrebbe avvenire nel momento in cui la fiera é completamente allestita , queste motivazioni hanno portato a scartare queste possibilità. Per riuscire a creare una versione beta dell'applicazione e' stato deciso quindi di cercare una soluzione che permettesse almeno di rappresentare la mappa dell'ambiente dedicato all'evento fieristico con la suddivisione in stand , senza dover essere necessariamente presenti nel luogo dell'evento fieristico. Per fare questo si é deciso di utilizzare un servizio di mappe online, come Google Maps ad esempio, che mostri una mappa con focus sul luogo della fiera sulla quale sovrapporre una planimetria, ovvero una immagine in scala dei locali suddivisi in stand realisticamente ed orientata nel modo corretto. In questo modo si avrebbe la visualizzazione della mappa della fiera come realmente si presenta , ma utilizzando un'immagine non si ha alcuna informazione ad esempio sulla locazione degli stand ne la possibilità di cliccare su uno di essi, occorre quindi una metodologia per georeferenziare alcuni punti dell'immagine così da poterli utilizzare nell'app. Le problematiche da risolvere per la creazione di un sistema di questo tipo sono dunque :

- Georeferenziare una Immagine
- Sovrapporre una Immagine ad una Mappa



### 3.3.2 Georeferenziare una Immagine con JOSM (Java OpenStreetMap Editor)

JOSM [8] é un software di editing per OpenStreetMap scritto in Java . Questo software permette di modificare mappe di OpenStreetMap grazie a una moltitudine di funzioni , puó essere utilizzato per la progettazione di strade o edifici , l'aggiunta di collegamenti ipertestuali alla mappa, la visualizzazione di dati 3D e moltissime altre funzionalitá vengono offerte dai numerosi plugin scaricabili gratuitamente. Permette inoltre di estrapolare le informazioni relative alle mappe in vari formati.

#### OpenStreetMap

OpenStreetMap [9] é un progetto collaborativo nato nel 2006 nato con la finalitá di creare mappe a contenuto libero del mondo , con questo si intende che i dati geografici presenti in OSM possiedono una licenza di tipo "Open Database License" é cioé possibile utilizzarli gratuitamente per qualsiasi scopo tenendo conto solamente del vincolo di citare le fonti ed utilizzare lo stesso tipo di licenza in caso di pubblicazione. Nella piattaforma OpenStreetMap chiunque puó contribuire ad arricchire o correggere i dati. Le Mappe vengono create tramite dati registrati da dispositivi GPS portatili, fotografie aeree ed altri tipi di fonti libere. OpenStreetMap ha come obiettivo quello di creare mappe sempre piú dettagliate ed aggiornate del mondo grazie all'interazioni con gli utenti , maggiore ' la partecipazione di quest'ultimi, maggiore sará la precisione dei dati. Openstreetmap permette anche la mappatura di ambienti indoor , per fare questo nel 2014 ' stato creato un sistema chiamato Simple Indoor Tagging con il quale sono stati aggiunti appositi tag per la mappatura Indoor.

### Mappare con JOSM

Grazie all'editor JOSM si può modificare una mappa di una determinata area di interesse, lo scopo finale è quello di ottenere dei dati geolocalizzati che si possano poi andare ad utilizzare in ogni tipo di mappa digitale (Google Maps, Maps di iOS, OpenStreetMap...) . Per iniziare è necessario scegliere l'area geografica sulla quale si vuole operare, per fare questo il programma permette l'utilizzo di una mappa di OSM attraverso la quale si deve selezionare un'area rettangolare delle dimensioni desiderate. Una volta scelta, l'area verrà rappresentata all'interno del programma JOSM con tutti i relativi elementi caratterizzati dai rispettivi tag. A questo punto si ha una rappresentazione fedele dell'area di interesse che può essere modificata a piacere grazie ai tools a disposizione.

Per creare la mappatura che ci interessa occorre innanzitutto sovrapporre alla mappa una immagine in scala che rappresenti gli stabili in cui si svolge l'evento fieristico suddiviso in maniera corretta con gli stand nelle relative posizioni. Per fare questo l'azienda "Blick Design ha messo a disposizione le planimetrie comprensive della suddivisione in stand di un evento fieristico passato ospitato da Bologna Fiera, più precisamente dislocato nei padiglioni 16,21 e 22. Per sovrapporre l'immagine della planimetria alla mappa si utilizza il plugin "ImportImagePlugin" di JOSM che permette di caricare una immagine in diversi formati e regolarne le dimensioni , l'angolazione e la posizione per allinearla con gli elementi presenti sulla mappa di OSM. L'immagine che è stata utilizzata per la sovrapposizione è una immagine PNG , questo perché garantisce la trasparenza. Una volta regolata la planimetria JOSM permette di salvare la calibrazione dell'immagine in un file di tipo cal , in previsione del caso in cui si necessiti di riutilizzarla in progetti futuri non sarà necessario posizionarla nuovamente nel modo corretto ma basterà caricare tale file. La planimetria con la suddivisione degli stand orientata precisamente serve per andare ad aggiungere tutti i riferimenti necessari che si utilizzeranno poi nell'app, tutti gli elementi presenti di default nella mappa di OSM non sono necessari , dato che per lo scopo prefissato servono so-

lamente gli elementi rappresentati nella planimetria , quindi possono essere eliminati tutti gli altri. A questo punto si otterrà una mappa vuota sulla quale è semplicemente sovrapposta la planimetria della fiera, si può quindi procedere con la mappatura vera e propria degli ambienti.

Per identificare gli stand si deve creare un waypoint per ognuno di essi, per fare questo si utilizzano gli elementi di editing della mappa forniti da JOSM, in particolare la funzionalità `drawnodes` permette di creare un nodo semplicemente facendo doppio click sul punto desiderato nella mappa. Ogni waypoint creato può essere modificato tramite l'utilizzo di tag, questi permettono di fornire informazioni aggiuntive per ogni nodo. Per identificare i punti che rappresentano gli stand vengono inseriti 2 tag, uno chiamato "Name" al quale viene assegnato come valore l'identificativo dello stand e l'altro chiamato "Description" al quale viene assegnato l'identificativo del padiglione di appartenenza, questo servirà successivamente per implementare il sistema di navigazione. Dopo aver mappato tutti gli stand si estrae il file GPX della mappa, il formato GPX è un particolare tipo di XML utilizzato per rappresentare dati GPS. Per utilizzare queste informazioni nell'app senza difficoltà il file GPX viene convertito in un file CSS tramite un servizio online " <http://www.gpsvisualizer.com/> ", il file convertito contiene informazioni sulla latitudine e longitudine di ogni punto e sul padiglione di appartenenza. Queste operazioni vengono ripetute per la creazione di un sistema di punti per la navigazione, questo procedimento sarà spiegato nel dettaglio nel capitolo 4.

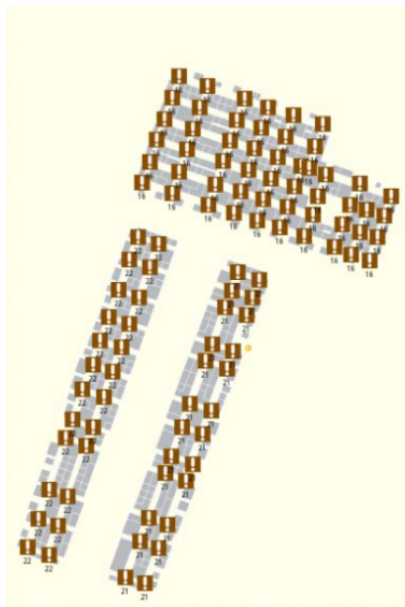


Figura 3.8: Mappatura degli Stand con Josm

### 3.3.3 Sovrapporre una Immagine ad una Mappa

JOSM permette di creare un file css contenente tutte le informazioni riguardanti gli stand della fiera, ora il problema è mostrare una mappa all'interno dell'applicazione che rappresenti tali informazioni e che dia una visione realistica dell'ambiente in cui si svolge l'evento sulla quale poi implementare i servizi di geolocalizzazione e navigazione. Per risolvere questo l'idea è di sovrapporre l'immagine delle planimetrie ad una mappa, per fare questo i principali servizi da utilizzare che permettono di mostrare mappe su applicazioni per dispositivi mobili sono :

- OpeenStreetMap
- Apple Maps
- Google Maps

### Quale Servizio Mappe Utilizzare?

L'utilizzo di un servizio di mappe è previsto in 2 diverse funzionalità all'interno dell'applicazione : "Come arrivare" e Navigazione Indoor. Si è deciso inizialmente di utilizzare lo stesso tipo di servizio per entrambe le problematiche così da avere un risultato simile sia graficamente che funzionalmente. Nel caso della funzionalità "Come arrivare" è necessario utilizzare un sistema che permetta di calcolare il percorso dalla posizione dell'utente fino alla locazione della fiera e che riesca ad elencare tutte le informazioni sul tragitto e i vari mezzi di trasporto a disposizione per raggiungere tale luogo. Le mappe di iOS in tal senso presentano delle limitazioni , infatti tale servizio prevede il calcolo dell'opzione di percorso secondo vari mezzi di trasporto , ma per quanto riguarda i servizi pubblici, come tram o treni ad esempio, le informazioni risultano abbastanza dettagliate unicamente nel suolo degli Stati Uniti , mentre questo servizio non funziona correttamente nel resto del mondo ,tranne che per alcune eccezioni. Anche per Openstreetmap tale funzionalità è disponibile , ma dato che gli utenti della piattaforma devono inserire tali informazioni , queste sono ancora presenti solo in un numero limitato di località. Google Maps è sicuramente il servizio più dettagliato in tal senso infatti esso possiede il maggior numero di informazioni riguardanti i trasporti pubblici rispetto agli, tale sistema è funzionante in Europa, Stati Uniti , Australia ed una buona parte dell'Asia ed oltre a permette di calcolare un percorso tramite l'utilizzo di mezzi trasporti pubblici è l'unico a prevedere anche la possibilità di volersi muovere in taxi o in aereo. La possibilità di sovrapporre una immagine alla mappa è presente in tutti i servizi elencati, tuttavia le problematiche relative a Maps di Apple e OSM per la realizzazione del servizio "Come Arrivare" hanno portato a scegliere l'utilizzo di Google Maps anche per tale scopo.

