

ALMA MATER STUDIORUM · UNIVERSITÀ DI BOLOGNA

SCUOLA DI SCIENZE
Corso di Laurea in Ingegneria e Scienze Informatiche

**WAYFINDING E NAVIGAZIONE:
UN PROTOTIPO DI APP MOBILE
ACCESSIBILE PER PERCORSI INDOOR
PERSONALIZZATI**

Relatore:
Chiar.ma Prof.ssa
SILVIA MIRRI

Presentata da:
GIACOMO MAMBELLI

Anno Accademico 2018/2019

Introduzione

”Orientarsi significa letteralmente ”trovare l’oriente”, cioè il punto in cui nasce il Sole (dal latino oriri, nascere) e, in senso più ampio, sapere in quale direzione spostarsi per raggiungere un determinato luogo.” [38]. Sapersi orientare è considerata una delle capacità essenziali dell’essere umano ed è una azione che aumenta in difficoltà quando l’individuo si allontana dagli ambienti a lui conosciuti. Per raggiungere un luogo specifico è necessario conoscere la propria posizione e sapere come muoversi a partire da essa. Tale compito risulta ulteriormente complesso per le persone con disabilità. In particolare coloro che soffrono di disturbi visivi trovano nel sapersi orientare una sfida da affrontare giornalmente. Per questo la ricerca scientifica si è concentrata, in tempi recenti, anche sullo studio delle disabilità visive e delle tecnologie a supporto di chi ne soffre. Le moderne tecnologie di posizionamento e i sistemi di navigazione hanno semplificato l’erogazione di indicazioni turn-by-turn tramite dispositivi digitali, dando origine a validi strumenti a supporto dell’orientamento. Ne sono un esempio i navigatori GPS installati in negli smartphone e sui sistemi multimediali dei veicoli. Essi costituiscono un concreto aiuto nel wayfinding fornendo indicazioni turn-by-turn ad alta precisione, tuttavia questi sistemi sono spesso pensati per scopi di navigazione in spazi aperti e ignorano le problematiche specifiche dei luoghi chiusi, dove il segnale satellitare non può giungere. Questi sistemi sono dunque un ausilio per gli utenti con disabilità visive, ma non possono trovare utilità all’interno degli edifici non riuscendo ad indicare il percorso da seguire e le barriere architettoniche da evitare. Per abilitare la navigazione indoor

è necessario un cambio tecnologico che porti all'utilizzo di strumenti per il posizionamento adatti allo scopo e fruibili da tutti. Affidabilità e precisione continuano ad essere i focus principali, ma la strumentazione adottata deve necessariamente cambiare per incontrare le nuove esigenze. Numerose soluzioni sono state proposte con lo scopo di consentire operazioni di wayfinding in ambienti chiusi, alcune di esse sono general purpose ed altre pensate specificamente per essere sfruttate da utenti che soffrono di disturbi visivi. Ad oggi non esiste una tecnologia dominante e le idee suggerite dal mondo della ricerca prevedono soluzioni tecnologiche ed algoritmiche differenti tra loro per approccio al problema, modalità di utilizzo, affidabilità, precisione e costi. Ciò che pare chiaro è la necessità di sviluppare sistemi di navigazione indoor economici e che tengano in considerazione le esigenze del più ampio spettro di utenti possibile mettendo a disposizione interfacce accessibili ed erogando indicazioni corrette e precise in dipendenza delle specifiche necessità dell'utilizzatore.

In questa tesi vengono presentati lo studio e la realizzazione di una applicazione accessibile per la navigazione indoor. Gli scopi di questo software, disponibile sia per device Android che per quelli iOS, sono molteplici e includono l'intenzione di migliorare l'accessibilità degli edifici pubblici mediante uno strumento a basso costo. I principi all'origine del lavoro sono l'inclusione di utenti con disabilità sia visive che motorie abilitando la navigazione turn-by-turn indoor per gli individui che spesso trovano negato l'accesso a questo genere di strumenti. Inoltre, trattandosi di navigazione in palazzi storici, il sistema si occupa anche della presentazione di informazioni turistico/culturali legate alla posizione dell'utilizzatore.

La struttura di questo documento rispecchia i passi compiuti nello studio, nella progettazione e nella realizzazione del prodotto e si articola nel modo seguente.

Il primo capitolo presenta una overview sulle tecnologie di posizionamento e navigazione esistenti, sia analogiche che digitali, spostando l'attenzione dai sistemi pensati per l'outdoor a quelli nati come alternativa per l'indoor.

Vengono analizzati il funzionamento ed i pro e contro di ogni tecnica tenendo anche conto delle tipologie di utenti target della tecnologia stessa.

Successivamente, nel secondo capitolo, si presentano le specifiche del progetto. Tali specifiche prendono forma a partire dall'analisi dei requisiti ad alto livello osservati dalla prospettiva tecnologica della compatibilità con le possibilità offerte dal mercato in termini di sistemi operativi, tecniche di posizionamento, linguaggi, ecc. Le scelte vengono espone mantenendo accesa l'attenzione sul tema dell'accessibilità e sulle categorie di utenti finali delineatesi.

Le scelte implementative vengono invece presentate nel dettaglio nel terzo capito. Esso espone il processo realizzativo dell'applicazione considerando la tecnologia dei beacon e l'accessibilità i due cardini su cui sviluppare la struttura del prodotto. Le funzioni messe a disposizione dell'utente vengono elencate e approfondite singolarmente con l'intento di mostrare sia le possibilità di utilizzo del software che le scelte a basso livello che caratterizzano questo progetto. Una sezione è dedicata al server ed all'interfaccia web realizzati a completamento del sistema.

Il 4 capitolo espone le metodologie dell'User-Centered Design. In particolare tratta i test di usabilità come metodo adottato per la valutazione dell'interfaccia e delle interazioni tra utente e applicazione.

Il quinto ed ultimo capitolo propone alcuni dei possibili sviluppi futuri del sistema. Si tratta di nuove funzionalità, integrazioni e possibili modifiche a basso impatto sull'esperienza utente. Tali proposte nascono come raccolta di idee sorte durante lo sviluppo dell'applicazione e, per quanto coerenti con il progetto, richiedono uno studio di fattibilità ulteriore prima di una eventuale implementazione.

Seguono le conclusioni

Indice

Introduzione	i
1 Posizionamento e navigazione: Una overview	1
1.1 Tipologie di utenti	2
1.1.1 Utenti non esperti	3
1.1.2 Utenti con disabilità visive	3
1.1.3 Utenti con disabilità motorie	4
1.2 Posizionamento e navigazione outdoor	4
1.2.1 GPS	5
1.2.2 Prodotti in commercio	6
1.3 Posizionamento e navigazione indoor	8
1.3.1 Cartellonistica	8
1.3.2 Percorsi tattili	9
1.3.3 Wearables	11
1.3.4 Wi-Fi	12
1.3.5 RFID	13
1.3.6 VLC	15
1.3.7 Computer vision	15
1.3.8 BLE Beacon	16
1.4 Altre tecnologie di posizionamento	20
1.5 Mappe per il wayfinding	21
1.5.1 Google Maps	22
1.5.2 OpenStreetMap	22

1.5.3	Here Maps	22
1.5.4	Algoritmi	23
2	Specifiche e design	25
2.1	Scenario	25
2.1.1	Utenti non esperti	26
2.1.2	Utenti con disabilità visive	27
2.1.3	Utenti con disabilità motorie	27
2.2	Analisi dei requisiti	28
2.2.1	Sistemi operativi	28
2.2.2	Precisione	29
2.2.3	Estendibilità	29
2.2.4	Accessibilità	30
2.2.5	Personas	30
2.3	Scelte progettuali	32
2.3.1	Tecnologia di posizionamento	32
2.3.2	Framework	35
2.3.3	Librerie	38
2.3.4	Sistema di navigazione	39
2.3.5	Accessibilità	40
2.4	Specifiche di progetto	41
2.5	Prototipo esistente	42
2.5.1	Analogie	43
2.5.2	Differenze	43
3	Implementazione	45
3.1	Team	45
3.2	Ambiente di sviluppo	46
3.2.1	Librerie	46
3.2.2	Permessi	47
3.3	Interfaccia	47
3.4	Formato dati	49

3.4.1	Immagini	50
3.5	Posizionamento	50
3.5.1	GPS	50
3.5.2	Beacon	51
3.6	Calcolo percorso	52
3.7	Funzione "naviga verso..."	53
3.7.1	Calcolo Indicazioni	53
3.7.2	Top bar	56
3.7.3	Mappa	56
3.7.4	Bottone "vicino"	57
3.7.5	Bottom bar	58
3.8	Funzione "intorno a te"	59
3.8.1	Lettura contenuto	60
3.9	Funzione "tour"	60
3.10	Impostazioni	61
3.11	Server	61
3.11.1	Database	62
3.11.2	Interfaccia web	62
4	Test di interazione con l'utente	65
4.1	Human-Computer Interaction	65
4.1.1	User-Centered Design	67
4.2	Venues	73
4.3	Categorie di utenti	74
4.4	Tipologia di valutazione	74
4.5	Feedback	75
4.6	Altri test	76
5	Sviluppi futuri	79
5.1	Mappe SVG	79
5.2	Here Maps	80
5.3	Report	81

5.4	Login Unibo	81
5.5	Analisi statistica	82
5.6	Integrazione con software per la navigazione outdoor	82
	Conclusioni	85
	Bibliografia	87

Elenco delle figure

1.1	Triangolazione, il concetto alla base del sistema GPS	6
1.2	Esempio di percorso tattile	10
1.3	Esempio di pagamento con RFID	14
1.4	Un modello di beacon	16
1.5	Here Maps su smartphone	23
2.1	Mercato dei sistemi operativi per smartphone	28
2.2	Approcci alla localizzazione con i beacon	34
2.3	Applicazioni che utilizzano React Native	36
2.4	Bridging in React Native	37
2.5	Architettura del sistema	42
3.1	Interfaccia grafica	48
3.2	Presentazione POI turistico/culturale	57
3.3	Funzionamento della bottom bar	58
4.1	Experience prototyping durante il progetto ilocalapp	72
4.2	Esempio di percorso di test	74
4.3	Test presso Lettere e Beni Culturali	76
5.1	Form di login dell'Università di Bologna	81

Listings

3.1	Permessi Android	47
3.2	Accessibilità in React Native	48
3.3	Esempio di nodo in formato JSON	49
3.4	Caricamento dati e identificazione edificio	50
3.5	Gestione dell'evento beaconsDidRange	51
3.6	Rimozione degli archi non accessibili	52
3.7	Calcolo della direzione	54
3.8	Prima indicazione	54
3.9	Ultima indicazione	55
3.10	Calcolo percorso completo	58
3.11	Costruzione file JSON	61

Capitolo 1

Posizionamento e navigazione: Una overview

Il recente sviluppo tecnologico dei dispositivi mobili ha abilitato l'implementazione di sistemi di navigazione sia indoor che outdoor con differenti target di utenti e declinati in più forme di assistenza all'individuo. Comunque questi sistemi fanno affidamento al posizionamento tramite GPS e consentono di ricevere indicazioni riguardo al tragitto da seguire per poter raggiungere la propria destinazione. I problemi di questi sistemi sono ben noti in letteratura e riguardano principalmente l'impossibilità di guidare l'utente attraverso spazi dove il segnale GPS non riesce a giungere. Sono un esempio di questa tipologia di luoghi la quasi totalità degli edifici, le gallerie e in generale i siti sotterranei, alcune strade in cui edifici e strutture di varia natura impediscono il propagarsi del segnale GPS in maniera efficace. Dunque un tema di grande interesse è oggi la ricerca di un valido sistema di wayfinding e navigazione che possa funzionare in ambienti dove la localizzazione tramite satellite fallisce.

Lo studio di tecnologie che possano abilitare la navigazione indoor è una materia affrontata da più team di ricerca e ha portato allo sviluppo di nuove tecniche, le quali devono necessariamente utilizzare un nuovo meccanismo di localizzazione e di presentazione delle informazioni. Tale meccanismo non

può prescindere dall'essere compatibile con gli utenti target del sistema ed è fondamentale che sia disponibile all'uso per la più ampia parte della popolazione possibile. Recenti studi sulle peculiarità che ogni utente può presentare e sui possibili scenari di utilizzo del sistema di navigazione, sono stati pubblicati dall'università del Maryland e forniscono linee guida su cui basare la progettazione di tool assistivi compatibili con l'utilizzo da parte di individui con disabilità visive [5]. In questo capitolo prenderemo in esame le principali tipologie di utente finale del sistema per poter implementare un prodotto utile e di ampio utilizzo, con particolare attenzione alle seguenti categorie: utenti con disabilità visive, utenti con disabilità motorie ed in generale utenti non esperti, cioè che non conoscono gli spazi entro i quali si stanno muovendo (es. turisti). Successivamente valuteremo le principali tecnologie e tecniche disponibili per la navigazione e analizzeremo come muovendoci da sistemi outdoor a sistemi indoor sia necessario un cambiamento sostanziale di tecnologie e approcci alla progettazione. Infine considereremo i prodotti disponibili sul mercato per la presentazione di materiale cartografico digitale con attenzione alle licenze e alla compatibilità con mappe sia outdoor che indoor.