### Sovrapporre una immagine alla mappa con Google Maps

Dato che l'applicazione è stata realizzata in Swift , si analizzano le possibilità disponibili per sovrapporre una immagine ad una mappa fornite dal

SDK di Google Maps per iOS . Per realizzare tale obiettivo Google Maps mette a disposizione l'utilizzo di 2 particolari tipi d'oggetto :

**Ground Overlays** I Ground Overlays [10] sono sovrapposizioni di immagini sulla mappa legate alle coordinate, in questo modo l'immagine si adatterà in caso l'utente si sposti sulla mappa o in caso diminuisca o aumenti lo zoom , questo tipo di oggetto è particolarmente indicato quando si vuole fissare una immagine in un determinato punto che vada a ricoprire una area geografica non particolarmente grande.

**Tile Layers** I Tile Layers [11] sono sovrapposizioni di immagini che vengono visualizzate solo ad un determinato livello di zoom in una determinata area geografica. Questo tipo di oggetto è particolarmente utile quando si vogliono sovrapposizione che coprono aree geografiche particolarmente grandi.

Visto che l'area geografica in cui si svolge un evento fieristico solitamente non è eccessivamente estesa la soluzione migliore da utilizzare per risolvere il problema in questione sono i Ground Overlay. L'oggetto Ground Overlay per essere istanziato utilizzando Swift ha bisogno di 2 parametri , una UIImage cioè una immagine di qualsiasi formato ed un oggetto GMSCoordinateBounds, quest'ultimo identifica un rettangolo nella mappa caratterizzato da un punto minimo sud est e un punto massimo a nord ovest , questi 2 punti corrisponderanno rispettivamente all'angolo in basso a sinistra dell'immagine e l'angolo in alto a destra, attraverso i quali verrà posizionata l'immagine sulla mappa . L'oggetto GMSCoordinateBounds essendo creato tramite le coordinate si allargherà o restringerà in base allo zoom e con questo anche l'immagine che contiene. Per la corretta sovrapposizione l'immagine in questione deve essere già orientata nel modo giusto , dato che non esiste la possibilità di dare una angolazione all'oggetto GMSCoordinateBounds. La qualità della sovrapposizione dipende dalla qualità dell'immagine che si utilizza, inoltre questo tipo di oggetto va letteralmente a coprire la mappa , quindi nel caso si utilizzino formati che non gestiscono la trasparenza l'immagine

andrà a ricoprire l'intera area specificata senza possibilità di mostrare quello che si trova sulla mappa di Google Maps. E' possibile dare diversi parametri al Ground Overlay come ad esempio il grado di trasparenza o decidere se l'immagine è cliccabile o meno.

### Gestione dei Punti Georeferenziati con Google Maps

Google Maps oltre a permettere di sovrapporre l'immagine all'area desiderata in modo ottimale, permette anche in modo semplice la gestione di punti georeferenziati attraverso i marker. Tramite JOSM si riesce ad ottenere un file csv con latitudine e longitudine dei vari stand, i marker rappresentano singole posizioni all'interno della mappa quindi possono essere utilizzati per visualizzarli sulla mappa. Utilizzando Swift per istanziare un GMSMarker è necessario passare come parametro una CLLocationCoordinate, questo oggetto viene istanziato passando come parametri la longitudine e la latitudine di un punto, utilizzando i file csv creati grazie all'utilizzo di JOSM è dunque possibile creare un marker per ogni stand. I marker vengono rappresentati di default da google maps con il simbolo classico, tuttavia questi possono essere modificati a piacimento utilizzando una UIImage. Ad ogni GmsMarker è possibile associare un titolo, inoltre questi sono cliccabili ed è quindi possibile associare ad essi dei metodi. Questi possono essere inseriti anche sopra un Ground Overlay, tramite questo è possibile ottenere informazioni riguardanti ogni singolo stand, quindi rendere l'immagine interattiva.

## 3.4 Conclusioni Capitolo

Dopo aver valutato diverse possibilità per risolvere il problema di mappatura di ambienti indoor si è capito che non esiste una soluzione standard applicabile in qualsiasi caso d'uso. Non esiste infatti ancora una tecnologia di questo tipo della quale si è certi del perfetto funzionamento essendo questo un argomento di studio piuttosto attuale. I vari sistemi presi in considerazione presentano delle notevoli limitazioni, particolarmente se messe

in correlazione alle esigenze di realizzazione del progetto "Blick's Eye". In particolar modo le problematiche principali sono i tempi di realizzazione e le condizioni necessarie per effettuare la mappatura, infatti la maggior parte delle tecnologie di questo tipo richiede che tale operazione venga effettuata quando il luogo di interesse è nelle stesse condizioni nelle quali dovrà operare l'app una volta che sarà funzionante. Nel caso d'uso in questione rispettare queste specifiche significherebbe avere un tempo estremamente limitato tra questa fase e l'inizio dell'evento fieristico, questo porterebbe ad un rischio estremamente elevato nel caso si commettano errori. I tempi e i costi per effettuare una mappatura di questo tipo inoltre sono variabili in base alle dimensioni del luogo in cui si svolge l'evento fieristico, la totalità delle tecnologie prese in analisi infatti necessita di un operatore che effettui fisicamente un processo di mappatura ricoprendo tramite lo smartphone, con tecniche differenti, la totalità dell'area interessata. Non esistono dati certi sul tempo necessario ad esempio per ricoprire un tot di metri quadri e di conseguenza non può essere escluso il rischio che la quantità di tempo necessaria alla mappatura sia superiore a quella a disposizione dell'operatore, che nel caso d'uso in questione va dalla fine dell'allestimento della fiera all'inizio dell'evento. Tali considerazioni hanno portato ad escludere l'utilizzo di tali tecnologie perchè non adatte alle specifiche di realizzazione richieste dal progetto "Blick Eye". La soluzione che si è scelto di adottare evita quindi queste problematiche, infatti permette di effettuare tale operazione senza essere presenti fisicamente nell'area interessata. I tempi di realizzazione per la mappatura tramite l'utilizzo di JOSM sono sempre variabili a seconda delle dimensioni del luogo in cui si svolge l'evento, tuttavia queste operazioni possono essere effettuate con largo anticipo rispetto all'inizio della manifestazione fieristica con l'unica necessità di conoscere la disposizione degli stand. Questa tecnica di mappatura, inoltre permette di rimanere indipendenti sia dal tipo di tecnologia di posizionamento sia dal tipo di navigazione che si vuole implementare. La problematica principale che riguarda questo sistema è la necessità di conoscere la planimetria esatta degli stand, tuttavia per gli



eventi fieristici solitamente questa viene messa a disposizione con largo anticipo. In conclusione non è possibile trovare una tecnologia per la mappatura indoor che sia completamente priva di problematiche o limitazione, tuttavia la soluzione adottata risulta corretta per la realizzazione di una mappatura nell'ambito di un evento fieristico poichè rispetta le esigenze e le limitazioni imposte da questo caso d'uso.



# Capitolo 4

## Tecnologie di Navigazione Indoor

### 4.1 Problema

Per Navigazione Indoor , si intende un sistema che fornisca indicazioni all'utente per muoversi all'interno di un ambiente chiuso tenendo conto di eventuali ostacoli , punti di ingresso o di uscita , punti di accesso a piani inferiori o superiori e punti di interesse. Nello specifico il sistema di Navigazione Indoor che si vuole progettare per il progetto "Blick's Eye" ha come punti di interesse principali gli stand che ospitano gli espositori, questi naturalmente dovranno essere considerati anche come ostacoli , cioè dovranno essere evitati quando si calcola un percorso. Solitamente le fiere sono situate su diversi padiglioni il sistema di navigazione deve quindi essere in grado di riconoscere i punti di ingresso ai vari padiglioni in modo da calcolare percorsi che possano essere realmente utilizzabili. La navigazione indoor deve facilitare l'utente nel muoversi all'interno degli ambienti che ospitano la fiera , deve quindi essere facilmente utilizzabile ed intuitiva in modo che il visitatore dell'evento sia spronato ad utilizzarla . Non avendo ancora preso decisioni sul tipo di tecnologia di geolocalizzazione indoor da utilizzare l'obiettivo è di creare un sistema di navigazione indipendente da questo, in modo che

eventuali tecnologie di posizionamento possano essere integrate in futuro per migliorare tale sistema.

Alla luce di tali considerazioni sono state valutate diverse possibilità di realizzazione.

## 4.2 Alternative

### 4.2.1 Navigazione Inerziale

La navigazione inerziale [12] è una tecnica che si basa sull'utilizzo di sensori come giroscopio, accelerometro e magnetometro, al fine di stimare la posizione, la velocità, l'accelerazione e l'orientamento dell'oggetto in questione, questa tecnica è stata sviluppata inizialmente per la guida dei razzi ed è attualmente utilizzata per i sistemi di pilotaggio automatico in ambito marittimo, aeronautico ed aerospaziale. Quasi la totalità degli smartphone moderni possiedono dei sensori, tra cui anche accelerometro, giroscopio e magnetometro, in linea teorica è quindi possibile creare un sistema di navigazioni basato su questi.

#### Dead Reckoning

Il meccanismo di Dead Reckoning consiste nel calcolare la posizione di un corpo in movimento avendo come dati di input la posizione di partenza ed altre informazioni come la velocità, la direzione e la durata dello spostamento. Il concetto che sta alla base di questa tecnica può essere pensato come ad una somma vettoriale, ponendo la posizione di partenza in un punto di un piano cartesiano, posso valutare lo spostamento come un vettore che partendo da tale posizione ha come direzione la direzione dello spostamento calcolata grazie ai dati del giroscopio e come modulo la distanza percorsa calcolata in base alla velocità e alla durata dello spostamento. Calcolando un percorso con la tecnica di Dead Reckoning, è necessario tener conto degli errori cumulativi generati dai sensori ad ogni nuova computazione, questi possono essere causa

di una incorretta rilevazione della velocità del moto o della direzione. Gli errori possono essere causati dalla precisione non elevata dei sensori, anche se questi risultano minimi sulle singole rilevazioni utilizzare questo metodo per percorsi di durata molto ampia può portare ad errori di deriva molto significativi. Per evitare che il percorso calcolato si allontani dal percorso reale con il passare dei secondi è necessario che il sistema sia in grado di autocalibrarsi periodicamente, per fare questo deve utilizzare una posizione reale ed affidabile che viene considerata come nuovo punto di partenza per riapplicare l'algoritmo. Le tecniche di navigazione inerziale devono essere integrate ad altri tipi di tecnologia per fare in modo che il loro funzionamento sia realistico, utilizzandole autonomamente il rischio di commettere errori di deriva è troppo grande. Per ricalibrare l'algoritmo di navigazione inerziale è possibile utilizzare diverse metodologie, è stata analizzata in particolare la possibilità di utilizzo a tale scopo della tecnologia Qr-Code e della tecnologia NFC.