1.1 Tipologie di utenti

Un sistema di navigazione deve poter essere utilizzato da chiunque necessiti di indicazioni precise, univoche e real-time per raggiungere una destinazione fisica in tempi ragionevoli e percorrendo tragitti adatti alle proprie esigenze. L'evoluzione tecnologica ha consentito una rapida transizione dagli strumenti cartacei a quelli elettronici come navigatori e smartphone, oggi disponibili sul mercato a prezzi accessibili. In prima battuta questi sistemi si sono limitati ad aggiungere al sistema classico di navigazione tramite mappa, meccanismi di localizzazione automatici, sintesi vocale, e calcolo del percorso più breve. Queste nuove proposte tuttavia non espandono la possibile platea di utenti, continuando a negare la fruizione dei contenuti ad una parte della popolazione. Suddividiamo gli individui in 3 categorie per semplicità:

chi visita un luogo per la prima volta, chi non può vedere mappe e cartelli, chi necessita di percorsi specifici, per capire se sia possibile con le tecnologie odierne rendere la navigazione tramite dispositivi tascabili disponibile per tutti.

1.1.1 Utenti non esperti

E' il target user principale dei navigatori moderni, colui che, pur non avendo nessun tipo di disabilità visiva o motoria, non sa raggiungere autonomamente la sua destinazione perché non ha mai conosciuto i luoghi oggetto della navigazione. Questa categoria di individui rappresenta coloro a cui è stato da subito consentito l'accesso ai navigatori satellitari. Non necessitano di particolari tecnologie assistive e possono scegliere, secondo le loro preferenze, come fruire dei contenuti, se in modo visivo, osservando le mappe proposte, o se ascoltando le indicazioni vocali. Usano il dispositivo come sostituto della mappa cartacea per raggiungere una destinazione stabilita e possono interagire con esso tramite l'interfaccia messa a disposizione (solitamente touch screen). Nel caso della navigazione outdoor generalmente si intendono automobilisti e ciclisti, ma sempre più spesso anche pedoni. Nel caso della navigazione indoor, invece, potrebbero rientrare in questa categoria di utenti turisti alla ricerca di un'opera d'arte all'interno di un edificio storico o cittadini che necessitino di raggiungere un ufficio ubicato in un palazzo della pubblica amministrazione.

1.1.2 Utenti con disabilità visive

Gli utenti con disabilità visive non possono osservare le mappe presentate dal sistema di navigazione, ma possono ricevere indicazioni tramite sintesi vocale o segnali sonori. La difficoltà principale viene riscontrata durante l'interazione uomo macchina, ad esempio, per l'inserimento della destinazione. E' necessario progettare dispositivi dove l'interazione non avvenga solo attraverso uno schermo touch o sistemi analoghi. Una soluzione è quella di

dotare il dispositivo di uno screen reader progettato per supportare l'utente nel compiere, tramite gesture, le operazioni precedenti alla navigazione vera e propria, come ad esempio la scelta della destinazione, la modifica di impostazioni, il setup iniziale, ecc...

Usano il dispositivo per raggiungere una destinazione stabilita e fruiscono delle indicazioni tramite suoni. Se il sistema è preciso può indicare all'utente l'esatto momento in cui svoltare (navigazione turn-by-turn), se non lo è a sufficienza può solamente suggerire la direzione da tenere. Se un individuo senza disabilità visive di solito usa il navigatore come sostituto della mappa, gli utenti con disabilità visiva non fanno uso di questa tecnologia con lo scopo di sostituire gli ausili classici come il bastone o il cane guida, bensì la sfruttano come ulteriore aiuto. Spesso il navigatore è in grado di suggerire il percorso, mentre un cane guida aiuta nell'evitare gli ostacoli che si possono trovare lungo di esso.

1.1.3 Utenti con disabilità motorie

La terza categoria proposta è quella degli utenti con disabilità motorie, intesi come individui che necessitano di ausili per spostarsi come deambulatori o sedie a rotelle. Questi utenti possono interagire con i dispositivi di navigazione senza dover ricorrere a particolari tecnologie assistive, ma necessitano di percorsi specifici che evitino il passaggio in luoghi per loro non accessibili. Nel concreto sono indispensabili software di navigazione che tengano conto dell'ingombro e della conformazione degli ausili e proponano solo tragitti percorribili, ad esempio, nel caso di una navigazione indoor, evitando porte troppo strette per consentire l'accesso con una sedia a rotelle o preferendo l'ascensore alle scale.

1.2 Posizionamento e navigazione outdoor

Nell'immaginario collettivo quando si parla di navigazione ci si riferisce implicitamente alla navigazione tramite sistema di posizionamento satelli-

tare. Il primo satellite che fu lanciato con questo scopo (1959) si chiamava Transit 1A e non raggiunse mai la sua destinazione a causa di un malfunzionamento. Negli anni successivi molti lanci da parte degli Stati Uniti d'America andarono a buon fine e diedero corpo alla flotta di satelliti statunitensi. Il servizio di posizionamento offerto da questo sistema, principalmente sfruttato per uso militare, cessò definitivamente nel 1996 lasciando il campo a tecnologie più avanzate come il GPS. Questa sezione illustra brevemente il funzionamento dei principali dispositivi GPS in quanto principale tecnologia per la navigazione outdoor. Esistono però numerose altre proposte che non proporremo essendo più costose, meno utilizzate e comunque non utili agli scopi di questo lavoro di tesi.

1.2.1 GPS

Il GPS (NAVigation Signal Timing And Ranging Global Position System) [15] è un sistema per il posizionamento e la navigazione di proprietà del governo americano e di uso libero per chiunque possieda un ricevitore apposito. Il progetto nacque negli anni 60 come innovazione rispetto ai precedenti sistemi satellitari e diventò disponibile per uso civile solo nel 1991 quando il dipartimento della difesa statunitense differenziò il servizio erogato in Precision Positioning System per uso militare e Standard Positioning System accessibile alla popolazione, ma con la limitazione di introdurre (intenzionalmente) errori per fornire un posizionamento con precisione massima di circa 1km. Nel 2000 tale limitazione fu rimossa e i sistemi acquisirono una precisione di circa 15 metri. La localizzazione era, ed è tuttora, possibile solo se il dispositivo ha un contatto senza ostacoli con almeno 4 satelliti e riceve il segnale radio emesso da questi ultimi per poi elaborarlo localmente. L'idea alla base di questo sistema è una estensione del concetto di triangolazione che permette di calcolare la distanza tra punti sfruttando le proprietà dei triangoli.

Oggi è possibile raggiungere una precisione di alcuni centimetri, in dipendenza di alcuni fattori come le condizioni meteo [14]. Per questo il GPS viene

frequentemente utilizzato con fini topografici. Tuttavia non tutti i ricevitori sono in grado di abbassare a questi valori l'errore.

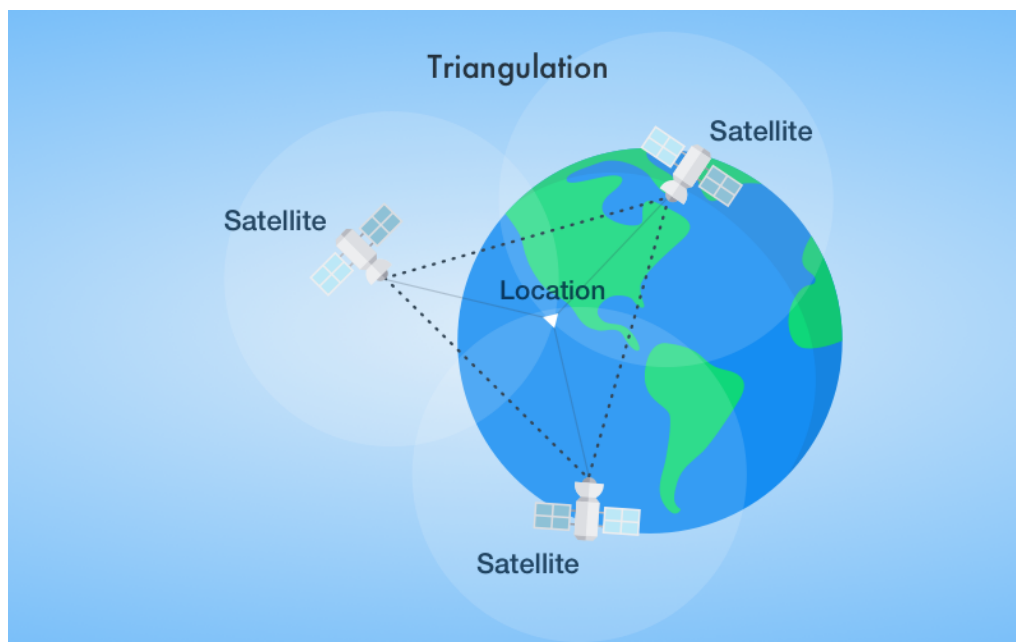


Figura 1.1: Triangolazione, il concetto alla base del sistema GPS

1.2.2 Prodotti in commercio

I ricevitori GPS moderni hanno raggiunto un buon livello di precisione e parallelamente sono scesi a prezzi accessibili rendendo il navigatore satellitare uno strumento di uso comune. Ne esistono diverse soluzioni in commercio:

1. Integrate: sono sistemi all-in-one composti da un ricevitore GPS, un display con input touch, un altoparlante ed un sistema operativo proprietario che, data la mappa generalmente memorizzata su scheda di memoria, sono in grado di calcolare il percorso ottimale per raggiungere la destinazione richiesta.
2. Ibride: si tratta di prodotti informatici come computer e smartphone che incorporano un ricevitore GPS o ai quali sia possibile collegarne uno

esterno. Sono in grado di assolvere a funzioni di navigazione tramite software installabili che possono essere proprietari o open-source.

La quasi totalità dei produttori di smartphone, spinti da un repentino abbassamento dei costi dei ricevitori, ha deciso di inserirne uno nei propri dispositivi. Ciò ha abilitato l'utilizzo di questi device come navigatori satellitari ed ha aperto la strada ad una serie di servizi e applicazioni che fanno del posizionamento il loro fulcro principale. Oggi anche i giochi sfruttano la possibilità di localizzare il giocatore e ciò ha abilitato, tra le altre cose, interessanti funzioni di gamification [16, 17]. Inoltre l'iniziale scoglio dell'alto tempo di localizzazione al momento dell'attivazione del GPS (anche oltre ad 1 minuto), è stato superato grazie all'introduzione dell'Assisted-GPS che, attraverso la rete telefonica, è in grado di ricevere le informazioni sui satelliti visibili dalla cella a cui il device è agganciato. Supponendo che questi satelliti siano gli stessi visibili dallo smartphone, in pochi secondi è possibile ottenere la propria posizione senza aver performato esplicitamente la ricerca dei vari segnali radio.

I primi sistemi all-in-one a conquistare il mercato sono stati TomTom e Garmin che ancora oggi commerciano prodotti nel campo della localizzazione e della navigazione. Successivamente molte case automobilistiche hanno sviluppato il proprio sistema di navigazione per poi fornirlo preinstallato sulle vetture. Infine sono state proposte applicazioni per smartphone, alcune proprietarie come Google Maps ed altre basate su mappe open-source (approfondimento in seguito). Il più famoso e completo progetto open source in questo ambito è certamente OpenStreetMap [18] che mette a disposizione il proprio database a chiunque voglia implementare un nuovo sistema di navigazione. Navigatori satellitari GPS possono, con le tecnologie odierne essere implementati per consentirne l'uso a tutte le tipologie di utente citate in precedenza. In particolar modo i moderni smartphone, dotati di screen reader e supporto per le gesture, consentono anche a utenti con disabilità visive di fruire del servizio. Resta spesso irrisolto il problema della accessibilità per utenti con disabilità motorie perché, pur essendo possibile il calcolo

di percorsi appositi (date le indicazioni sull'accessibilità dell'area coinvolta), resta in pratica impossibile monitorare costantemente la situazione di tutti i percorsi pedonali esistenti. Generalmente si considera verosimile che in ogni città moderna siano presenti facilitazioni per chi può incontrare difficoltà.

1.3 Posizionamento e navigazione indoor

L'abitudine all'utilizzo di sistemi GPS e il crescente bisogno di localizzare le persone all'interno di luoghi pubblici ha, negli anni, spostato l'attenzione dalla navigazione outdoor a quella indoor. Come già noto, le onde radio dei satelliti si propagano per enormi distanze ma, giunte in prossimità dell'utente possono incontrare ostacoli, quali muri, tetti, fitta vegetazione che ne impediscono la ricezione. Per questo all'interno degli edifici è inverosimile sperare di ricevere un segnale costante e sufficiente per traslare i servizi di navigazione pensati per l'esterno verso ambienti chiusi. E' dunque necessario, per localizzare accuratamente gli individui al chiuso, trovare metodi e tecnologie alternative. Esistono varie proposte più o meno tecnologiche, ognuna delle quali utile a particolari scopi di navigazione, wayfinding, localizzazione, ausilio, ecc...

Di seguito prenderemo in considerazione le principali proposte esaminando pro e contro di ognuna, nell'ottica di definire quali tra quelle conosciute siano al giorno d'oggi indispensabili, quali utili e quali superflue o superate traendo da ognuna spunti utili per la progettazione di un sistema moderno, funzionale ed efficiente per la localizzazione e navigazione indoor fruibile da ognuna delle tipologie di utente proposte in questo lavoro di tesi.

1.3.1 Cartellonistica

La cartellonistica è il primo sistema che si adotta nella totalità degli edifici pubblici per consentire al visitatore di trovare gli ambienti nei quali quest'ultimo necessita di recarsi. E' un sistema di facile progettazione e realizzazione, può avvalersi, oltre che di scritte, di simboli e di colori. Non necessita di alcu-

na moderna tecnologia ed è facilmente estendibile con l'aggiunta di ulteriori segnali. Tuttavia presenta delle problematiche.

1. La cartellonistica non può tenere conto di ogni possibile destinazione che un individuo potrebbe voler raggiungere. Ciò suggerisce che con questo tipo di sistema sia possibile coprire solo una parte (seppur la più significativa) delle destinazioni.
2. Attraverso i cartelli non è possibile fornire grandi quantità di informazioni aggiuntive a causa dell'esigenza di mantenere semplice e intuitivo questo tipo di supporto alla navigazione. Ciò implica che eventuali informazioni riguardo all'ambiente (es. informazioni su una statua incontrata lungo il percorso, orari di apertura, ecc...) dovranno essere tralasciate o riportate su ulteriori supporti col rischio di introdurre disordine e di limitare particolari categorie di individui (es. stranieri che necessitano di simbologia standard come frecce e colori)
3. Tutti gli individui con disabilità visive non saranno in grado di usufruire della cartellonistica a meno che essa non sia disponibile in formato braille, introducendo ulteriore complessità. Inoltre indicazioni scritte in braille assumono che l'individuo sappia come raggiungere l'indicazione stessa per poterla consultare.

1.3.2 Percorsi tattili

Dato per scontato che una qualche forma di cartellonistica essenziale sia disponibile in tutti gli edifici pubblici è necessario occuparsi della parte di popolazione che non trova giovamento dalla sua presenza. Cioè coloro che non sono in grado di usufruirne o coloro che pur utilizzandola non trovano la soluzione alle proprie esigenze.

I percorsi tattili (in inglese tactile paving), proposti in Giappone nel 1965 da Seiichi Miyake ed attualmente utilizzati in tutto il mondo, ad esempio, cercano semplificare gli spostamenti entro un determinato spazio ad individui

con disabilità visive [1]. Sono stati adottati in primis dal trasporto ferroviario nazionale del Giappone e, ad oggi, in Europa vengono applicati in molti edifici pubblici ed anche all'esterno per agevolare i pedoni con disabilità. Tali percorsi possono comunicare differenti informazioni grazie alla forma della loro superficie, come ad esempio la presenza di un attraversamento, un percorso dritto, una barriera, ecc... Facilitano l'individuazione di ostacoli, svolte e percorsi sicuri, ma non possono essere adottati come unico sistema.

1. Indicano che quello su cui si sta camminando è un percorso valido, ma non sono in grado di indicare dove questo percorso porti.
2. Sono costosi e necessitano di un accurato lavoro di progettazione e posa.
3. Aiutano le persone con disabilità visive, ma non le altre categorie in esame.

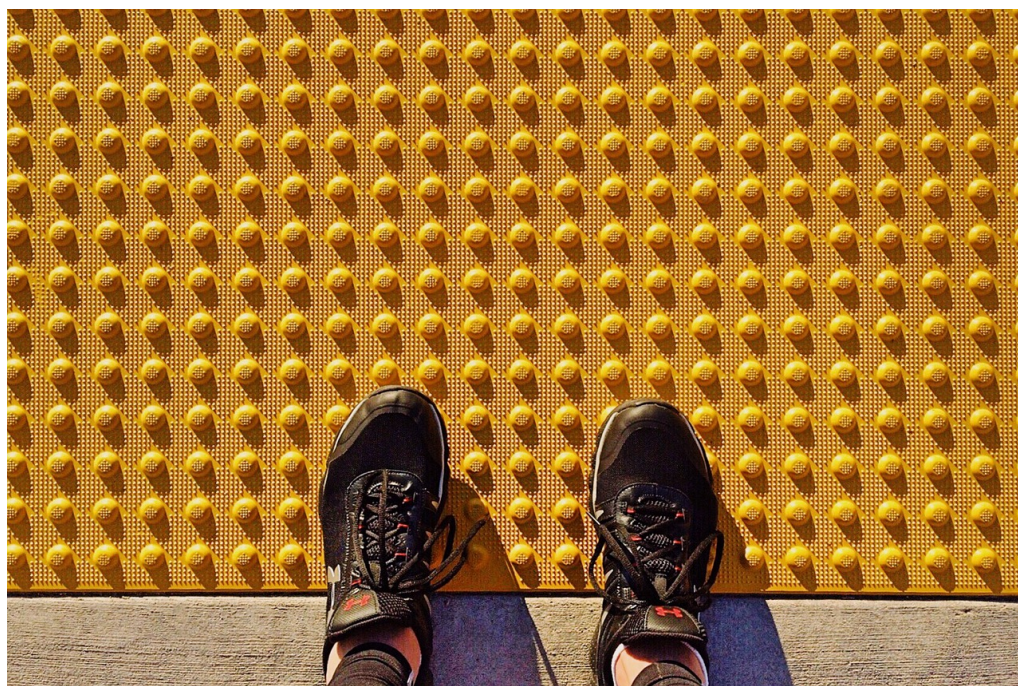


Figura 1.2: Esempio di percorso tattile

1.3.3 Wearables

Mantenendo il focus sugli utenti con disabilità visive, un ampio numero di tecnologie indossabili sono state proposte dagli accademici di tutto il mondo. Linee guida per lo sviluppo di sistemi indossabili accessibili sono state proposte alla comunità scientifica nel 2016 con il paper "Wearables for All: Development of Guidelines to Stimulate Accessible Wearable Technology Design" [6], dove particolare accento viene posto sulla necessità di una presentazione multimodale delle informazioni all'utente, il quale deve poter anche modificare le impostazioni del software per ottenere un set-up a lui congeniale. Ulteriore aspetto di non trascurabile impatto è la gestione di comportamenti inattesi o sbagliati dell'utente.

La quasi totalità dei wearables sono progettati a partire da un dispositivo vibrotattile [7, 8, 9, 10, 11], per poi essere estesi con ulteriori sensori e meccanismi per fornire una esperienza di navigazione ancora più precisa e che risponda ad un numero più ampio possibile di bisogni. E' il caso del lavoro "An Assistive Navigation Framework for the Visual Impaired" [12], che riesce a integrare il dispositivo vibrotattile all'interno di un ecosistema di sensori (pedometro, sensore RGB-D, GPS, ecc...) ed attuatori (auricolari, display, ecc...) presentando una complessa struttura che, controllata da un device mobile connesso, è anche in grado di utilizzare i dati estratti via API dai social network (Twitter), aumentando per l'utente la percezione globale dell'ambiente circostante e dei recenti eventi che lo influenzano.

Sebbene questa categoria di ausili offra una vasta gamma di prodotti possiamo sintetizzare le sue proprietà come segue.

1. Non necessitano di modifiche fisiche al luogo dove vengono utilizzati.
2. Non devono conoscere a priori la mappa del luogo, la scaricano dalla rete al bisogno.
3. Il costo del sistema è a carico dell'utente che deve acquistare un dispositivo già funzionante o realizzare il proprio.

4. Generalmente questi ausili richiedono connettività internet e una ampia batteria per poter fornire aiuto per un tempo adeguato.
5. Spesso i wearables aiutano l'utente nell'evitare gli ostacoli che si trovano sul cammino, ma non sempre sono implementati per poterlo aiutare a raggiungere una specifica destinazione con indicazioni turn-by-turn.
6. Ambienti dinamici e con molte persone possono risultare di difficile interpretazione da parte del sistema.

Percorsi tattili e dispositivi wearable sono dunque soluzioni costose che, nel primo caso, implicano modifiche all'ambiente e, nel secondo, obbligano l'utente ad una investimento personale. Inoltre entrambe le soluzioni nascono per persone con disabilità visive e solo per esse trovano una vera applicazione, non considerando perciò le altre due categorie di utenti individuate nell'ambito di questa tesi, cioè utenti con disabilità motorie e coloro che non hanno disabilità, ma necessitano di un sistema di navigazione indoor. Per questo altre tecnologie hanno trovato applicazione nel mondo della navigazione turn-by-turn e sono state utilizzate per fornire un supporto all'utente tramite applicazioni per smartphone.

1.3.4 Wi-Fi

Avere a disposizione un sistema di navigazione sul proprio dispositivo mobile significa non costringere l'utente ad acquistare attrezzature. Nell'ottica di contenere la spesa, anche da parte di chi mette a disposizione l'edificio accessibile, esistono proposte che basano il loro funzionamento sulla presenza di segnali Wi-Fi (Wireless Local Area Networks) tipicamente presenti in ogni luogo di pubblico utilizzo [13]. La limitazione di questo approccio è che la distribuzione degli access point è studiata per una corretta fruizione di contenuti via internet e non per la localizzazione. Ciò può portare ad avere zone dell'edificio dove risulta impossibile fornire una assistenza turn-by-turn anche integrando sensori supplementari come accelerometro e bussola, che

spesso assumono un ruolo di supporto al sistema di localizzazione principale adottato. Possiamo riassumere pro e contro di questa soluzione.

1. Wi-Fi già presente negli edifici pubblici.
2. Basso costo per utente e fornitori del servizio.
3. Segnale che, specialmente nelle svolte, può non essere sufficiente per fornire un corretto posizionamento.

1.3.5 RFID

Anche quella degli RFID (Radio-Frequency IDentification) è una tecnologia nata in ambito militare che ha trovato applicazione nel mondo civile. Sfrutta le frequenze radio per effettuare comunicazione senza fili tra un tag e un ricevitore. Il tag è un elemento dotato di antenna che memorizza al suo interno una piccola quantità di informazioni e può essere passivo se non necessita di batteria e si attiva con la sola energia fornita dalle onde radio che lo interrogano, oppure attivo se dotato di una qualche forma di alimentazione. Questa seconda tipologia può avere distanze operative che raggiungono i 200 metri, contro i pochi centimetri della prima. Questa tecnologia ha trovato vari utilizzi.

1. Come sostituzione dei codici a barre sui prodotti, non necessitando di essere visibili per essere letti.
2. Per biglietti e abbonamenti del trasporto pubblico.
3. Per le carte bancarie contact-less.
4. Per il servizio Telepass (tag attivi) del gruppo Autostrade.

Tuttavia non sono mai stati adottati in larga misura in sistemi di navigazione per alcune limitazioni tecnologiche, in particolare per la necessità che il lettore passi ad una ridottissima distanza dal tag per poterlo rilevare. Questo ha permesso però di realizzare tracce sul terreno composte da file di tag che

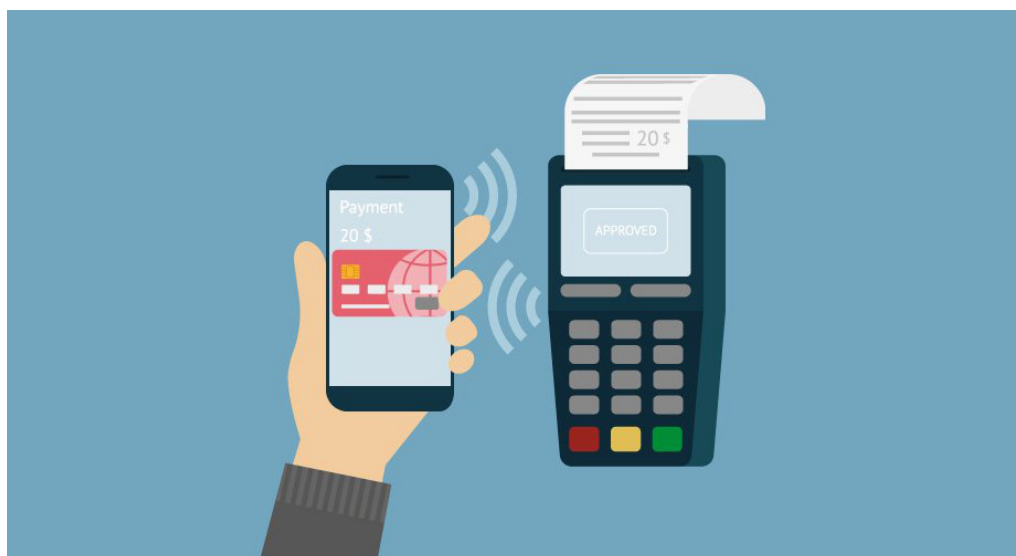


Figura 1.3: Esempio di pagamento con RFID

vengono sfruttate ad esempio da robot per spostarsi all'interno di magazzini e impianti di logistica. Un approccio simile è stato proposto in rari casi come ausilio agli utenti con disabilità visive [20, 21]. Si tratta di una soluzione che prevede un percorso fatto di tag sul terreno (ad esempio lungo un percorso tattile) ed un lettore opportunamente installato in cima ad un bastone e collegato ad un dispositivo mobile. Ciò funziona solo per questa tipologia di utenti ed assume che il tracciato venga seguito con una alta precisione.