### QR-Code

Qr Code è l'abbreviazione di "Quick Response Code" ed è un codice a barre bidimensionale, cioè una matrice composta da moduli neri disposti all'interno di uno schema di forma quadrata. Questi codici sono predisposti per contenere informazioni generalmente allo scopo di essere lette tramite smartphone o altri dispositivi mobili. I QR code possono contenere qualsiasi informazione di tipo alfanumerico, queste sono accessibili attraverso ogni dispositivo mobile dotato di applicazioni apposite per la lettura di tali codici. Esistono diverse applicazioni anche gratuite che permettono la lettura dei QR-Code, queste necessitano semplicemente di una fotocamera, quindi la codifica di tali codici è possibile da qualsiasi smartphone o dispositivo mobile che ne possieda una. Il vantaggio principale di questa tecnologia è il costo particolarmente basso, infatti una volta generati i QR-Code con le informazioni necessarie, basta stamparli su dei semplici fogli di carta per renderli immediatamente accessibili agli utenti. Nel caso d'uso in questione

dunque potrebbero essere applicati dei QR-Code in diversi punti di interesse, ad esempio uno per ogni stand fieristico, con informazioni relative alla loro posizione. Questi potrebbero essere utilizzati sia per definire il punto di partenza esatto sia per ricalibrare l'algoritmo di Dead Reckoning nel caso la navigazione sia iniziata da troppo tempo e dunque occorre reimpostare il punto di inizio e rilanciare l'algoritmo per non andare incontro ad errori di deriva. L'utente per garantire il corretto funzionamento dell'applicazione dovrebbe dunque scannerizzare un QR-Code all'inizio per definire il punto di partenza ed iniziare la navigazione, poi dovrebbe scannerizzarne altri in caso il sistema valuti che si sta andando incontro ad errori di deriva. Il numero di volte le quali un utente dovrà scannerizzare un codice con il cellulare dunque variano in base alla lunghezza del percorso, ciò significa che se magari per un tragitto molto breve queste operazioni non creano nessun disturbo, in caso di un percorso molto lungo l'utente potrebbe preferire orientarsi senza l'aiuto dell'app piuttosto che dover ricalibrarla un elevato numero di volte.

### **NFC (Near Field Communication)**

NFC sta per Near Field Communication, in Italiano "comunicazione di prossimità", ovvero una tecnologia che fornisce una connettività wireless a corto raggio cioè fino ad un massimo di 10 centimetri. Questa tecnologia è stata sviluppata in collaborazione da Philips, Lg, Sony, Samsung e Nokia. NFC permette una comunicazione bidirezionale, ovvero avvicinando 2 dispositivi dotati di tale tecnologia si instaura una connessione automatica peer to peer che permette lo scambio di informazioni. Esistono in commercio Tag NFC applicati a targhette con dimensione di pochi centimetri quadrati a prezzi relativamente bassi, cioè di circa 1 euro a targhetta. Avendo a disposizione un qualsiasi dispositivo mobile dotato di tale tecnologia è possibile sia inserire che ricevere informazioni dal Tag NFC semplicemente avvicinandolo ad una distanza inferiore a 10 cm. L'utilizzo di tale tecnologia per ricalibrare l'algoritmo di Dead Reckoning è praticamente identico alla modalità di utilizzo dei QR-code. Inserendo informazioni sulla posizione si potrebbe ap-

plicare una targhetta NFC in dei punti di interesse ,ad esempio applicandone una per ogni stand. L'utente per ottenere informazioni dovrebbe avvicinare l'applicazione a una delle tag , questa operazione dovrebbe essere effettuata per indicare il punto di partenza e ripetuta ogni volta che l'algoritmo necessita di essere ricalibrato. L' unica differenza di utilizzo con i QR-code è che quest'ultimi necessitano che la fotocamera sia in funzione per poter essere decodificati , mentre per i tag NFC se il dispositivo è dotato di questa tecnologia , questo necessita semplicemente di essere avvicinato , quindi tale operazione potrebbe anche essere eseguita in background mentre il telefono è bloccato. Tuttavia come per i QR-code il numero delle operazioni di calibrazione da effettuare varia in base alla dimensione del percorso, in caso di un percorso molto esteso questo potrebbe risultare piuttosto svantaggioso per l'usabilità dell'applicazione.

### **Problematiche e Limitazioni**

La limitazione principale dei sistemi di navigazione inerziale è il fatto che devono essere necessariamente integrati con un'altra tecnologia per far sì che il sistema funzioni correttamente. Analizzando la possibilità di utilizzare QR-Code o Tag NFC si pongono grosse limitazioni all'usabilità dell'app con il rischio che il sistema di navigazione diventi troppo complicato per gli utenti.

#### **4.2.2 Navigazione tramite Database Fotografico**

La navigazione tramite fotografie non prevede di rappresentare un percorso su una mappa, bensì tramite questo sistema l'utente viene guidato verso la destinazione tramite delle foto che rappresentano in modo chiaro e riconoscibile punti del percorso. L'utente una volta selezionata la destinazione ottiene una sequenza di immagini che partono dalla più vicina alla sua posizione fino ad arrivare alla destinazione . Avviando la navigazione l'utilizzatore visualizza l'immagine più vicina alla sua posizione della sequenza che forma il percorso, a questo punto gli basterà guardarsi intorno per poter identificare la zona rappresentata dall'immagine e dirigersi in quel punto ,

questa operazione dovrà essere ripetuta dall'utente ad ogni nuova foto fino ad arrivare alla destinazione . [13]

### **Fotografie Georeferenziate**

Per creare un sistema di navigazione di questo tipo, è necessario avere informazioni riguardanti le posizione delle immagini . Con processo di Geotagging si intende applicare informazioni geografiche a vari tipi di dati , come immagini, video, sms eccetera. Esistono già numerosi servizi che permettono di aggiungere automaticamente informazioni sulla posizione una volta scattata una fotografia ad esempio come Facebook o Instagram. Avere fotografie che contengono informazioni sulla loro posizione in coordinate GPS permette di poter utilizzare qualsiasi servizio per la navigazione per calcolare automaticamente un percorso che passi per tali punti. Per creare un sistema di navigazione tramite immagini efficiente è necessario avere un elevato numero di immagini che ritraggono ogni possibile strada , punto di ingresso o di uscita e punti di interesse di tutta l'area interessata, queste immagini complete delle relative informazioni geografiche devono essere salvate in un Database. Utilizzare un sistema del genere in un'applicazione per dispositivi mobili necessita sicuramente dell'utilizzo di un Server che ospita il database dalla quale le immagini vengano scaricate gradualmente quando necessarie , altrimenti l'app andrebbe ad occupare uno spazio in memoria decisamente eccessivo.

### **Scorrimento delle immagini**

Lo scorrimento delle immagini può avvenire manualmente, ovvero quando l'utente si rende conto di trovarsi in prossimità dell'immagine rappresentata deve passare alla successiva per poter avanzare sul percorso . In alternativa lo scorrimento può essere automatico se esiste un sistema in grado di riconoscere gli spostamenti del dispositivo. Per un sistema di navigazione outdoor l'algoritmo potrebbe basarsi semplicemente sulle indicazioni del GPS per cambiare le immagini automaticamente , ma parlando di navigazione indoor sarebbe



ro necessari altre metodologie come l'utilizzo di un algoritmo di navigazione inerziale o di tecnologie di geolocalizzazione indoor.

### **Problematiche e Limitazioni**

L'implementazione di un sistema di navigazione tramite un database fotografico comporta notevoli limitazioni se si considera il suo utilizzo nell'ambito del progetto "Blick Eye". La problematica principale è il tempo necessario per la creazione del database di fotografie geolocalizzate, è necessario infatti avere un elevato numero di immagini per poter ottenere un sistema funzionante variabile in base alla grandezza della fiera. Questo problema potrebbe non essere rilevante nel caso sia possibile effettuare questa operazione con largo anticipo sul lancio dell'app, tuttavia come già detto spesso gli stand fieristici vengono allestiti con scarso anticipo sull'inizio della manifestazione, scattare le foto prima che questa sia completamente allestita porterebbe ad avere delle fotografie che non rappresentano fedelmente la realtà che l'utente dovrà essere poi in grado di riconoscere. Per evitare quest'ultima problematica si potrebbe decidere di utilizzare solo fotografie di elementi stabili e invariabili all'interno dell'area fieristica come ad esempio bar, ristoranti, servizi igienici eccetera, ma in caso questi punti non siano in numero elevato si potrebbero avere errori sulla precisione del sistema e la navigazione potrebbe rivelarsi complicata.

### **4.2.3 Navigazione tramite realtà Aumentata**

[13] Con Navigazione tramite realtà aumentata si intende un sistema che utilizzi costantemente la videocamera per riconoscere l'ambiente che lo circonda. Le indicazioni sul percorso da intraprendere devono essere direttamente visualizzate sulle immagini che la videocamera sta riprendendo in tempo reale. Il sistema deve essere in grado di riconoscere punti di riferimento semplicemente inquandrandoli, esistono 2 possibilità per applicare questa soluzione. Il primo è predisporre dei landmark artificiali riconoscibili dalla videocamera, in modo che quando un utente si trova in prossimità di uno

di questi dovrà semplicemente inquadrarlo per permettere al sistema di riconoscere il punto di riferimento in questione e di procedere con l'algoritmo di navigazione andando ad indicare il prossimo punto di interesse e dando indicazioni su come raggiungerlo. I Landmark artificiali potrebbero essere predisposti ad esempio tramite l'utilizzo di QR-Code o altri codici grafici. La seconda possibilità è quella di allenare l'algoritmo a riconoscere landmark naturali dell'ambiente, un esempio è il funzionamento dell'algoritmo di Area Learning del progetto Tango di Google [7].