1. Questa tecnologia ha, come vantaggio, un basso costo di tag e lettori.
2. Necessita di interventi di modifica all'ambiente per il fissaggio sul terreno (tipicamente sotto al pavimento).
3. Fornisce un reale aiuto solo ad utenti con disabilità visive.
4. Non può prescindere dall'uso del bastone apposito.

1.3.6 VLC

E' possibile sfruttare la luce visibile per ricavare la posizione del device [2, 24]. VLC (Visible Light Lommunication) è un sistema composto da LED debitamente etichettati per essere riconosciuti dallo smartphone, il quale attraverso la fotocamera rileva le fonti di luce e in base a vari parametri come l'angolo di ricezione può calcolare la sua stessa posizione.

1. E' una soluzione economica considerato il costo dei LED.
2. Con i giusti accorgimenti per una interfaccia accessibile sono utilizzabili da chiunque.
3. Richiede l'utilizzo costante della fotocamera dello smartphone con conseguente consumo di batteria.
4. Il dispositivo deve essere tenuto nella corretta posizione.
5. Il contatto visivo LED-Device potrebbe essere interrotto da elementi esterni.

1.3.7 Computer vision

Quelle di computer vision sono tecniche di difficile implementazione che necessitano di accurati algoritmi per l'identificazione di oggetti nell'ambiente. Sfruttano una fotocamera, ad esempio quella dello smartphone, e localizzano l'utente basandosi su tag visibili.

1. Aspetto positivo di questa tecnologia è quello di poter rilevare oggetti in tempo reale, magari appena aggiunti all'ambiente.
2. Tuttavia il problema del consumo di batteria è particolarmente accentuato.
3. Può essere usata come tecnologia assistiva.
4. Occlusioni in spazi affollati sono inevitabili.

1.3.8 BLE Beacon

I beacon sono piccoli dispositivi che consentono agli smartphone di captare la loro stessa presenza in un raggio di qualche metro. Si tratta di oggetti che trasmettono periodicamente una piccola quantità di dati in un'area che può arrivare ad alcune decine di metri. I beacon sono spesso utilizzati per la localizzazione indoor come alternativa al Global Positioning System e ne sono stati proposti diversi tipi che differiscono tra loro per protocolli utilizzati, alimentazione, tecnologie di trasmissione, ecc...



Figura 1.4: Un modello di beacon

Alimentazione

La metodologia di alimentazione di questi oggetti dipende esclusivamente dalle scelte costruttive del produttore e generalmente può essere suddivisa in 3 categorie.

1. Batterie: contengono batterie a bottone, AA o AAA, generalmente sostituibili e sono adatti per installazioni provvisorie o dove non sia possibile creare collegamenti alla rete elettrica. La durata della batteria

dipende dal settaggio del beacon stesso e può andare da un mese a svariati anni.

2. Una minoranza di beacon è progettata per essere alimentata direttamente dalla rete elettrica, risolvendo così il problema della sostituzione delle batterie. L'installazione richiede però la vicinanza ad una presa di corrente, non sempre possibile.
3. L'alimentazione tramite USB è un terza soluzione che può trovare applicazione nel caso in cui il beacon si trovi nei pressi di un dispositivo con con porte USB alimentate e si voglia sfruttare queste ultime come fonte di energia. Possono trovare applicazione anche in fase di testing.

Protocolli

Esistono diversi protocolli per i beacons, cioè diverse tipologie di messaggio che vengono emesse nel raggio di azione del dispositivo.

1. iBeacon.

È il primo protocollo per beacon immesso sul mercato da Apple nel dicembre del 2013. iBeacons è compatibile sia con dispositivi Apple che Android e funziona trasmettendo una particolare stringa di 24 caratteri detta UUID (Universally Unique Identifier) standardizzata da Open Software Foundation [19].

- Questo protocollo è il più utilizzato ed è ampiamente supportato.
- Semplice nell'utilizzo e nell'implementazione.
- Essendo sviluppato da Apple le performance sui dispositivi con sistema operativo iOS sono certe.

2. AltBeacon.

Radius Networks ha proposto il suo protocollo a luglio del 2014. Questa azienda ha voluto puntare sull'open-source per risolvere il problema dei protocolli che favoriscono un produttore rispetto agli altri.

- E' un protocollo completamente open-source.
- Mira alla piena compatibilità con tutti i sistemi operativi mobili.
- E' flessibile e consente la personalizzazione del codice sorgente per soddisfare necessità particolari.

3. Eddystone.

Introdotta da Google a luglio del 2015 con il nome di UriBeacon e poi rinominata in Eddystone. I beacon che supportano questo protocollo sono in grado di gestire tre differenti tipi di data frame in maniera compatibile con i sistemi operativi iOS e Android. Il singolo beacon può essere abilitato alla trasmissione di uno, due o di tutti e tre i tipi di frame. I frametypes sono:

- URL: un URL è inviato al device, eliminando la necessità di app apposite installate sullo stesso.
- UID: simile all'UUID di Apple, è composto da 16 caratteri e identifica il singolo beacon. Può attivare app specifiche installate sul device.
- TLM: dati di supporto per la gestione dei beacon stessi, inclusi livello della batteria, temperatura, ecc...

Le proprietà di questo protocollo sono:

- Possibilità di lavorare senza app preinstallate mediante gli URL.
- Ottima integrazione con i prodotti Google.
- Flessibilità e formati open.

4. GeoBeacon.

Tecno-World ha annunciato l'uscita del suo protocollo a luglio 2017. E' completamente open-source ed è stato progettato nell'ambito di applicazioni GeoCaching.

- E' un protocollo open source.

- Compatibile con vari sistemi operativi.
- Supporta coordinate ad alta precisione.
- Supporta TLM per monitorare il livello della batteria.
- Possibilità di aggiungere 8 Bytes di dati utente.

Tecnologie

La quasi totalità dei beacon si affida al BLE (Bluetooth Low Energy) che può garantire consumi di energia nettamente inferiori rispetto al Bluetooth classico, pur avendo un minor range ed una trasmissione dati ridotta. Il BLE è infatti progettato appositamente per l'invio periodico di piccoli frammenti di dato. Esiste una simile proposta anche per il Wi-Fi chiamata Wi-Fi Aware. Anche essa si basa sul basso consumo energetico.

Progetti

I BLE-Beacon, per le loro caratteristiche come raggio di azione, consumo energetico e dimensioni, hanno suscitato da subito interesse nella comunità scientifica e nel mondo delle aziende e numerosi sono i progetti che si basano o fanno uso di questi prodotti. I beacon sono stati utilizzati come attivatori di servizi ad esempio da McDonald's che con la sua strategia di proximity marketing ha premiato i clienti che entravano in un McCafé offrendo loro un prodotto. In campo museale sono stati usati, accanto alle opere d'arte, come metodo per individuare il punto di interesse più vicino al visitatore fornendo a quest'ultimo informazioni multimediali riguardanti l'oggetto e la sua storia. Per la navigazione, outdoor ma soprattutto indoor considerando la portata del segnale, sono stati utilizzati in svariati progetti sia a livello di ricerca che aziendale per localizzare l'utente e indicargli la giusta direzione o per monitorarne la posizione nel tempo. Meno frequentemente sono stati alla base dello sviluppo di sistemi di wayfinding più sofisticati che prevedano indicazioni turn-by-turn [22], specialmente con attenzione a tutte le tipologie di utente che potrebbero richiederne l'uso. La navigazione turn-by-turn

supportata dalla tecnologia dei beacons e fruibile da individui con disabilità visive è stata studiata e corredata di pubblicazioni scientifiche, ad oggi, da 2 gruppi di ricerca.

1. Il primo composto da Karen Duarte, José Cecílio e Pedro Furtado ha proposto una app per soli dispositivi Android presentata alla International Wireless Internet Conference del 2014 [23].
2. Il secondo composto da Dragan Ahmetovic, Cole Gleason, Chengxiong Ruan, Kris Kitani, Hironobu Takagi e Chieko Asakawa ha proposto una app per soli dispositivi iOS presentata nel 2016 [3, 4] e conosciuta come NavCog.

Entrambe le soluzioni si focalizzano solo sull'utente con disabilità visive e ognuna delle due proposte è stata sviluppata per solo una piattaforma mobile. Per localizzare l'utente utilizzano metodi simili con alla base il RSSI (Received Signal Strength Indication). NavCog si basa su quello che viene definito un fingerprint di RSSI. In pratica, durante l'installazione del sistema, ad ogni metro viene rilevato l'array di intensità dei segnali di tutti i beacons che poi sarà confrontato runtime con l'array rilevato dal device in ogni istante. La corrispondenza tra due fingerprint indica la posizione con un errore di 1,5 metri. SmartNav invece rileva l'RSSI e computa in tempo reale una stima della posizioni basata solo sulla lettura attuale. Queste tecnologie sono facilmente applicabili a diversi contesti e diverse tipologie di utente, sviluppando apposite interfacce.

1.4 Altre tecnologie di posizionamento

Esistono altre tecnologie di posizionamento, che in questo lavoro di tesi ci limiteremo a citare, che possono essere utilizzate per realizzare sistemi di navigazione sia indoor che outdoor con diverse performance in termini di disponibilità e precisione. Una panoramica su queste tecnologie è stata

pubblicata nel 2017 dal dott. Alessio De Vita nell'ambito del suo progetto di tesi presso Unibo [25].

1. Cell Identification: disponibile per ogni telefono connesso alla rete. Indica la posizione in base alla cella a cui il telefono è collegato. Ha una bassa precisione (200 metri / 10Km) che dipende dalla densità delle celle.
2. Time of Arrival: con precisione di 10 / 100 metri sfrutta la triangolazione del segnale proveniente da varie celle e si basa sul tempo, necessitando perciò di orologi sincronizzati.
3. Observed Time Difference: simile a TOA, lascia al device il calcolo anziché alle celle dell'operatore.
4. Direction or Angle of Arrival: sempre basato sulla rete, fonda il suo funzionamento sulla direzione dei segnali e non più sul tempo.
5. Ultrasuoni: questi sistemi sono composti da un emettitore e da una matrice di ricevitori (o il contrario), possono offrire una buona precisione indoor (anche poche decine di centimetri), ma sono di difficile installazione e risentono della presenza di oggetti che possono far rimbalzare il segnale in modo imprevedibile.

1.5 Mappe per il wayfinding

Il wayfinding è il *"sapere dove si è in un edificio o in generale in un ambiente, sapendo dove si trova la posizione desiderata e sapendo come arrivare ad essa a partire dalla posizione attuale. Il wayfinding è dunque il modo in cui le persone possono essere aiutate per sostenere o per indirizzare il loro orientamento. L'orientamento è appunto la capacità di determinare la propria e l'altrui posizione rispetto ad un quadro concettuale di riferimento spaziale, nonché una disposizione ad affrontare ambienti e persone sia noti che sconosciuti."* [25]. Il principale strumento per il wayfinding sono le

mappe, utilizzate da larga parte delle tecnologie di localizzazione precedentemente citate per abilitare la navigazione. Le mappe infatti consentono di rappresentare porzioni del territorio includendo itinerari, punti di arrivo e partenza, elementi di interesse, ecc... La cartografia digitale ha incontrato in tempi recenti l'interesse della comunità informatica e delle aziende che hanno proposto interi framework per lo sviluppo di sistemi in tale ambito.

1.5.1 Google Maps

E' un servizio per la ricerca e visualizzazione di carte geografiche che integra la possibilità di conoscere la posizione di luoghi di interesse come ristoranti, musei, ecc... Propone mappe stradali e un algoritmo di route planning per utenti a piedi, in bici, in macchina, o che viaggiano con mezzi pubblici. E' di proprietà di Google e nasce per un utilizzo outdoor, anche se esiste un servizio di mappe indoor pensato per i grandi luoghi pubblici.

1.5.2 OpenStreetMap

E' un progetto collaborativo e open-source. OSM consente l'accesso al proprio database di mappe in modo libero e propone un sistema attraverso il quale gli utenti possono proporre modifiche mantenendo aggiornato il sistema. Sono disponibili tool per il caricamento di mappe indoor, opzione che ha contribuito alla realizzazione di progetti di successo [26].

1.5.3 Here Maps

Here Maps nasce come ventaglio di servizi per la mappatura nel mondo automobilistico e aziendale e propone una sua applicazione di navigazione unitamente ad una serie di librerie ed API per lo sviluppo ulteriore software. Compatibile con Android e iOS permette l'utilizzo delle mappe presenti sul database o il caricamento di nuove porzioni anche per l'indoor. Interessante il tool per l'indoor wayfinding con supporto a tecnologie di localizzazione Wi-Fi e Bluetooth.

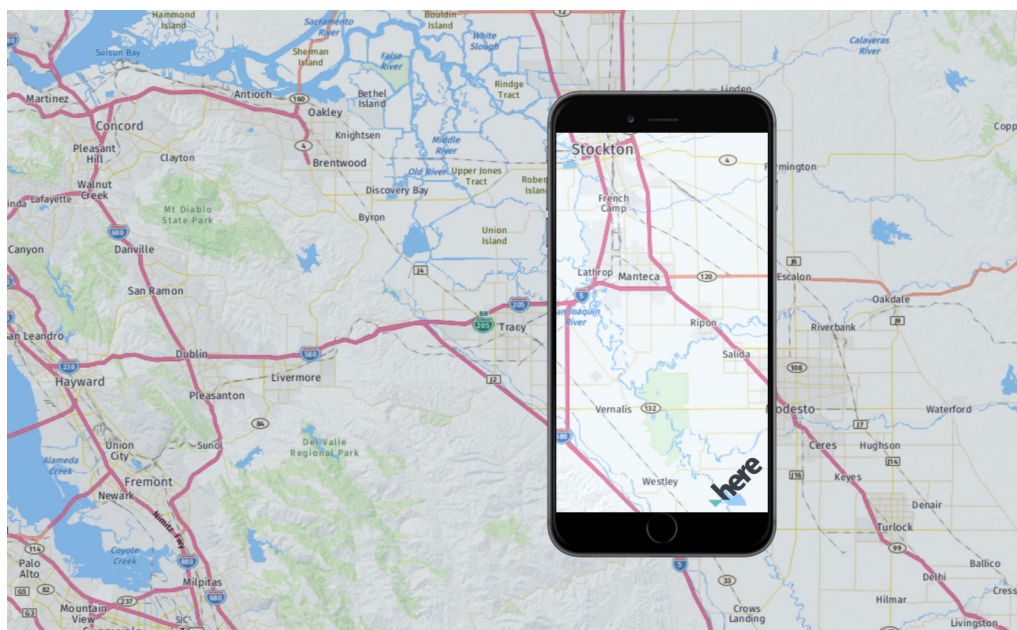


Figura 1.5: Here Maps su smartphone

1.5.4 Algoritmi

I servizi di routing messi a disposizione dai sistemi citati hanno lo scopo di ottimizzare il percorso fornendo il tragitto più breve/veloce possibile tenendo conto del mezzo di trasporto. L'algoritmo più diffuso per compiere questo task è quello di Dijkstra, in grado di risolvere il problema del cammino minimo su un grafo orientato e pesato. Le mappe infatti vengono solitamente intese come grafi dove le intersezioni sono i nodi e i collegamenti sono gli archi ai quali è assegnato un peso.

Capitolo 2

Specifiche e design

In questo capitolo verranno presentati gli scenari che hanno suggerito l'implementazione della applicazione di wayfinding ed attraverso i quali si sono potuti delineare alcuni requisiti della stessa. Tali requisiti funzionali e non, con le dovute integrazioni, verranno poi esposti. Sono successivamente elencate e motivate le scelte progettuali prese nel rispetto dei requisiti formulati. E' poi riportata una sezione di specifiche di progetto a più basso livello che comprendono e sintetizzano i requisiti iniziali alla luce delle scelte progettuali sostenute.

2.1 Scenario

Questa sezione illustra le motivazioni per cui è stata ritenuta necessaria l'implementazione di una applicazione mobile per la navigazione indoor, suddividendole in tre categorie, sulla base dei profili degli utenti che esprimono tali necessità. Questi tre insiemi sono riconducibili ai tre presentati nel primo capitolo come possibili gruppi di utenti di un generico sistema di navigazione. Di seguito contestualizzeremo questa suddivisione all'interno di un progetto per la navigazione turn-by-turn indoor con aspetti turistico/culturali paralleli a quelli di wayfinding.

2.1.1 Utenti non esperti

Gli edifici universitari del quartiere studentesco del centro di Bologna ed in generale i palazzi storici, sono ricchi di storia e spesso contengono opere d'arte e caratteristiche architettoniche in grado di attrarre l'interesse di turisti e curiosi. Non trattandosi però di percorsi museali veri e propri non vengono offerti servizi che promuovano questo tipo di visita. E' quindi frequente notare turisti, in particolare stranieri, chiedere informazioni rispetto a mezzi busti, quadri, ecc... a chi questi edifici li vive giornalmente per lavoro o per motivi di studio. E' facile immaginare a tal proposito uno strumento digitale in grado di fornire dettagli riguardo agli oggetti e ai luoghi di interesse al turista, il quale deve poterne fruire in una lingua a lui comprensibile e senza interferire con il personale al lavoro negli stessi ambienti. Questa tipologia di utente può essere ulteriormente estesa considerando anche coloro che pur frequentando assiduamente un particolare ambiente, non si sono mai interessati alla storia racchiusa tra i suoi muri, spesso per la semplice mancanza di un sistema di erogazione delle informazioni, per esempio sotto forma di didascalie. Durante lo sviluppo della app si dovranno dunque considerare le seguenti situazioni:

- Un turista entra in un palazzo storico dell'Università di Bologna e utilizza la app per raggiungere un punto di interesse che sa essere all'interno dell'edificio, per poi leggerne la storia.
- Un turista si trova all'interno dell'edificio e nota un'opera d'arte. Avvia la app e riceve le informazioni sul patrimonio artistico che si trova di fronte a lui.
- Uno studente necessita di ricevere indicazioni per raggiungere lo studio del suo professore. Utilizza la app per la navigazione e, durante il percorso, si sofferma a leggere la storia di un quadro accanto al quale è passato.

2.1.2 Utenti con disabilità visive

I palazzi universitari, edificati in più epoche ed adattati e riconvertiti nel tempo, possono presentare una piantina complessa. Tale complessità unita alla elevata quantità di ambienti, studi, aule e corridoi rappresenta una difficoltà per chi non conosce l'ubicazione dell'ambiente dove deve recarsi. La situazione di difficoltà si accentua nel caso di individui con disabilità visive che non possiedono ancora una mappa mentale del susseguirsi degli spazi. Un navigatore con funzione turn-by-turn consente contemporaneamente di raggiungere la destinazione e di cominciare a conoscere la conformazione del palazzo. L'applicazione dovrà quindi rispondere positivamente in situazioni come la seguente:

- Uno studente cieco in compagnia del suo cane deve recarsi in segreteria. Avvia la app e imposta la destinazione aiutato dallo screen reader. Segue le indicazioni vocali fino all'arrivo.

2.1.3 Utenti con disabilità motorie

L'evolversi delle normative ha richiesto adeguamenti in termini di sicurezza e accessibilità imponendo l'installazione di rampe, ascensori e le predisposizioni di percorsi agevoli per individui che fanno uso di sedie a rotelle. E' però possibile che non tutti i tragitti percorribili per raggiungere una determinata posizione siano accessibili. Il sistema di navigazione dovrà tenere conto quindi della accessibilità/non accessibilità dei percorsi per poter essere utilizzata anche in casi come il seguente:

- Un ospite, che può muoversi solo con la sedia a rotelle, deve recarsi in sala riunioni per partecipare ad un meeting. La applicazione lo guida verso un percorso che evita le scale e fa uso di ascensori con larghezza sufficiente, per fargli raggiungere il locale selezionato.

2.2 Analisi dei requisiti

Per poter creare una applicazione che sia utile ed efficiente in tutte le situazioni presentate è necessario scegliere con attenzione alcune delle caratteristiche e delle proprietà che il sistema dovrà avere. Tali dettagli sono di seguito riportati astraendo, dove possibile, dal linguaggio, dal framework, e da tutti gli aspetti implementativi di basso livello.

2.2.1 Sistemi operativi

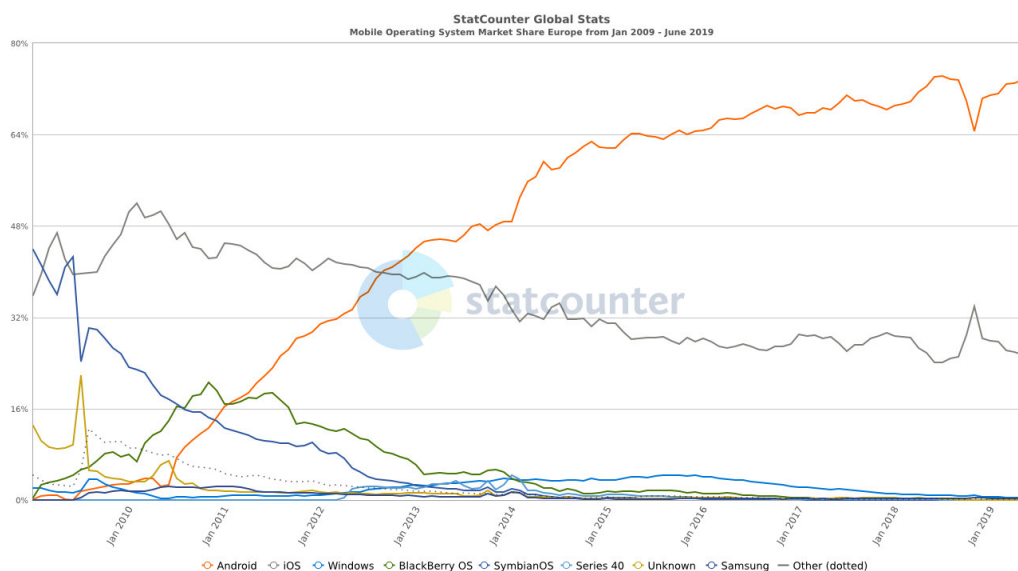


Figura 2.1: Mercato dei sistemi operativi per smartphone

Il mercato degli smartphone è in costante evoluzione e gli acquirenti richiedono funzionalità e caratteristiche sempre più specifiche promuovendo lo sviluppo dell'hardware e del software dei dispositivi, ognuno dei quali con peculiarità, caratteristiche e performance differenti. Ciò che invece tende a stabilizzarsi è la distribuzione dei sistemi operativi. Fino al 2011 infatti varie proposte si spartivano più o meno equamente il mercato, ma l'ascesa del sistema Android e il confermarsi di iOS hanno modificato il grafico delle

installazioni spostandolo verso una situazione di quasi stabilità che si è mantenuta negli ultimi 4 anni. Android è installato su circa il 75% dei device e iOS sul restante 25% (dati Global Stats). Appare subito evidente come rinunciare ad una di queste due piattaforme implicherebbe la non disponibilità del servizio per una ampia parte della popolazione. Uno dei requisiti è dunque la presenza sui principali store di entrambe le aziende leader.

2.2.2 Precisione

La precisione è un aspetto cruciale per i sistemi di navigazione sia outdoor che indoor ed è stata considerata come uno degli aspetti chiave del progetto. La precisione viene influenzata da innumerevoli aspetti come tecnologia di posizionamento, approccio algoritmico, ambiente, e fattori imprevisti. La app deve poter fornire un aiuto concreto in fase di navigazione e consentire lo spostamento negli edifici senza introdurre ambiguità nelle indicazioni o incertezza sulla direzione da tenere. Inoltre l'utilizzo da parte di individui con disabilità visive deve avvenire con modalità finalizzate a semplificare e agevolare gli spostamenti. Il prodotto però non deve mirare a sostituire gli ausili classici come cane guida o bastone, deve infatti proporsi come aiuto e integrazione agli stessi. Dunque la precisione deve essere tale da individuare correttamente le svolte lungo il percorso. L'errore non dovrà mai essere superiore ai 2 metri e, quando possibile, scendere a valori inferiori.

2.2.3 Estendibilità

Essendo impossibile la produzione di mappe e l'installazione di sensoristica per tutti gli edifici storici dell'Università di Bologna in fase di sviluppo, requisito fondamentale è l'estendibilità intesa come possibilità di integrare nuove mappe riferite a edifici aggiunti in momenti successivi al completamento dell'applicazione. Tale aggiunta di contenuti dovrà essere possibile anche da parte di personale non esperto di tecnologia, cioè da individui sen-

za conoscenze informatiche ai quali dovrà essere consentito di produrre e pubblicare materiale con modalità semplici e intuitive.

2.2.4 Accessibilità

Accessibilità: ” *In ingegneria informatica, proprietà che devono possedere le applicazioni per essere utilizzate con facilità dagli utenti, in particolare da coloro che si trovano in condizioni di disabilità o di svantaggio.*” [35]. Tali proprietà si riferiscono principalmente all’interazione uomo-macchina che potrebbe avvenire con modalità specifiche. Proprietà che dovranno essere garantite per tutti, con particolare attenzione agli utenti con disabilità visive. Gli utenti che fanno uso, ad esempio, di sedie a rotelle dovranno aver garantito un diverso tipo di accessibilità, cioè la certezza di venire indirizzati su percorsi accessibili ovvero adatti al transito con tale tipo di ausilio.

2.2.5 Personas

Le scelte progettuali che verranno presentate nelle sezioni seguenti si basano sulle specifiche già esposte, le quali sono in larga parte riconducibili a due *personas*, riportate nella pubblicazione ”AlmaWhere: a Prototype of Accessible Indoor Wayfinding and Navigation System” a cura di Giovanni Delnevo, Lorenzo Monti, Francesco Vignola, Paola Salomoni e Silvia Mirri, datata 2018 [27]. Tale pubblicazione riguarda un prototipo precedente al lavoro presentato in questa tesi con sostanziali differenze e evidenti aspetti comuni, riportati in seguito in questo capitolo. Le *personas* alla base del lavoro sono qui riportate. Esse pongono l’accento sugli aspetti di wayfinding per persone con disabilità e sulla volontà di integrare aspetti turistico/culturali nel prodotto finale.