### **Problematiche e Limitazioni**

La seconda possibilità rende la navigazione più semplice tuttavia tali metodi sono ancora in fase di sviluppo e non esistono tecnologie del tutto efficaci, inoltre il problema dell'utilizzo di tali tecnologie sta nella Mappatura dell'ambiente come analizzato nel capitolo 3. Predisporre invece dei landmark artificiali comporterebbe gli stessi problemi riscontrati nell'utilizzo di QR-Code o tag NFC come punti di calibrazione per algoritmi di navigazione inerziale, l'utente infatti per seguire il percorso dovrebbe prima trovare tali punti di riferimento e quindi inquadrarli attraverso la videocamera, questo potrebbe portare gli utenti ad avere difficoltà nel'utilizzo del sistema di navigazione.

## 4.3 Scelte di Implementazione

### 4.3.1 Soluzione

Analizzando le possibili soluzioni per l'implementazione di un sistema di navigazione indoor sono state evidenziate diverse limitazioni. Considerato il caso d'uso in questione l'utilizzo di sistemi di navigazione inerziale risulta particolarmente scomodo in particolare per la necessità di utilizzare altri tipi di tecnologie per ricalibrare l'algoritmo di "Dead Reckoning" questo può comportare una difficoltà all'utente nell'utilizzare tale sistema. Anche sistemi di navigazioni tramite immagine e realtà aumentata risultano difficilmente applicabili a tale caso d'uso a causa dei tempi ristretti che vanno dall'allestimento dell'ambiente fieristico all'inizio della manifestazione. In questo tempo dovrebbe essere infatti settato il sistema, cioè si dovrebbe creare il database fotografico per la navigazione tramite immagine o allenare l'algoritmo al riconoscimento dei landmark naturali per la navigazione tramite realtà aumentata. Alla luce di tale considerazione per creare una versione beta dell'applicazione è stato deciso di cercare una soluzione che permettesse di creare di una funzione di navigazione momentaneamente indipendente dalla tecnologia di geolocalizzazione che permetta di visualizzare un percorso sulla mappa di Google Maps sulla quale è stata sovrapposta l'immagine della planimetria, come spiegato nel capitolo 3.

### Problematiche

Sono due le problematiche principali che si sono poste per la realizzazione di tale sistema di navigazione, la prima riguarda gli stand fieristici mentre la seconda riguarda i punti di ingresso e uscita dai padiglioni. La prima problematica riguarda gli stand fieristici, il sistema di navigazione deve tener conto della loro presenza e non permettere ai percorsi calcolati di oltrepassare i loro bordi. Come spiegato nel capitolo 3 effettuando la mappatura con JOSM e sfruttando le funzionalità di Google Maps è possibile ottenere una mappa che rappresenti gli stand, ognuno di essi però viene localizzato

nella mappa da un singolo punto, mentre la struttura di tali stand è solitamente rettangolare o quadrata, non è possibile dunque per l'applicazione riconoscere autonomamente i bordi di questi e di conseguenza non è possibile ottenere un percorso che eviti di passarci attraverso. La seconda problematica riguarda i punti di ingresso, l'algoritmo di navigazione deve essere in grado di riconoscere quando il punto di partenza e la destinazione si trovano in padiglioni diversi, deve quindi costruire il percorso facendo passare l'utente per le porte. Questi problemi possono essere risolti creando un sistema di punti di navigazione utilizzando sempre JOSM.

### 4.3.2 Progettazione e Creazione della Griglia di Navigazione

La soluzione per evitare che il percorso calcolato oltrepassi i confini degli stand è creare una griglia di punti che saranno gli unici utilizzabili per il calcolo del percorso. L'idea è di considerare gli stand come rettangoli e sfruttare la loro disposizione solitamente schematica. Creando una serie di punti esterni ad ogni stand, precisamente in prossimità dei loro angoli è possibile creare una griglia di navigazione che può essere vista come un insieme di rettangoli. Muovendosi unicamente lungo i lati di tali rettangoli è impossibile oltrepassare i bordi degli stand, ma questo può accadere se ci si muove lungo la diagonale. E' facilmente dimostrabile tramite il teorema di Pitagora che i lati di un rettangolo misurano sempre meno della diagonale, questa proprietà può essere sfruttata per implementare un algoritmo che implementi questo ragionamento.

#### Creare la griglia di navigazione con JOSM

Come spiegato nel capitolo 3 utilizzando JOSM è possibile selezionare l'area di interesse della fiera e sovrapporci un'immagine che rappresenta la suddivisione in stand. Una volta compiuta questa operazione ed ottenuta una rappresentazione realistica dell'ambiente è necessario creare dei way-

point esterni agli stand in prossimità degli angoli , per fare questo si utilizza la funzione "drawnodes" che si trova tra gli strumenti di editing di JOSM. E' necessario creare un waypoint in prossimità di ogni angolo dello stand rimanendo tuttavia staccati di qualche centrimetro , questo per far si che una volta che il percorso sarà calcolato su questi punti , non sembri che oltrepassi i loro bordi. Ogni WayPoint creato necessita di informazioni , in questo caso questi punti non sono legati agli stand fieristici e nemmeno a degli espositori , tuttavia è necessario impostare un tag che mi permetta di capire all'interno di quale padiglione si trovi il punto, per far questo si può utilizzare il tag "description". L'utilizzo di tale tag serve per differenziare i punti di navigazione appartenenti a padiglioni differenti , nel caso esistano delle zone esterne a tutti i padiglioni è necessario creare dei punti di navigazione anche in queste porzioni di mappa , quest'ultimi non avranno nessun valore nel tag description perchè non appartengono a nessun padiglione. Una volta creata questa griglia di navigazioni è possibile esportarla in un file gpx per poi convertirla in un file csv contenente latitudine, longitudine e padiglione di appartenenza di ogni punto, questo file prende il nome di DirectionPoints.csv. Come si può vedere dal fig: 4.2 i punti contrassegnati da un punto interrogativo presentano dei tag infatti si trovano all'interno dei padiglioni, mentre quelli esterni rappresentati da dei puntini gialli non contengono nessun tag. Si utilizza JOSM anche per creare un documento che rappresenta i punti di ingresso e uscita dei vari padiglioni , per fare questo occorre creare un waypoint in ogni porta , anche per questi punti è necessario indicare il padiglione di appartenenza tramite il tag "description", e possono essere esportati in un file che prende il nome di "DoorPoints.csv" .

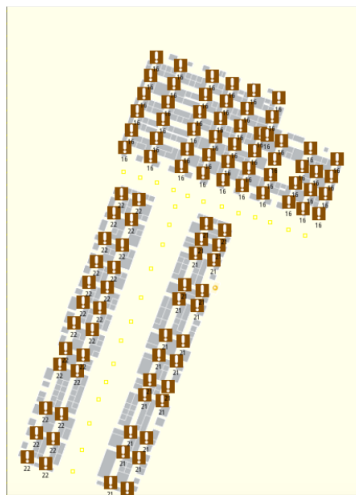


Figura 4.1: Mappatura DirectionPoints con Josm

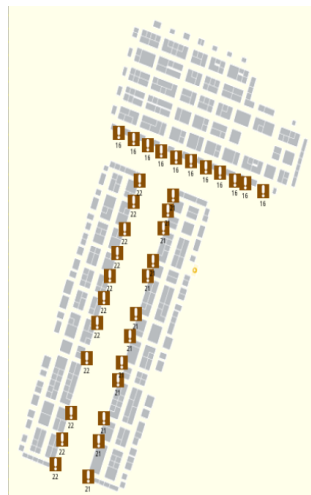


Figura 4.2: Mappatura DoorPoints con JOSM

### 4.3.3 Algoritmo di Navigazione Indoor

Una volta completate le operazioni appena elencate con l'utilizzo di JOSM si è a disposizione di 2 file csv contenenti dei punti georeferenziati sulla quale basare l'algoritmo di navigazione. Un file chiamato "DirectionPoints.csv" che rappresenta la griglia di punti esterni agli stand i quali sono gli unici attraverso i quali il percorso può passare. Il secondo file è chiamato "DoorPoints" e rappresenta i punti di entrata ed uscita da ogni padiglione. E' necessario dunque creare un algoritmo che permetta di calcolare un percorso il più rapido possibile tra un punto di partenza e un punto di destinazione sulla mappa, il comportamento dell'algoritmo deve differenziarsi nel caso il percorso preveda uno spostamento interno allo stesso padiglione o se invece prevede lo spostamento da un padiglione ad un altro, nel secondo caso sarà necessario tenere conto anche dei "DoorPoints". In questo capitolo si analizza logicamente il funzionamento degli algoritmi in entrambi i casi, la loro implementazione sarà analizzata nel dettaglio nel capitolo 5.

### Navigazione Interna al Padiglione

Per la navigazione interna ad un unico padiglione non è necessario prendere in considerazione i punti d'entrata e d'uscita, ma l'unico problema da risolvere è evitare che il percorso calcolato oltrepassi i bordi degli stand . Il file `DirectionPoints.csv` rappresenta tutti i punti della griglia creata con JO-SM. Considerando i quattro punti esterni ad ogni stand come un rettangolo si è sicuri che muovendosi attraverso questi punti è possibile oltrepassare i bordi dello stand solo spostandosi lungo la diagonale , questo è invece impossibile nel caso ci si muova solamente attraverso i lati del rettangolo. Partendo da un determinato punto l'idea dell'algoritmo è di controllare inizialmente quali sono i punti più prossimi a a se stesso appartenenti allo stesso padiglione creando una lista ordinata per distanza . Dopodichè tramite un ciclo vengono esaminati gli elementi della lista, se la distanza tra la posizione attuale e la destinazione è maggiore della distanza tra l'elemento iesimo della lista e la destinazione, allora tale elemento sarà considerato come secondo punto del percorso , altrimenti il ciclo continua finchè gli elementi della lista non sono finiti . Questo processo viene ripetuto fino ad arrivare alla destinazione. L'algoritmo si arresta quando il punto iesimo che rispetta le condizioni è più distante alla destinazione rispetto alla posizione attuale oppure quando non c'è nessun punto che rispetta la condizione del ciclo, questa condizione indica che l'algoritmo si trova nel punto della griglia più vicino alla destinazione. Muovendosi attraverso i punti in questo modo si è sicuri che il percorso non oltrepasserà mai i bordi degli stand , perchè si muoverà seguendo i lati dei rettangoli costruiti intorno ad essi e mai lungo la diagonale . Quando l'algoritmo si trova nel vertice di un angolo di un rettangolo i 2 punti più vicini ad esso saranno quelli che se collegati formano un lato del rettangolo, uno dei 2 sarà sicuramente più vicino alla destinazione rispetto al punto iniziale , per questo è impossibile che venga selezionato il vertice dell'angolo opposto che se collegato rappresenterebbe la diagonale. Questo algoritmo non garantisce il calcolo del percorso più rapido per arrivare da un punto ad un'altro ma permette di risolvere il problema che questo caso d'uso pone , inoltre tende

a costruire percorsi piuttosto rettilinei facilmente utilizzabili dagli utenti.

## 4.4 Navigazione tra diversi Padiglioni

Nel caso in cui il punto di partenza e di destinazioni si trovino in padiglioni diversi è invece necessario tenere conto dei DoorPoints , ovvero le coordinate che rappresentano i punti di ingresso e di uscita dal padiglione. La logica dell'algoritmo è simile a quella della navigazione interna al padiglione ma questa volta dovrà considerare una lista delle porte relative al padiglione di appartenenza più vicine al punto di partenza e scegliere la più vicina che mi permetta di avvicinarmi anche al padiglione di destinazione. Per calcolare il percorso dal punto di partenza fino alla porta di uscita dal padiglione posso utilizzare l'algoritmo per la navigazione interna al padiglione , prendendo in considerazione solo i punti appartenenti a quello di partenza. Una volta che l'algoritmo ha raggiunto la porta di uscita, utilizza la stessa logica per calcolare la porta più vicina ad esso ma del padiglione di destinazione. Questa volta per calcolare il percorso dalla porta di uscita alla porta di ingresso, potrà utilizzare lo stesso algoritmo di navigazione interna analizzato precedentemente ma prendendo in considerazioni solo i punti esterni ovvero quelli non appartenenti a nessun padiglione. L'ultima fase dell'algoritmo è quella di calcolare il percorso dalla porta di ingresso al padiglione di destinazione fino alla destinazione , per fare questo utilizzerà sempre lo stesso algoritmo con i punti appartenenti al padiglione di destinazione.

## 4.5 Conclusioni Capitolo

Dopo aver analizzato nei dettagli le diverse possibilità per la creazione di un sistema di navigazione in ambiente indoor si è giunti alla conclusione che non esiste una tecnologia standard che permetta applicabile a qualsiasi caso d'uso. Le alternative prese in considerazioni presentato delle notevoli limitazione particolarmente se messe in relazione con le esigenze di realizza-



zione del progetto "Blick's Eye" . In particolare si è scelto di non utilizzare un sistema di navigazione inerziale per le problematiche legate all'utilizzo di altre tecnologie necessarie ad evitare gli errori di deriva. Sono stati scartati i sistemi di navigazione tramite fotografie o realtà aumentata per la quantità di tempo necessaria a raccogliere le fotografie o ad allenare l'algoritmo al riconoscimento di landmark.

La soluzione che si è scelto di adottare evita queste problematiche infatti permette la creazione di un sistema di navigazione indipendente dalla tecnologia di geolocalizzazione dell'utente e che richiede una tempistica relativamente breve per la sua realizzazione. Tale sistema di navigazione permette di calcolare percorso tra diversi punti della mappa evitando di oltrepassare i bordi degli stand e tenendo in considerazione la presenza delle porte di ingresso e di uscita dei padiglioni. Tale metodologia di navigazione può essere integrata tramite l'utilizzo di una tecnologia che permetta di calcolare automaticamente la posizione dell'utente.



# Capitolo 5

## Sviluppo dell'App

### 5.1 Tecnologie Utilizzate

#### Swift

È un linguaggio di programmazione objected-oriented progettato da Apple per sistemi macOS, iOS, watchOS, tvOS e Linux. Questo linguaggio di programmazione è creato per coesistere con il linguaggio Objective C.

#### XCode

XCode è un ambiente di sviluppo integrato, contenente strumenti utili allo sviluppo di software per dispositivi Apple. X-Code supporta diversi linguaggi di programmazione tra cui Objective-C e Swift.

#### CocoaPods

CocoaPods è uno strumento che permette l'integrazione di fonti di codici scritte da parti terze in progetti XCode realizzati in Objective-C o Swift.

#### Sketch

Strumento di sviluppo grafico professionale per applicazioni progettate per funzionare su dispositivi Apple.

## **GitHub**

Servizio Hosting per progetti software, un programmatore può rendere disponibile il codice sorgente di un programma per permettere il suo utilizzo ad altri sviluppatori.

## **Google Maps SDK for iOS**

Permette di sviluppare applicazioni per dispositivi iOS utilizzando i servizi di Google Maps. Prevede un set di oggetti e funzioni che permettono di sfruttare tutti i servizi di Google Maps.

## **Librerie Swift Aggiuntive**

**PXGoogleDirection** Libreria Swift che permette l'utilizzo semplificato dell'API "Google Maps Directions".

**SWRevealViewController** Libreria Swift che permette di creare dei menù laterali nell'app.

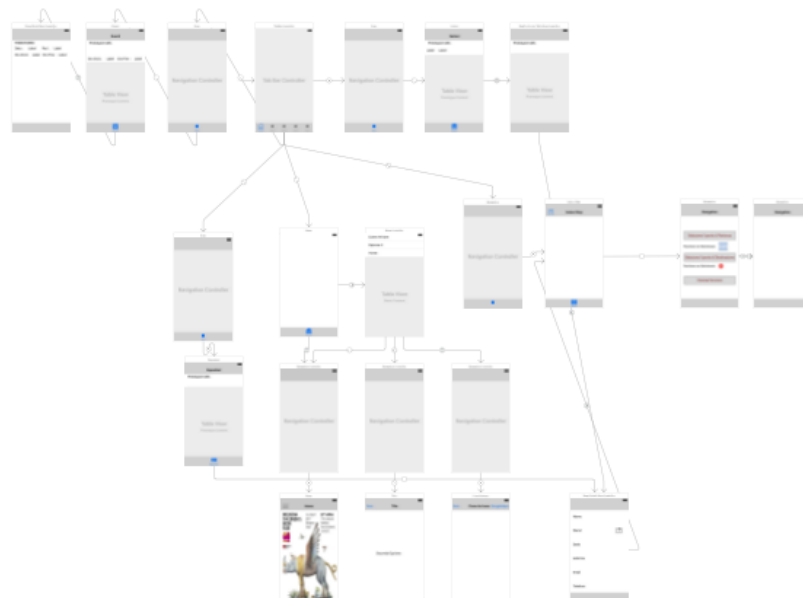


Figura 5.1: Main.Storyboard App

## 5.2 Struttura dell'App

### 5.2.1 Struttura Generale

Il file Main.Storyboard (fig 5.1 ) rappresenta la struttura generale dell'app, ovvero l'insieme di tutte le View e il modo in cui queste interagiscono tra loro. Con View si intendono le varie schermate che vanno a comporre l'applicazione, ognuna di queste è collegata ad un ViewController che ne gestisce il funzionamento, esistono anche particolari tipi di Controller che si utilizzano per gestire il funzionamento complessivo dell'app , ad esempio permettendo l'utilizzo di barre o menù di navigazione. In particolare nell'app sviluppata si utilizzano 2 Controller principali che facilitano il suo utilizzo da parte dell'utente : UINavigationController e UITabBarController.

## UITabBarController

L'UITabBarController permette di creare una barra di navigazione ai piedi dello schermo, tramite la quale è possibile muoversi attraverso le varie schermate principali. Ogni elemento della barra di navigazione è formato da una immagine ed un titolo che rappresenta la View al quale è collegato, in questa applicazioni la UITabBar presenta 5 icone che rappresentano: Homepage, Mappe, Espositori, Settori merceologici ed Eventi. Questo può essere considerata come la barra di navigazione principale dell'app, infatti tramite essa è possibile accedere a tutte le funzionalità principali ed è sempre presente in ogni schermata.

## Navigation Controller

Il Navigation Controller permette di gestire gerarchicamente le connessioni tra una view ed un'altra permettendo all'utente di spostarsi avanti e indietro tra queste senza perdere informazioni. Al contrario della UITabBarController non basta utilizzare un unico UINavigationController, ma è necessario crearne uno per ogni View, questo da la possibilità di gestire autonomamente ognuna di essa ed impostare per ciascuno una NavigationBar. La NavigationBar o Topbar è la barra di navigazione situata in cima allo schermo, creando un UINavigationController per ogni view è possibile impostarne una diversa per ogni view con funzionalità specifiche.

### 5.2.2 Home

L'Homepage presenta nella parte centrale una immagine che rappresenta l'evento fieristico in questione, come ad esempio la locandina. Nella topbar è presente il titolo della view in questione ed un pulsante che permette di aprire un menù laterale dal quale è possibile accedere ad alcune funzionalità di base, come le info sulla fiera o la funzionalità "come arrivare".



Figura 5.2: EventsTableView

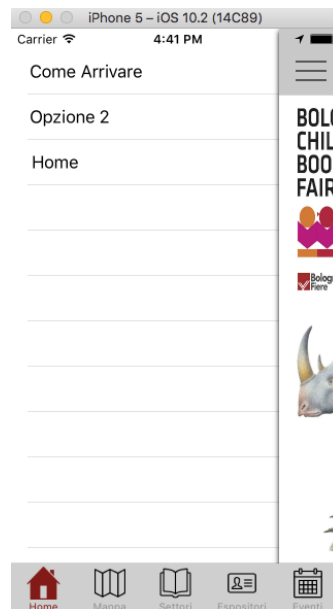


Figura 5.3: EventDetailView

### 5.2.3 Navigazione Outdoor

L'applicazione presenta 2 livelli di navigazione una per arrivare da un località qualsiasi al luogo della fiera ed una per navigare all'interno degli ambienti che ospitano la manifestazione. La navigazione Outdoor è gestita dalla funzione "Come Arrivare". Cliccando su questa opzione nel menù laterale della HomePage l'app riconosce la posizione attuale dell'utente, questo avviene automaticamente se il gps è attivo, altrimenti verrà richiesto all'utente di accenderlo. Una volta conosciuta la posizione di partenza questa funzione calcola un percorso per arrivare nel luogo dove si svolge la Fiera. Per calcolare il percorso si utilizza il servizio "Google Directions", in particolare il lavoro è facilitato dall'utilizzo dell'apposita libreria per swift "PXGoogleDirection". Questa libreria permette di calcolare direttamente il percorso passando come parametri le coordinate del punto di partenza e della destinazione, tale percorso viene poi rappresentato su una mappa di "Google Maps". Tramite la view come arrivare è possibile riportare il percorso calcolato in un'altra applicazione di mappe, per far questo basta cliccare sul pulsante "Google

Maps” in alto a destra, in caso il dispositivo sia dotato di Google Maps allora verrà aperta questa applicazione con il percorso calcolato dando la possibilità all’utente di sfruttare tutti i servizi annessi come ad esempio la scelta del mezzo di trasporto. Nel caso il dispositivo non disponga di tale app verrà aperta l’applicazione di mappe di default, nel caso di dispositivi iOS l’applicazione Maps di Apple.