Giuseppe

Giuseppe è uno studente dell’Università di Bologna, iscritto al corso di laurea triennale in Filosofia. Ha scelto il curriculum di ”Concetti e Contesti”.

Utilizza una sedia a rotelle dall'età di sette anni. Si è trasferito a Bologna da Calderino, una piccola città vicina a Bologna, per motivi di studio. Lui è appassionato di musica e film e va spesso al cinema con gli amici. E' una matricola ed ha aggiunto al proprio piano di studi il corso di Filosofia della Scienza.

Giuseppe deve assistere alla sua prima lezione di Filosofia della Scienza in aula Tibilleti, che si trova al secondo piano. Apre l'applicazione e seleziona aula Tibilleti come punto di arrivo. L'applicazione rileva la sua posizione, usando il segnale ricevuto dai beacon vicini e crea un percorso in modo che Giuseppe possa raggiungere la sua destinazione.

Quando Giuseppe ha fatto il primo accesso all'applicazione, ha specificato tra le impostazioni di utilizzare una sedia a rotelle. Per questo motivo, il sistema di wayfinding non ha considerato scale e piccoli ascensori presenti nell'edificio, ma ha creato un percorso che utilizzasse l'ascensore per sedie a rotelle. Una volta che l'applicazione ha generato il percorso, Giuseppe può cominciare la navigazione. L'applicazione lo guida lungo il piano, utilizzando i beacon circostanti per determinare la sua posizione finché non ha raggiunto l'aula Tibilleti, precedentemente impostata come arrivo. Mentre si sta recando a lezione, Giuseppe passa vicino ad un mezzobusto. L'applicazione propone a Giuseppe di leggere una breve descrizione di quest'opera d'arte. Siccome Giuseppe non riconosce la persona, decide di leggere la descrizione e scopre che si tratta di Giosuè Carducci. Poi continua a seguire le istruzioni di navigazione.

Linda

Linda è una studentessa dell'Università di Bologna, iscritta al corso di laurea triennale in Filosofia. E' nata cieca ed ha sempre vissuto a Bologna. Lei è con Bob, il suo cane guida da 3 anni. E' appassionata di libri e passa molto del suo tempo libero leggendo pubblicazioni di tutti i tipi. Frequenta il secondo anno e ha aggiunto al suo curriculum il corso di Antropologia Filosofica.

Linda ha lezione di Filosofia della Conoscenza in aula 5 che si trova al secondo piano. Quando arriva a Palazzo Riario, lei apre l'applicazione e seleziona l'aula numero 5 come punto di arrivo. L'applicazione rileva la sua posizione, usando il segnale ricevuto dai beacon vicini e crea un percorso in modo che Linda possa raggiungere la sua destinazione.

Quando Linda ha fatto il primo accesso all'applicazione, ha specificato tra le impostazioni di essere cieca e che preferisce le scale all'ascensore per via di Bob. Per questa ragione, il sistema di wayfinding non ha considerato l'ascensore e ha creato un percorso che usa le scale. Una volta che l'applicazione ha generato il percorso, Linda può cominciare la navigazione. L'applicazione, grazie alla sintesi vocale, la guida lungo il tragitto, utilizzando i beacon circostanti per determinare la sua posizione finché lei non ha raggiunto l'aula numero 5, precedentemente impostata come arrivo.

2.3 Scelte progettuali

Le scelte progettuali di seguito illustrate sono state fatte per rispettare le richieste già presentate e, successivamente, per consentire una implementazione rapida di un prodotto di alta qualità in termini di esperienza utente. Il prodotto finale deve poter essere utilizzato in tutti gli scenari presentati.

2.3.1 Tecnologia di posizionamento

La tecnologia scelta è quella dei beacon. Tale adozione deriva dalla necessità di utilizzare componenti dalla facile installazione, che garantiscano una buona precisione e sfruttabili dagli utenti senza l'acquisto di strumenti specifici. Pro e contro di questa tecnologia sono presentati nel capitolo 1.

Approccio alla localizzazione

I diversi approcci proposti dalla comunità scientifica sono generalmente sintetizzabili in 4 categorie:

1. Proximity: permettere di localizzare il device se esso si trova in prossimità di un beacon, ovvero nel suo raggio di emissione del segnale. Se un device rileva il segnale di un beacon si assume che il device si trovi nella stessa posizione del beacon. Se vengono ricevuti più segnali si considera soltanto quello con l'intensità di segnale più alta. Questo approccio è di facile implementazione e non introduce complessità algoritmica.
2. Triangulation: come già visto nella sezione dedicata ai satelliti GPS, la triangolazione è un metodo per stimare la posizione del device grazie alle proprietà dei triangoli. La triangolazione si può suddividere in trilaterazione e angolazione. La prima si basa sulla distanza tra il device e svariati emettitori, stimata tramite Time Of Arrival, Signal Strength, ecc. La seconda esegue la computazione a partire dagli angoli di ricezione del segnale. In entrambi i casi è indispensabile la ricezione delle onde trasmesse da più emettitori. Ciò significa che i raggi di azione dei beacon devono obbligatoriamente sovrapporsi.
3. Fingerprinting: questa tecnica confronta le intensità dei segnali rilevati con un set di valori noti. Il processo si sviluppa in due fasi. Durante la prima array formati dai valori di potenza del segnale di ogni beacon vengono rilevati in diverse posizioni e salvati. Durante la seconda fase si effettua una operazione di pattern matching. Il segnale rilevato viene confrontato con i valori noti e si sceglie quello più simile. La precisione dipende principalmente dal numero di rilevazioni effettuate in fase di set-up. Con questa tecnica, spostare un emettitore significa dover ripetere l'intera fase off-line di acquisizione dei segnali. Ciò può essere evitato introducendo l'uso di modelli che, di contro, introducono complessità e possibilità di errore. La soluzione con fingerprint è quella adottata nell'ambito del progetto NavCog citato nel capitolo precedente [3, 4].
4. Pedestrian Dead Reckoning: è una tecnica di navigazione relativa. Si basa infatti sul calcolo della posizione del device a partire da una po-

sizione nota ed aggiungendo i successivi spostamenti. Gli spostamenti possono essere stimati rilevando valori di velocità, distanza. La natura ricorsiva dell'approccio genera inevitabilmente un accumularsi di errori generati durante le successive stime. L'errore può essere contenuto entro parametri stabiliti introducendo aggiornamenti basati sulla posizione assoluta.

L'approccio scelto è quello della proximity. Ciò porta alcuni vantaggi. L'evidente semplificazione del codice, la possibilità di aggiungere beacon senza particolari accorgimenti, la possibilità di utilizzare una potenza di emissione del segnale dei beacon inferiore. Infatti non è necessario che più segnali si sovrappongano per abilitare processi di triangolazione o di fingerprinting, è sufficiente mettere in campo un numero di beacon tale da evitare zone dove nessun messaggio si propaghi.

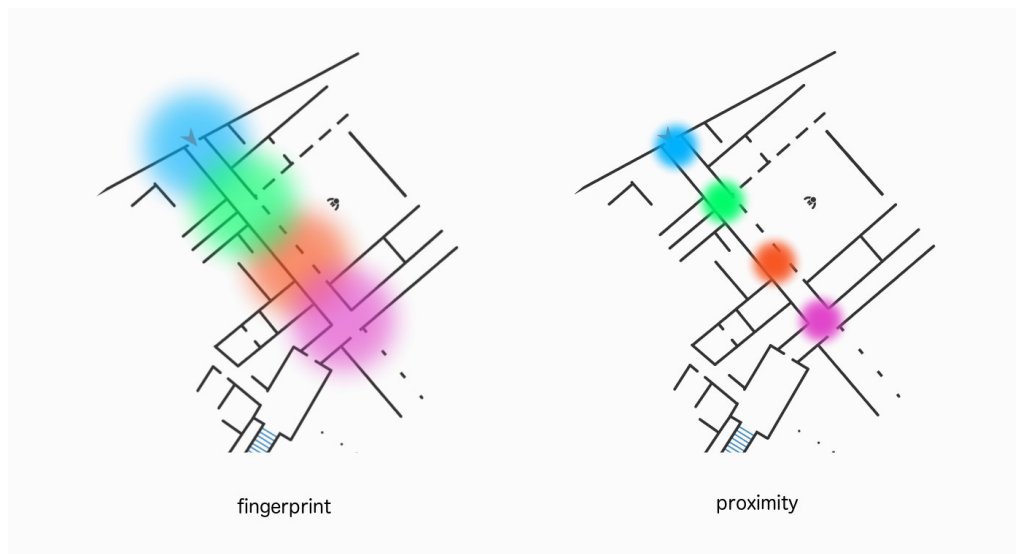


Figura 2.2: Approcci alla localizzazione con i beacon

I beacon acquistati sono i "Kontak Smart Beacon SB16-2" adatti allo scopo per le funzionalità offerte:

1. Supporto per i protocolli iBeacon ed EddyStone.

2. Fino a 4 anni di durata della batteria.
3. Il firmware dà la possibilità di scegliere potenza, intervallo di emissione e protocollo.
4. E' disponibile una app per il monitoraggio della batteria e per la modifica dei parametri.
5. La documentazione è ampia e completa.

Problemi: Segnale ed errori

Il segnale dei beacon è spesso incostante e può subire oscillazioni impreviste. Inoltre la propagazione del segnale viene influenzata da oggetti, muri, presenza di altri dispositivi. Anche l'altezza da terra alla quale vengono posizionati influisce sulla propagazione dei messaggi.

Soluzione

Il sistema di localizzazione basato su proximity mitiga queste oscillazioni e limita gli errori. Tuttavia in fase di implementazione dovranno essere messe in campo tecniche atte a nascondere tali problematiche all'utente.

2.3.2 Framework

L'ingegnere Jordan Walke ha sviluppato presso Facebook una libreria JavaScript chiamata React, utilizzata a partire dal 2011 da Facebook stesso e successivamente da Instagram. Tale libreria serve per la creazione di interfacce utente e viene mantenuta dall'azienda creatrice insieme ad una comunità di sviluppatori. Dal 2013 è diventata open-source. A partire da questa libreria è stato sviluppato il framework React Native che consente agli sviluppatori mobile di usare React insieme alle funzionalità native della piattaforma. I sistemi supportati sono iOS, Android e UWP. Si tratta di un framework che consente la realizzazione di applicazioni mobili utilizzando JavaScript e organizzando l'interfaccia grafica tramite components [29].

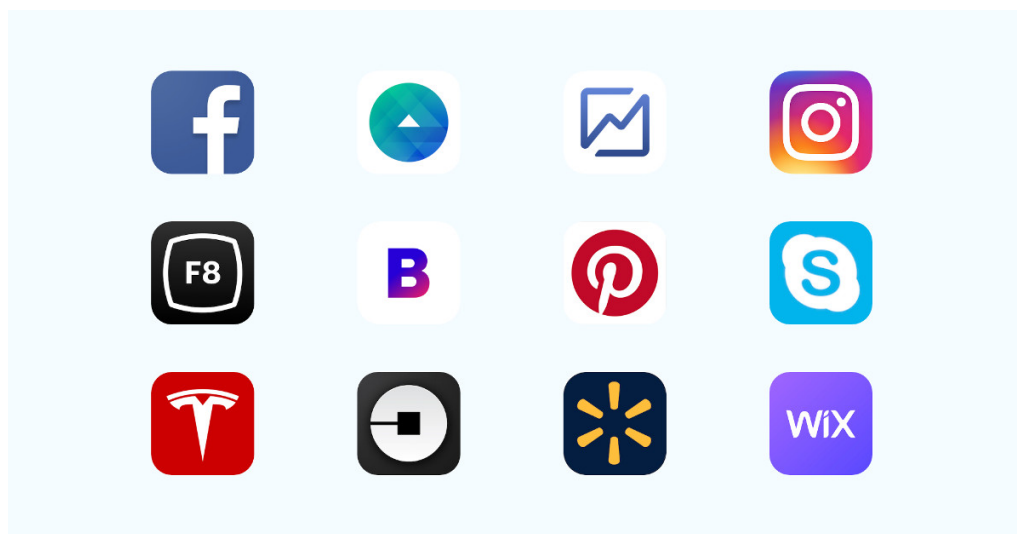


Figura 2.3: Applicazioni che utilizzano React Native

Non si tratta di un processo per la costruzione di applicazioni ibride che necessitano di una WebView per essere correttamente eseguite. La WebView infatti consentirebbe di unificare lo sviluppo per la pubblicazione sui diversi store, ma a discapito delle performance. React Native si basa, al contrario, sui blocchi costruttivi nativi delle due piattaforme e li mappa in component. Tramite il linguaggio JavaScript possiamo organizzare questi component e crearne di nuovi tramite la composizione di quelli esistenti. Ogni component possiede un suo stato e ha un ciclo di vita. Questa struttura semplifica, tra le altre cose, l'esecuzione di Unit test. React Native sfrutta un processo in background sul device che interpreta il codice JS e comunica con la piattaforma nativa tramite bridging [28]. Non usa perciò il linguaggio HTML: gli eventi del thread JavaScript vengono utilizzati per manipolare le view native. Per lo sviluppo della applicazione per il wayfinding indoor è stato adottato questo framework per alcune delle sue caratteristiche:

1. Il team possiede una solida esperienza nella programmazione JavaScript sia in campo web che mobile.
2. Tempi di implementazione: il codice può essere scritto una sola volta

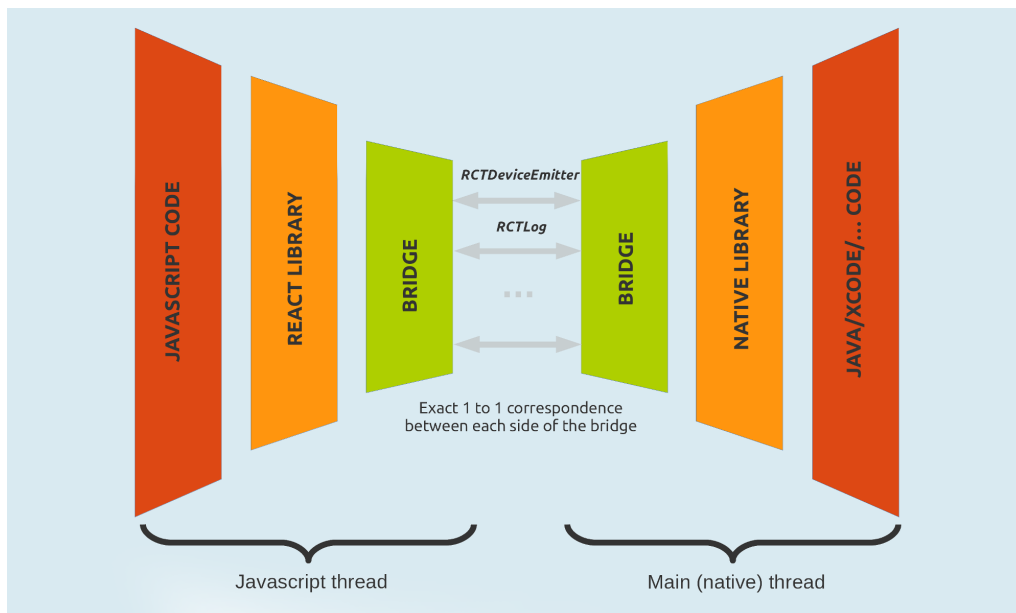


Figura 2.4: Bridging in React Native

con eventuali piccoli interventi in linguaggio nativo iOS/Android.

3. React Native può contare su una vasta community che sviluppa e testa una grande quantità di librerie.
4. Codice nativo può essere utilizzato nel caso di esigenze particolari nella gestione delle risorse.
5. La velocità di esecuzione è superiore a quella di applicazioni ibride ed è paragonabile con applicazioni native.
6. Supporta i meccanismi di accessibilità forniti dai vari sistemi operativi e consente una corretta etichettatura di ogni componente per permettere agli screen reader di interagire con essi.

2.3.3 Librerie

In questo paragrafo sono elencate le librerie scelte per l'implementazione della applicazione. La scelta è stata in alcuni casi obbligata per la mancanza di alternative, in altri studiata e presa in seguito all'analisi delle funzionalità proposte, della qualità del codice, ecc... Caratteristica comune a tutte le librerie è la compatibilità con iOS e Android.

1. react-native-beacons-manager [30]: Il codice viene proposto come alternativa a due librerie precedenti ognuna delle quali specializzata su di un solo sistema operativo. E' compatibile con il protocollo iBeacon e con EddyStone. Questa libreria si occupa di monitorare una regione di spazio e di restituire un array di tutti i beacon rilevati nella regione, ognuno dei quali con la seguente struttura:

- uuid: per distinguere i beacon della propria installazione da altri.
- major: spesso usato per identificare il piano all'interno dell'edificio.
- minor: per identificare il singolo beacon all'interno di un piano.
- rssi: intensità del segnale da -100 a 0.
- proximity: valore di prossimità che può assumere valore "far", "near" o "immediate".
- accuracy: precisione.

Verrà usata per la localizzazione.

2. react-native-simple-compass [31]: scelto per la buona compatibilità con i due sistemi operativi nonostante non si tratti di un prodotto recente. Consente di monitorare la bussola del device chiamando una callback ad ogni variazione del valore della stessa. Può essere specificato dopo quanti gradi di rotazione deve essere chiamata la callback, per evitare chiamate troppo frequenti.

Verrà utilizzata per sapere in che direzione sta camminando l'utente.

3. node-dijkstra [32]: codice per la risoluzione del problema del cammino minimo di Dijkstra. Implementazione in JavaScript con buone performance. E' possibile specificare eventuali nodi da evitare e se tener conto dei pesi. Restituisce un array.

Verrà utilizzata per calcolare il percorso verso la destinazione richiesta.

4. react-native-tts [33]: si appoggia alle librerie di iOS e Android per fornire un'interfaccia uniforme per il text-to-speech, con possibilità di regolare velocità di lettura, lingua, ecc...

Verrà utilizzata per fornire indicazioni in fase di navigazione.

5. react-native-navigation [34]: questa libreria è in costante aggiornamento e rappresenta la vera essenza di React Native. Fornisce una modalità cross-platform in linguaggio JavaScript per implementare tutte le funzioni di navigazione. Tutto è mappato in codice nativo e a fronte di un'abbondante lettura di documentazione garantisce ottime performance. I principali componenti sono:

- Gli stack di navigazione multipli sui quali si possono eseguire operazioni di push e pop di schermate.
- La top bar con gestione di menu, back button, ecc...
- Menu laterale
- Bottom tabs per spostarsi da uno stack di navigazione all'altro.

Verrà utilizzata per le funzioni di navigazione e organizzazione delle schermate.

2.3.4 Sistema di navigazione

Il sistema di navigazione deve fornire le indicazioni dopo averle estratte da un documento in formato open e standard. JSON consente di specificare tutti i dati necessari con un file human-readable. Il percorso deve essere calcolato

con l'algoritmo dei cammini minimi di Dijkstra presentato nel capitolo 1 di questa tesi e presentato all'utente in più formati:

- **Mappa:** per comunicare informazioni riguardanti l'intero ambiente. Per rispettare la tempistica si preferisce utilizzare una piantina in formato compresso .jpg ricavabile dalle planimetrie dell'edificio. Tale scelta è dettata da un sostanziale risparmio di tempo nell'implementazione. L'integrazione di un sistema di mappe più complesso potrà essere oggetto di sviluppi futuri.
- **Testo:** per consentire a chi lo preferisce, di leggere le indicazioni. Tutte le indicazioni devono essere disponibili per la lettura in qualunque momento della navigazione. L'istruzione che dovrà essere eseguita a breve dall'utente deve occupare una posizione di rilievo.
- **Simboli:** universalmente compresi, per indicare le svolte. Devono essere presentati con colori ad alto contrasto ed in una posizione di rilievo.
- **Audio:** per fornire istruzioni nell'esatto momento in cui devono essere eseguite. Deve trattarsi di informazioni vocali e non di semplici segnali sonori.

2.3.5 Accessibilità

All'utente con disabilità motorie deve essere consentito di utilizzare l'applicazione con l'ausilio di uno screen reader. Deve inoltre risultare possibile la fruizione di tutti i contenuti, anche quelli a stampo turistico/culturale. All'utente con disabilità motorie deve essere presentato un percorso, se esistente, caratterizzato dall'assenza di barriere architettoniche, calcolato con lo stesso algoritmo dei percorsi standard. Tale algoritmo deve dunque essere utilizzato con modalità atte a raggiungere il risultato atteso.

2.4 Specifiche di progetto

Le scelte progettuali di maggior rilievo sono state presentate. Esse, integrate con ulteriori scelte finora tralasciate per l'inferiore interesse (per quanto riguarda tecnologie, accessibilità, innovazione, ricerca, ...) danno origine alle specifiche di progetto che seguono:

A-GPS Per l'identificazione dell'edificio in cui si trova l'utente all'avvio della applicazione. Deve essere possibile scegliere manualmente l'edificio nel quale ci si trova in assenza di segnale GPS.

Beacon Per la precisa localizzazione dell'utente con sistema basato su proximity. Ogni Point-Of-Interest deve coincidere con un beacon.

Bussola Per l'orientamento del device.

Mappa, Testo, Simboli, Voce Per l'erogazione delle indicazioni e delle informazioni culturali.

Server Per l'archiviazione dei dati. Nessun vincolo sull'implementazione.

Interfaccia web Per l'esecuzione di operazioni CRUD da parte di personale abilitato.

Funzione di navigazione Guida l'utente verso la sua destinazione. Le destinazioni sono selezionabili da un elenco di Point Of Interest dell'edificio filtrabile tramite un campo di testo. Vengono fornite indicazioni ad ogni svolta ed all'arrivo. Durante la navigazione è possibile richiedere informazioni riguardanti opere d'arte incontrate sul percorso.

Funzione "intorno a te" Mostra all'utente i POI più vicini a lui e presenta le informazioni turistico/culturali di quello selezionato.

Funzione "tour" Guida l'utente verso una serie di POI storico/culturali all'interno dello stesso palazzo, legati da un tema comune.

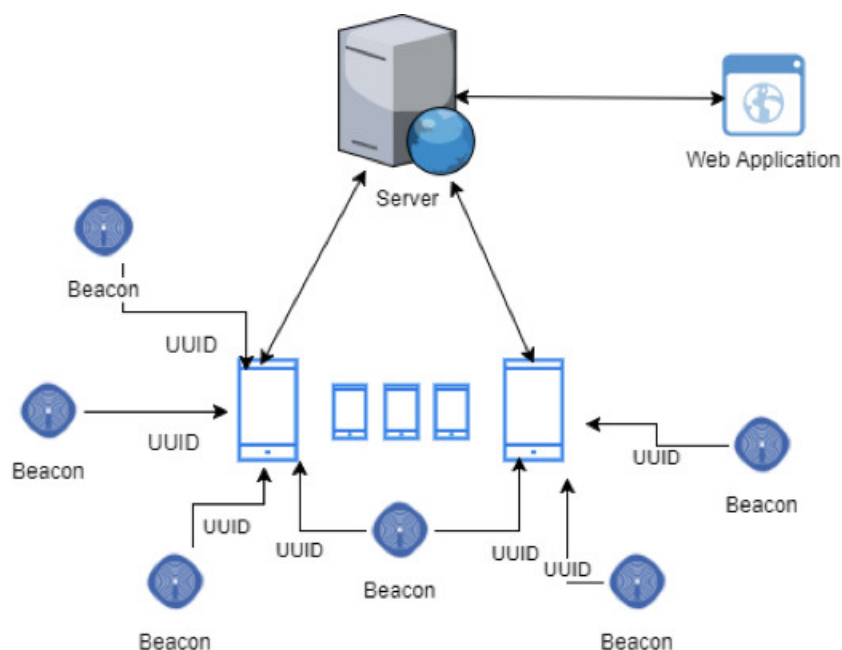


Figura 2.5: Architettura del sistema

2.5 Prototipo esistente

Come già accennato in questo capitolo, un prototipo di applicazione per la navigazione indoor con l'utilizzo di beacon era stato progettato in precedenza nell'ambito di progetti di tesi presso l'Università di Bologna. Tale prototipo ha consentito di studiare approfonditamente il tema della navigazione turn-by-turn e delle tecnologie di posizionamento. Tuttavia il codice non è mai stato completato e l'implementazione di un nuovo prodotto è stata necessaria per superare alcuni problemi tecnico/implementativi e per l'integrazione di nuove funzionalità. Il codice preesistente non è stato riutilizzato e lo studio di requisiti, tecnologie, strumenti è stato condotto nuovamente, supportato dalle nozioni apprese dal precedente progetto. E' dunque doveroso sottolineare alcune analogie con tale prototipo ed evidenziare le fondamentali differenze.

2.5.1 Analogie

1. Sistema pensato per la navigazione indoor accessibile.
2. Utilizzo dei beacon.
3. Stessi scenari di utilizzo.

2.5.2 Differenze

Oltre alle differenze nell'utilizzo della sensoristica, nell'interfaccia, ecc... si possono riassumere tutte le disuguaglianze in due punti centrali per l'intero progetto:

1. L'approccio alla localizzazione: l'utilizzo dei beacon tramite fingerprint è risultato, nell'ambito del precedente progetto, complesso e poco affidabile. Sebbene tale soluzione sia stata già adottata da vari team citati in precedenza, essa è influenzata numerosi da fattori esterni come la presenza di altri individui. Inoltre per consentire l'aggiunta di nuove mappe e nuovi beacon necessità di un delicato lavoro di acquisizione dei segnali lungo tutti i possibili percorsi. Per questi motivi è stato scelto un approccio basato su proximity che garantisce una implementazione semplificata del codice di rilevamento del segnale e, soprattutto, permette l'introduzione di nuovi beacon da parte del personale universitario.
2. Le funzionalità: rispetto alla sola navigazione accessibile sono state introdotte la funzione "intorno a te" per conoscere le opere d'arte vicine e la funzione "tour" per visitare il palazzo conoscendo il patrimonio artistico che contiene. Tali funzionalità coprono pienamente i requisiti legati agli aspetti storico/culturali considerati marginali nel precedente studio.