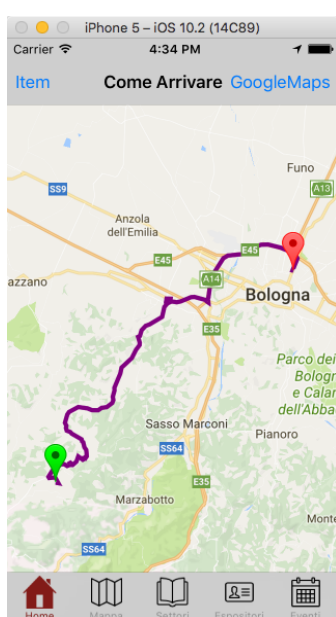


Figura 5.4: Funzione "Come Arrivare"

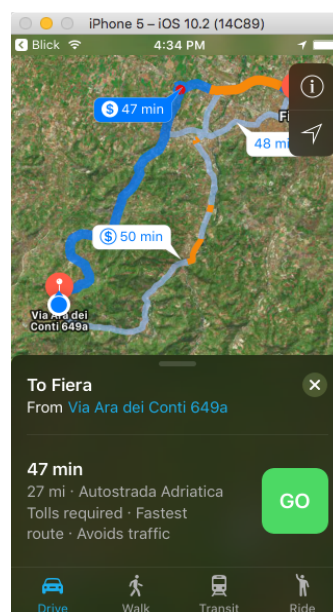


Figura 5.5: Percorso rappresentato su Maps di Apple



### 5.2.4 Navigazione Indoor

Nei capitoli precedenti è stato analizzato teoricamente il funzionamento di base del sistema di navigazione Indoor , si analizza ora come è stato implementato utilizzando le tecnologie a disposizione .

#### Gestione degli Stand

Tramite l'utilizzo di JOSM viene creato un file csv che riporta latitudine, longitudine, numero identificato e padiglione di appartenenza di ogni singolo stand. Questi punti vengono gestiti dalle seguenti classi :

**Stand** Questa classe rappresenta un singolo stand, contiene dunque le informazioni relative a longitudine e latitudine tramite una variabile di tipo "CLLocationCoordinate2D" , il numero identificato ed eventualmente una variabile di tipo "Expositor" che rappresenta l'espositore associato allo stand.

**StandModel** Questa classe rappresenta tutti gli stand presenti alla fiera . La classe contiene un array di oggetti di tipo "Stand" creata sulla base del file csv ottenuto grazie alla mappatura effettuata con JOSM. Il costruttore di questa classe non necessita di parametri ed esegue la funzione parseCsv , questa funzione permette di leggere il file csv relativo agli stand e riempire l'array di stand. All'interno del ciclo per ogni nuovo elemento viene controllato anche l'eventuale esistenza di un espositore collegato, per fare questo si istanzia un oggetto di tipo "ExpositorModel" che rappresenta tutti gli espositori della fiera e controlla se esiste una connessione con lo stand in questione tramite un ciclo.

#### Mappa degli Stand

La mappa degli Stand accessibile dalla seconda icona della TabBar, è una mappa di "Google Maps" con focus sul luogo della fiera , alla quale è sovrapposta la piantina degli stand. Per fare questo si utilizzano i Ground

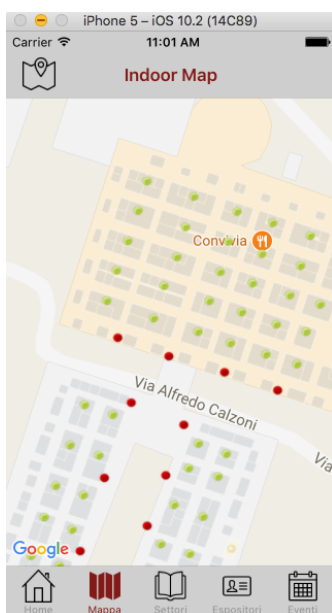


Figura 5.6: Esempio mappa degli stand

Overlays , tecnologia analizzata nel capitolo 3, inoltre vengono visualizzati tramite i marker tutti i punti relativi agli stand, cliccando su uno di questi è possibile ottenere informazioni sul relativo espositore. Queste operazioni vengono gestite da una classe chiamata "MapViewController".

### Gestione della Griglia di Navigazione

Tramite JOSM viene creato un file CSV che rappresenta la griglia di punti esterni agli stand che devono essere utilizzati per calcolare i percorsi di navigazione . I punti di navigazione vengono gestiti dalle seguenti classi :

**DirectionPoint** Questa classe rappresenta un singolo punto della griglia .

La latitudine e la longitudine sono memorizzate in una variabile di tipo "CLLocationCoordinate2D" , la classe contiene anche una variabile che indica il padiglione di appartenenza, questo può non avere nessun valore nel caso si tratti di un punto esterno ai padiglioni.

**DirectionPointList** Questa classe rappresenta tutti i punti della griglia di navigazione. La classe contiene un array di oggetti di tipo "Di-

rectionPoint” costruita in base al file csv contenente tutti i punti di navigazione.

**DoorPointList** Questa classe rappresenta tutti i punti di ingresso ed uscita , cioè i punti porta. Anche questi vengono rappresentati da una variabile di tipo ”DirectionPoint” , infatti le informazioni che contengono sono le coordinate e il padiglione di appartenenza. La classe contiene un array di oggetti ”DirectionPoint” costruiti in base al file csv che rappresente i punti porta.

**ModelDirection** Questa classe rappresenta la totalità del sistema di navigazione , infatti contiene un oggetto di tipo ”DirectionPointList” che rappresenta tutti i punti della griglia , un oggetto di tipo ”DoorPointList” che rappresenta tutti i punti porta e tutte le funzioni necessarie alla navigazione.

### Algoritmo di Navigazione

La navigazione avviene tramite una serie di algoritmi implementati nella classe ModelDirection. Si analizzano ora i diversi metodi utilizzati :

**OrderDirectionPointsByDistance** Questo metodo richiede un oggetto di tipo ”DirectionPoint” che rappresenta un punto della griglia di navigazione , e ritorna un array di ”DirectionPoint” appartenenti allo stesso padiglione, ordinati in base alla distanza da tale punto. L’algoritmo prende in considerazione gli elementi dell’oggetto ”DiretctionPointList”.

**OrderOutsidePointsByDistance** Ha lo stesso funzionamento dell’algoritmo descritto in precedenza ma prende in considerazione solo i punti esterni ai padiglioni.

**OrderDoorPointsByDistance** Questo metodo richiede in ingresso sia un ”DirectionPoint” di partenza che una variabile di tipo String che rappresenti il padiglione. Il metodo ritorna una lista di oggetti di tipo

"DirectionPoint" che rappresentano i punti di entrata e uscita dei padiglioni , cioè le porte. Il secondo parametro permette di indicare se si sta cercando una porta per uscire dal padiglione in cui ci si trova , o se si cerca una porta per entrare nel padiglione di destinazione. Il metodo prende in considerazione i punti dell'oggetto "DoorPointList".

**CalcPath** Questo è l'algoritmo che calcola il percorso tra 2 punti interni allo stesso padiglione. Il metodo ha come parametri 2 punti che devono essere appartenenti allo stesso padiglione, questi rappresentano il punto di partenza e la destinazione. Viene creata una variabile chiamata path che non è altro che una lista di "DirectionPoint" , il primo punto di questa lista è il punto di partenza. L'algoritmo partendo dall'inizio , ordina gli altri punti della griglia dal più vicino al più lontano attraverso il metodo "OrderDirectionPointByDistance" , dopodichè tramite un ciclo itera tutti gli elementi della lista fino a trovare un punto che sia più vicino alla destinazione rispetto alla posizione attuale. Se viene trovato un punto che soddisfa tali condizioni , questo sarà il secondo punto della variabile "path" e diventerà la posizione attuale dell'algoritmo. Questo procedimento viene ripetuto all'interno di un ciclo Do While , fino che non viene più rispettata la condizione precedentemente imposta, infatti se non esistono punti più vicini alla destinazione rispetto alla posizione attuale, significa che il percorso è stato costruito fino a raggiungere il punto della griglia più vicino alla posizione finale. Una volta terminato l'algoritmo verrà inserita la destinazione come ultimo punto della variabile path , questa lista di "DirectionPoint" rappresenta il percorso.

**CalcOutsidePath** Questo metodo ha lo stesso funzionamento del metodo CalcPath ma prende in considerazione solo i punti esterni ai padiglioni.

**CalcDirectionPath** Questo metodo permette di calcolare un percorso tra 2 punti qualsiasi della mappa indoor utilizzando la griglia dei punti creata con Jism. Questo metodo ha come parametri 2 oggetti del tipo "ClCoordinate2D", ovvero 2 punti qualsiasi sulla mappa, uno rappresenta la partenza e l'altro la fine del percorso da calcolare. Per prima cosa l'algoritmo controlla a quale padiglione appartengono i parametri tramite il metodo "sethPath". Se i 2 punti si trovano sullo stesso padiglione si utilizza il metodo "CalcPath" analizzato in precedenza, altrimenti si procede con l'algoritmo. Nel caso i 2 punti non si trovano sullo stesso padiglione, per prima cosa si utilizza l'algoritmo "OrderDoorPointsByDistance" per avere una lista delle porte di uscita dal padiglione iniziale ordinate in base alla distanza dal punto di partenza. Tramite un ciclo che itera tutti i punti porta della lista, viene scelta la prima porta che sia più vicina al punto di destinazione rispetto alla posizione di partenza, questo rappresenta la porta di uscita dal padiglione di partenza. A questo punto si utilizza l'algoritmo "CalcPath" per definire il percorso dal punto di partenza fino alla porta di uscita, dopodichè si utilizza nuovamente il metodo "OrderDoorPointsByDistance" per ordinare tutti i punti porta del padiglione di destinazione rispetto alla distanza dal punto in cui l'algoritmo si trova in questo momento. Tramite un ciclo che itera tutti questi punti, viene scelto il primo che è più vicino alla destinazione rispetto alla posizione attuale, questo rappresenta il punto di entrata nel padiglione che contiene il punto di arrivo. Ora può essere utilizzato il metodo "CalcOutsidePath" che permette di calcolare il percorso esterno ai padiglioni che dalla porta di uscita arriva fino alla porta di ingresso del padiglione di destinazione. L'ultima parte di percorso da calcolare è quella che va dalla porta di ingresso del padiglione di destinazione fino al punto finale, per questo può essere utilizzato di nuovo il metodo "CalcPath" dato che questi 2 punti si trovano all'interno dello stesso padiglione.

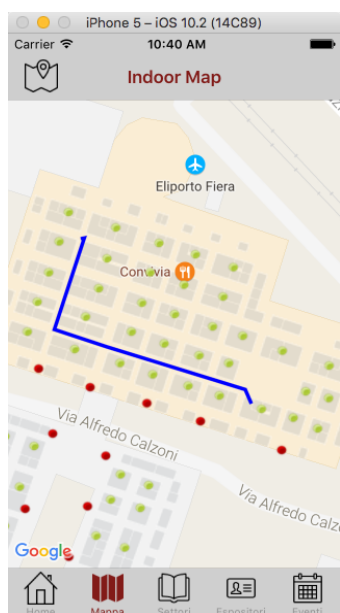


Figura 5.7: Esempio di Funzionamento del metodo CalcPath

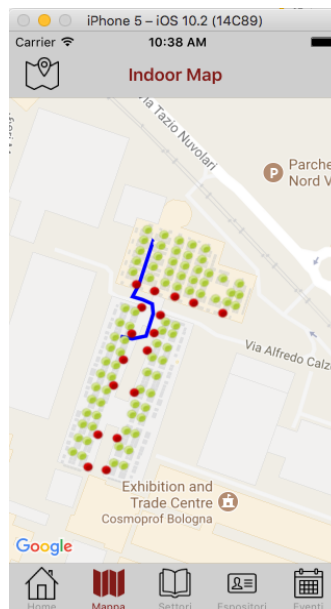


Figura 5.8: Esempio di Funzionamento del metodo CalculateDirectionPath

## Menù di Navigazione

Per avviare il calcolo di un percorso all'interno dei padiglioni che ospitano la fiera è necessario selezionare un punto di partenza e di destinazione. Per quanto riguarda il punto di partenza l'implementazione di un sistema di geolocalizzazione indoor eviterebbe la necessità di indicare la posizione nella quale ci si trova per iniziare il percorso, ma non essendo l'app dotata di tale sistema è necessaria definire chiaramente il punto di partenza. Per fare questo è necessario accedere alla View di Navigazione, gestita dalla classe "IndoorNavigationController", questa view è formata da 3 pulsanti, il primo serve a scegliere il punto di partenza, il secondo serve a scegliere la destinazione, mentre il terzo serve ad avviare il calcolo del percorso. Il pulsante calcola percorso non può essere cliccato prima di aver definito le 2 posizioni. I pulsanti "seleziona punto di partenza" e "seleziona punto di destinazione" funzionano in modo analogo, cliccando su uno dei 2 viene

visualizzata una mappa provvista della suddivisione in stand , per scegliere le posizioni necessarie per il calcolo del percorso basta cliccare in un punto qualsiasi della mappa. Una volta scelta sia la posizione di partenza che la destinazione viene richiamato l'algoritmo "CalcDirectPath" il quale ritorna un array di "CIDoordinate2D" che rappresenta tutti i punti che formano il percorso. Per unire i punti del percorso si utilizza l'oggetto "GMSPolyLine" che serve per disegnare delle linee sulla mappa di Google Maps, questa poly-line rappresenterà il percorso calcolato dall'algoritmo di navigazione indoor.

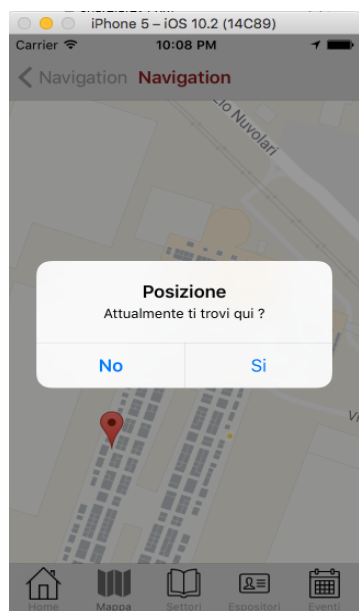


Figura 5.9: Seleziona Punto di Partenza

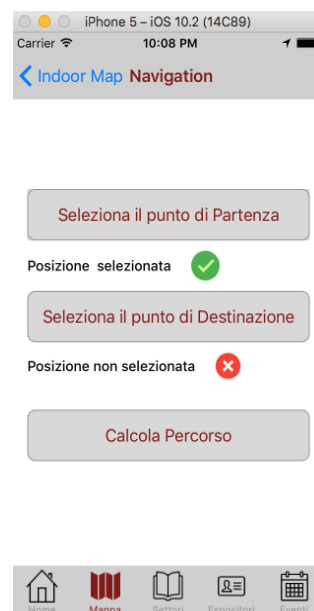


Figura 5.10: Mappatura Door-Points con JOSM

### 5.2.5 Espositori

Gli espositori vengono gestiti nell'applicazione tramite l'utilizzo di un file csv nel quale sono riportate tutte le informazioni principali su ciascuno di questi. Il file `Espositori.csv` contiene le seguenti informazioni : nome espositore, numero dello stand di appartenenza, email, indirizzo della sede, sito web, numero di telefono e settori merceologici trattati. Dato che ogni espositore può trattare diversi settori merceologici, questi vengono divisi dal simbolo "-" all'interno del file csv. Le informazioni riguardanti gli espositori sono accessibili attraverso 2 view presenti nell'app, una che mostra la lista di tutti gli espositori , ed un'altra che mostra i dettagli relativi al singolo espositore. Le classi utilizzate per gestire gli espositore sono :

**Expositor** Questa classe rappresenta un singolo espositore con tutte le informazioni che lo riguardano cioè nome, numero dello stand, indirizzo della sede, email, sito web e settori di appartenenza. Tutte le informazioni relative agli espositori è di tipo String tranne per quanto riguarda i settori di appartenenza, dato che questi potrebbero essere più di uno quindi vengono rappresentati da un array di tipo Sector classe creata per gestire i settori merceologici. La classe Expositor presenta un solo metodo `getStand()` che mi permette di risalire all'oggetto stand relativo alla mappa Indoor.

**ExpositorModel** Questa classe rappresenta la totalità degli espositori quindi è formata da un array che viene riempito andando a leggere le informazioni dal file `espositori.csv` , ogni riga del file rappresenterà un oggetto Expositor. Il costruttore del metodo non necessita di parametri ed esegue la funzione `parseCsv` , questo funzione permette di leggere le informazioni dal file csv tramite un ciclo e riempire l'array di espositori.

**ExpoTableViewController** Questa classe permette di visualizzare tutti gli espositori in una lista tramite una `UITableViewController` , ogni elemento della lista è cliccabile per ottenere le informazioni riguardanti



il singolo espositore. Per riempire la tabella si istanzia un elemento `ModelEspositori`, ogni elemento dell'array di espositori sarà una cella della tabella.

**ExpoDetailViewController** Questa classe permette di visualizzare tutte le informazioni relative all'espositore cliccato dalla lista tramite una view. In questa view è presente anche un pulsante che permette all'utente di visualizzare lo stand relativo all'espositore sulla mappa indoor, sfruttando il metodo `getStand()` della classe `Expositor`.

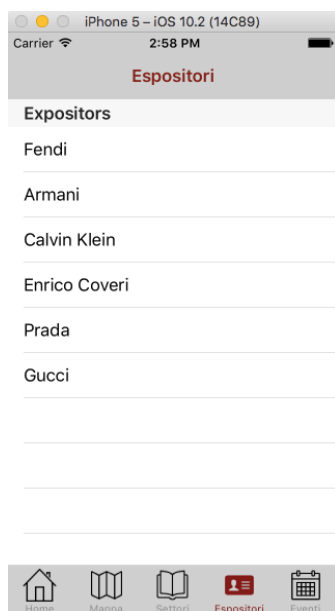


Figura 5.11: ExpoTableView



Figura 5.12: ExpoDetailView

### 5.2.6 Settori Merceologici

I Settori Merceologici vengono gestiti dall'applicazione attraverso l'utilizzo di un file csv nel quale sono riportati tutti i settori che la fiera tratta, ciascuno di questi è munito di un nome e di un id. E' possibile visualizzare tutti i settori merceologici tramite una lista, cliccando su uno di essi è possibile risalire inoltre ad un elenco di espositori che trattano tale settore. Le classi che gestiscono i settori merceologici sono:

**Sector** Questa classe rappresenta un singolo settore merceologico, contiene la descrizione e l'identificativo.

**ModelSector** Questa classe rappresenta l'insieme di tutti i possibili settori merceologici, dunque contiene un array che viene riempito andando a scorrere il file csv relativo ai settori. Il costruttore di tale classe non ha bisogno di parametri ed esegue la funzione `parseCsv` che permette di leggere le informazioni dal file csv e tramite un ciclo riempire l'array di settori. Questa classe inoltre contiene un metodo chiamato `getExhibitorsForSector` grazie al quale inserendo l'id di un settore è possibile ottenere una lista di espositori che trattano tale settore merceologico.

**SectorTableViewController** Questa classe permette di visualizzare tutti i settori merceologici in una lista, questi possono essere suddivisi in macrosettori, questo viene gestito in base alla struttura del file csv. Per riempire la tabella si istanzia un oggetto `ModelSector`, ogni elemento dell'array di settori merceologici rappresenta una cella della tabella. Cliccando su una cella della tabella si accede alla lista degli Espositori che trattano il settore merceologico corrispondente, questa lista viene gestita dalla classe `ExpForSectorTableViewController` che ha un funzionamento simile alla classe `ExpTableViewController`.

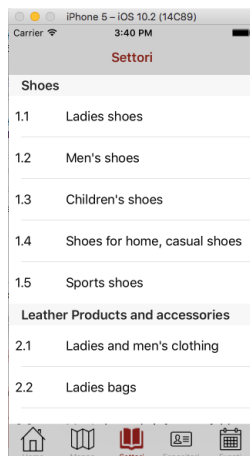


Figura 5.13: SectorTableView

### 5.2.7 Eventi

Gli Eventi vengono gestiti dall'app tramite l'utilizzo di un file csv nel quale sono riportati tutti gli eventi con le relative informazioni, ovvero: titolo, descrizione, data e ora di inizio e fine dell'evento e luogo in cui si svolge. È possibile visualizzare nell'app una lista di tutti gli eventi, cliccando su uno di questi viene visualizzata una nuova view con i dettagli relativi all'evento. Le classi che gestiscono gli eventi sono:

**Event** La classe Event rappresenta il singolo evento, e contiene tutte le sue informazioni ovvero: Data, ora di inizio, ora di fine, titolo, descrizione e luogo.

**EventsModel** Questa classe rappresenta l'insieme di tutti gli eventi, contiene un array di variabili di tipo "event" che viene riempito andando a scorrere il file csv relativo agli eventi. Il costruttore della classe "EventModel" non necessita di alcun parametro ed esegue la funzione parseCsv che permette di leggere le informazioni contenute nel file csv e tramite un ciclo riempire l'array di eventi.

**EventTableViewController** Questa classe permette di visualizzare tutti gli eventi in una tabella suddivisa per data, ogni riga riporta il nome

e l'ora di inizio e di fine. Per riempire la tabella si istanzia un oggetto `ModelEvent`, ogni elemento dell'array di eventi corrisponde ad una cella della tabella. Cliccando su una cella della tabella è possibile aprire una view che mostra i dettagli dell'evento corrispondente.

**EventDetailViewController** Questa classe permette di visualizzare tutte le informazioni relative ad un singolo evento quando viene cliccato dalla lista. In questa View è presente in alto il titolo, più in basso una `TextView` contenente la descrizione e sono riportati attraverso delle label l'ora di inizio e fine, la data e il luogo dell'evento.



Figura 5.14: EventsTableView



Figura 5.15: EventDetailView

# Capitolo 6

## Conclusioni

La realizzazione di sistemi di navigazione in ambienti indoor è una problematica per cui non esistono ancora tecnologie o metodologie standard da utilizzare. In Questa tesi si è cercato di realizzare un sistema adatto alle caratteristiche del progetto "Blick's Eye" ovvero adatto alla realizzazione di applicazioni in ambito fieristico. Le problematiche affrontate sono state in particolare quelle della mappatura in ambiente indoor e della realizzazione di un sistema di navigazione indipendente dalla tecnologia di geolocalizzazione. Per quanto riguarda la mappatura sono state valutate 2 metodologie una basata sull'utilizzo del campo geomagnetico terrestre e l'altra basata su un particolare utilizzo della fotocamera del cellulare. Entrambe le tecnologie nonostante siano molto interessanti sono ancora in fase di sviluppo e non esistono dati certi sul loro corretto funzionamento. Inoltre sono emerse numerose problematiche riguardo alla possibilità di un loro utilizzo in ambito fieristico in particolare dovute ai ristretti tempi a disposizione per effettuare la mappatura in loco.

Alla luce di queste considerazioni è stato progettato un sistema in grado di soddisfare le esigenze del caso d'uso , basato sulla sovrapposizione di una immagine georeferenziata ad una mappa tramite l'utilizzo di JOSM . Questo sistema permette di creare una mappa degli ambienti che ospitano la fiera suddivisa in stand e vengono rappresentati anche i punti di ingresso e di usci-

ta. Questa metodologia non garantisce il risultato migliore , ma permette di effettuare le operazioni di mappatura senza trovarsi nel luogo interessato, inoltre permette di rimanere indipendenti dalla tecnologia di geolocalizzazione che si intende utilizzare .

Per la progettazione di un sistema di navigazione indoor sono state valutate 3 possibilità. La prima opzione analizzata è stata la possibilità di utilizzare un sistema di navigazione inerziale basato sui sensori dei dispositivi mobili integrata con altri tipi di tecnologia , come QR-Code o NFC per garantirne il corretto funzionamento. Sono state poi valutate le possibilità di utilizzo di sistemi di navigazione che funzionano tramite database fotografici o realtà aumentata . La principale problematica che è stata riscontrata per quanto riguarda i sistemi di navigazione inerziali è la necessità di utilizzare tecnologie aggiuntive per evitare errori di deriva , questo può portare l'utente a dover ricalibrare l'app un numero elevato di volte rischiando di rendere l'utilizzo della navigazione scomodo. Utilizzare un sistema di navigazione tramite fotografie o realtà aumentata risulta svantaggioso in ambito fieristico a causa dell'alto numero di immagini dell'ambiente che devono essere raccolte , spesso le fiere vengono allestite poco prima dell'inizio dell'evento questo limita eccessivamente il tempo per effettuare tali operazioni.

Dopo aver analizzato queste alternative è stato progettato un sistema di navigazione basato su una griglia di punti geolocalizzati tramite JOSM. L'algoritmo di navigazione permette il calcolo di percorsi interni ai padiglioni che evitano di oltrepassare i bordi degli stand, inoltre tengono in considerazione i punti porta in caso si navighi tra padiglioni diversi. L'utilizzatore dell'app deve scegliere il punto di partenza e di destinazione per iniziare la navigazione , tuttavia il sistema può essere integrato tramite l'utilizzo di una tecnologia di geolocalizzazione indoor per calcolare automaticamente la posizione dell'utente.

Il progetto ha portato alla realizzazione di una prima versione dell'app capace di gestire le informazioni riguardanti la fiera, gli stand , gli espositori , i settori merceologici e gli eventi e in grado di fornire le principali funzionalità

richieste da specifiche. Questa versione dell'app non è ancora pronta per essere utilizzata ma rappresenta una base di partenza sulla quale poter andare a costruire altre applicazioni adattate a singoli eventi fieristici.

## 6.1 Sviluppi Futuri

L'applicazione può essere migliorata sia dal punto di vista grafico che dal punto di vista funzionale per poter avere sviluppi futuri anche di tipo commerciale. In particolare per quanto riguarda il sistema di navigazione interno alla fiera devono essere implementate 2 funzionalità:

**Posizionamento dell'utente** La prima funzionalità da implementare è la rilevazione automatica della posizione dell'utente. Per realizzare tale funzione può essere utilizzata qualsiasi tipo di tecnologia di geolocalizzazione indoor . La scelta migliore per questo tipo di sistema sembra essere l'utilizzo di beacon ovvero dispositivi Bluetooth low energy , tramite questa tecnologia oltre alla navigazione indoor è possibile implementare altre funzionalità interessanti per l'app come ad esempio funzioni di proximity marketing. Sono già state effettuate alcune prove per l'integrazione di tale tecnologia nella prima versione dell'app.

**Gestione dei piani** La seconda funzionalità da integrare per migliorare il sistema di navigazione è la gestione dei piani, infatti per ora l'applicazione è stata progettata senza considerare la possibilità che gli ambienti che ospitano la fiera si sviluppino su più piani. Per fare questo occorre implementare un sistema che mi permetta di sovrapporre alla mappa un'immagine diversa in base al piano che viene visualizzato.

## 6.2 Opinioni Personali

Lavorare al progetto "Blick's Eye" è stato per me molto stimolante innanzitutto perchè ho potuto sviluppare un progetto in ambito aziendale con possibilità di sviluppo futuro anche in ambito commerciale, inoltre ho acquisito una conoscenza abbastanza approfondita in particolare sui sistemi di navigazione e i sistemi di mappatura in ambito indoor. Essendo questi argomenti di estrema attualità anche per la comunità scientifica aver ottenuto una prima versione dell'app funzionante seppur non ancora completa è stato per me un risultato molto soddisfacente.



# Bibliografia

- [1] Here WeGo App . <https://here.com/en/products-services/consumer-app/here-wego-app>
- [2] Infsoft Indoor Positioning Systems. <https://www.infsoft.com/indoor-positioning>
- [3] Infsoft Indoor Navigation . <https://www.infsoft.com/solutions/indoor-navigation>
- [4] MapsPeople . <https://www.mapspeople.com/mapsindoors/>
- [5] Inmaps . <http://inmaps.co/>
- [6] IndoorAtlas . <http://www.indooratlas.com/>
- [7] Google Tango . <https://developers.google.com/tango/developer-overview>
- [8] JOSM . <https://josm.openstreetmap.de/>
- [9] OpenStreetMap.<https://it.wikipedia.org/wiki/OpenStreetMap>
- [10] Maps Ground Overlays for iOS. <https://developers.google.com/maps/documentation/ios-sdk/overlays>
- [11] Tile Layers for iOS. <https://developers.google.com/maps/documentation/ios-sdk/tiles>

- [12] Navigazione Inerziale. <https://it.wikipedia.org/wiki/Sistema-di-navigazione-inerziale>
- [13] Giovanni Cozza . Progettazione ed implementazione di un sistema di navigazione per dispositivi mobili in contesti ospedalieri.

# Ringraziamenti

Ringrazio innanzitutto i miei genitori che mi hanno permesso di intraprendere questo percorso , con la speranza di poterli ripagare in futuro di tutti i sacrifici che hanno fatto per me.

Ringrazio la mia fidanzata Giorgia perchè mi è sempre stata vicina soprattutto nei momenti più difficili facendomi sentire ogni giorno una persona speciale.

Ringrazio di cuore Zego, Kip, Pizza, Ciaffo, Tasso, Doccia , Zu, Sammy, Carpa, Gago, Fillo, Ruccio, Maso, Mezzo , Pipi , Bart, Tolle , Ghigga , Bolli e Mister perchè per me siete come fratelli e insieme a voi ho condiviso le esperienze più importanti della mia vita che mi hanno reso quello che sono.

Ringrazio tutti i ragazzi degli Ammaccabanane la miglior squadra di basket di Bologna perchè grazie a voi ho continuato a giocare allo sport più bello del mondo e ho vissuto momenti indimenticabili.

Ringrazio il mio collega, amico e socio in affari Gabriele Carlucci con il quale ho condiviso la realizzazione di questo progetto.

Infine ringrazio il professor Bononi per essere stato il relatore di questa tesi e la Dtt.ssa Yvonne Schmidt per avermi dato la possibilità di lavorare a questo progetto.