Capitolo 3

Implementazione

3.1 Team

Per l'implementazione del prodotto finale è stato formato un gruppo di due ingegneri informatici, studenti presso l'Università di Bologna:

- Vincenzo Rubano: studente magistrale di ingegneria informatica presso la sede di Bologna. Vincenzo è un esperto programmatore, laureato triennale nel 2016 con la tesi "L'(in)accessibilità degli articoli scientifici sul Web e l'uso di RASH e EPUB." [?]. Il suo contributo al progetto deriva dall'esperienza maturata nel corso degli anni di studio presso l'università e precedenti, durante i quali ha anche partecipato attivamente a progettazione e sviluppo di progetti di alto livello come ad esempio la piattaforma Drupal. Conoscitore del mondo dell'accessibilità in quanto programmatore con disabilità visive e utilizzatore di ausili specifici.
- Giacomo Mambelli: studente magistrale di ingegneria e scienze informatiche presso la sede di Cesena ed autore di questa tesi. Laureato triennale nel 2016 con la tesi "Progetto e realizzazione di una applicazione mobile a supporto di iniziative ed eventi culturali." [?]. Il suo contributo al progetto deriva dall'esperienza maturata nel corso degli anni di studio presso l'università nell'ambito della programmazione

mobile sia nativa che ibrida, della conoscenza dei meccanismi di programmazione event driven in linguaggio JavaScript e dall'approfondita conoscenza acquisita da autodidatta del framework React Native.

Coordinatrice del team è la professoressa Silvia Mirri, docente presso l'Università di Bologna, supportata dalla dottoressa Catia Prandi.

3.2 Ambiente di sviluppo

L'applicazione è stata creata da terminale e sviluppata all'interno dell'editor Atom. Il comando di seguito riportato indica la metodologia di creazione di un progetto non standard, cioè senza l'utilizzo dello strumento Expo, introdotto per semplificare lo sviluppo per programmatori meno esperti. Ciò che Expo propone è una inizializzazione veloce, importazione di librerie semplificata, condivisione della app tramite QRcode, ecc. Tuttavia questo strumento implica dimensioni maggiori (25MB) per la app vuota e, soprattutto, l'impossibilità di integrare moduli scritti in codice nativo o di utilizzare elementi ReactJS.

```
react-native init ProjectName
```

La scelta di rinunciare ad Expo comporta la necessità di utilizzare Android-Studio e xCode per eventuali modifiche al codice nativo ed anche per l'esecuzione del progetto. Essendo la disponibilità della app anche sullo store Apple un requisito, è stato necessario sviluppare l'intero progetto su macchine Mac per poter usufruire del compilatore proprietario. La gestione delle dipendenze senza Expo avviene da terminale e può implicare modifiche a più file di progetto sia con AndroidStudio che con xCode.

3.2.1 Librerie

La gestione delle librerie è stata effettuata da terminale con lo strumento di gestione delle dipendenze Yarn.

3.2.2 Permessi

La versione 6 di Android ha introdotto una semplificazione dei permessi individuando un ridotto numero di aree alle quali le app possono richiedere di accedere: servizi di localizzazione, fotocamera, microfono, contatti, telefono, SMS, calendario e sensori. Solo durante il primo utilizzo di una di queste funzionalità verrà chiesto il permesso all'utente e non più in fase di installazione. Il file `AndroidManifest.xml` è stato modificato come segue:

```
1 <uses-permission
2     android:name="android.permission.ACCESS_FINE_LOCATION" />
3 <uses-permission
4     android:name="android.permission.ACCESS_NETWORK_STATE" />
5 <uses-permission
6     android:name="android.permission.INTERNET" />
```

Listing 3.1: Permessi Android

La gestione dei permessi per iOS è semplificata dall'utilizzo dell'ambiente di sviluppo proprietario.

3.3 Interfaccia

Per implementare l'interfaccia si è scelto di utilizzare i colori Unibo (rosso, grigio, bianco). Lo stile della landing page segue la linea estetica proposta dalla prima applicazione ufficiale dell'Università di Bologna, ovvero `myAlmaOrienta`. Per tutte le altre schermate sono stati definiti alcuni aspetti comuni come lo sfondo bianco, la top bar rossa con scritte bianche, il testo nero, bottoni ampi. E' stata implementata una schermata dedicata alle impostazioni dalla quale è possibile attivare/disattivare le indicazioni vocali durante la navigazione, richiedere solo percorsi accessibili, cambiare i colori dell'intera app rendendo gli sfondi scuri ed adatti per una navigazione in condizioni di scarsa luminosità.

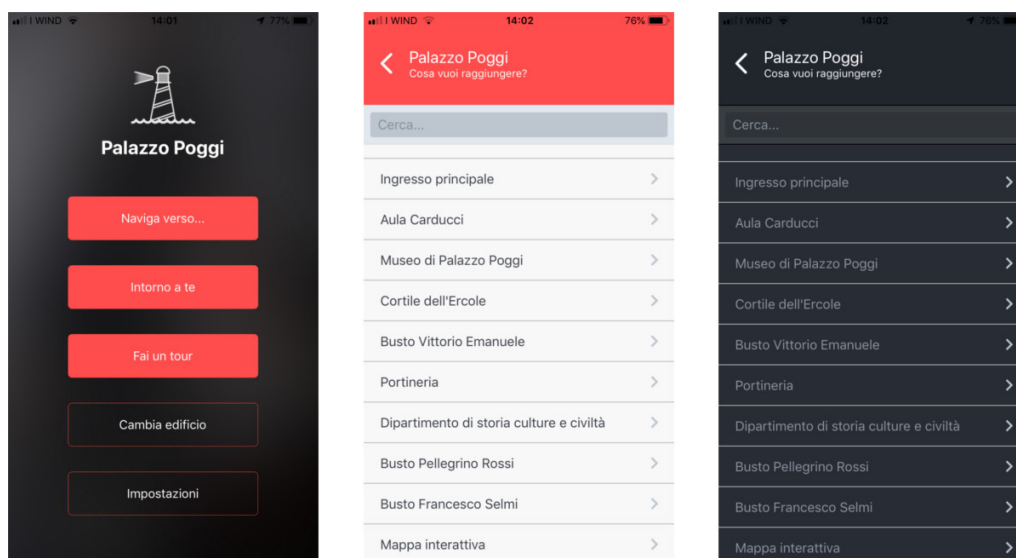


Figura 3.1: Interfaccia grafica

Accessibilità

In React Native quando un elemento è accessibile, esso raggruppa tutti i suoi figli in un unico componente selezionabile. L'attributo `accessible=true` rende un elemento accessibile e, ad esempio su Android, viene tradotto in `focusable=true`. E' considerata buona prassi ed è indispensabile per poter utilizzare screen reader, indicare l'attributo `accessibilityLabel` che specifica una stringa usata per capire che elemento è selezionato. Un altro attributo utilizzato nella app è `accessibilityRole` che comunica il tipo di elemento all'utente che sfrutta tecnologie assistive. Di seguito un esempio di immagine e bottone accessibili.

```

1 <Image
2   accessible={true}
3   accessibilityLabel={ t('App logo') }
4   accessibilityRole='image'
5   // ...
6 />
7 <TouchableOpacity
8   accessible={true}

```

```
9     accessibilityLabel={ t('Navigate to') }
10     accessibilityRole='button'
11     // ...
12 </TouchableOpacity>
```

Listing 3.2: Accessibilità in React Native

3.4 Formato dati

Nel rispetto delle specifiche i dati necessari al funzionamento della applicazione come l'elenco di edifici, beacon, Point-Of-Interest e immagini sono stati pensati per essere trasferiti per mezzo di un unico file in formato .JSON. Per ogni edificio sono specificati nome, latitudine, longitudine, UUID, e tre oggetti contenenti beacons, nodi e archi. I beacon contengono major, minor ed un id semplificato di 3 cifre per identificare il beacon con un solo numero. I nodi rappresentano i Point-Of-Interest e posseggono un id, l'id del beacon di riferimento, il nome, la tipologia (stanza, quadro, ecc...), l'orientamento in gradi rispetto al nord, informazioni ulteriori da erogare all'arrivo presso il nodo stesso. Gli archi rappresentano i tratti percorribili per spostarsi da un beacon a quello successivo e presentano un id, un beacon di partenza, uno di arrivo, l'orientamento in gradi rispetto al nord, un flag per l'accessibilità e l'immagine (mappa).

```
1 {
2   "id": "001",
3   "beacon": "001",
4   "type": "door",
5   "name": "Ingresso principale",
6   "degrees": "135",
7   "info": "attenzione alla doppia porta"
8 }
```

Listing 3.3: Esempio di nodo in formato JSON

3.4.1 Immagini

Le immagini, raffiguranti porzioni di mappa (tra due nodi), sono state prodotte e partire dalle piantine Unibo opportunamente corrette e semplificate. I file sono stati salvati in formato .png per poi essere convertiti in Base64 e inseriti come testo all'interno del file .JSON. Base64 è un sistema di codifica che consente la traduzione di dati binari in stringhe di testo ASCII che trova uno dei suoi primi utilizzi nell'invio di allegati nelle email.

3.5 Posizionamento

3.5.1 GPS

Per stabilire l'edificio in cui l'utente si trova e conseguentemente considerare solo i beacon installati all'interno di esso, si è scelto di utilizzare la localizzazione GPS del device che, pur non garantendo un alta precisione, è sufficiente per raggiungere tale scopo. La scelta manuale dell'edificio è comunque possibile in caso di selezione automatica errata o di assenza totale di segnale. Il posizionamento con GPS è la prima operazione che viene effettuata dopo il caricamento dei dati. Calcola la distanza tra l'utente e tutti gli edifici presenti nel file. Quello con la distanza minore è l'edificio dove l'utente si trova.

```
1 componentDidMount() {  
2     loadJson((json) => {  
3         this.setState({buildings: json.buildings});  
4         if(this.props.building===null) this.findBuilding(json);  
5     });  
6 });
```

Listing 3.4: Caricamento dati e identificazione edificio

3.5.2 Beacon

Il posizionamento ad alta precisione all'interno di un edificio è stato implementato, come da specifiche, con l'utilizzo di beacon e con un approccio basato su proximity. La classe preposta al monitoraggio dei segnali ricevuti ed al calcolo della posizione si chiama BeaconMonitor e mette a disposizione due funzioni, una per avviare il monitoraggio ed una per interromperlo, ad esempio, al termine della navigazione. La funzione di start accetta in ingresso una callback da chiamare ogni volta che il beacon più vicino cambia. Un beacon viene eletto "closest beacon" quando due letture consecutive ne confermano il valore di prossimità più alto tra tutti quelli rilevati.

```
1 bindEvents() {
2     this.beaconsDidRangeEvent = DeviceEventEmitter.
      addListener('beaconsDidRange', (data) => {
3         if(data.beacons[0] != null) this.selectClosestBeacon(
          data.beacons);
4     });
5     // ...
6 }
7 selectClosestBeacon(beacons) {
8     const sortedBeacons = [... beacons].sort(this.
      compareBeaconProximity)
9     var closestBeacon = sortedBeacons[0];
10    if (closestBeacon.major == this.closestBeaconCandidate.
      major && closestBeacon.minor == this.
      closestBeaconCandidate.minor) {
11        this.closestBeaconCandidate.count++;
12        if (this.closestBeaconCandidate.count ==
          BeaconMonitor.closestBeaconTreshold) {
13            this.reportClosestBeacon();
14        }
15    }
16    else { // a new candidate
17        this.closestBeaconCandidate.major = closestBeacon.
          major;
```

```
18     this.closestBeaconCandidate.minor = closestBeacon.  
        minor;  
19     this.closestBeaconCandidate.proximity = closestBeacon  
        .proximity;  
20     this.closestBeaconCandidate.count = 0;  
21 }  
22 }
```

Listing 3.5: Gestione dell'evento beaconsDidRange

3.6 Calcolo percorso

La classe che si occupa del calcolo del percorso migliore è denominata `BeaconMap`. Il costruttore prende in ingresso la lista dei beacon, quella degli archi e un valore booleano che, posto a `true`, indica la necessità di ottenere solo percorsi accessibili. I percorsi accessibili vengono calcolati con le stesse modalità. La differenza è a monte del calcolo con l'algoritmo dei percorsi minimi di Dijkstra. Gli archi vengono infatti filtrati direttamente nel costruttore, prima della chiamata `getPath()`.

```
1 const beaconEdges = (this.isAccessible) ? edges.filter((edge)  
    => { return edge.accessible })  
2 ): edges;
```

Listing 3.6: Rimozione degli archi non accessibili

La funzione `getPath()` avvia il calcolo del percorso dopo aver ricevuto in ingresso partenza e arrivo (beacon in formato JSON oppure id dei beacon) e restituisce un oggetto contenente sia l'array dei beacon che quello degli archi usato poi in fase di navigazione. Essendo gli archi presenti nel file JSON direzionati, è necessaria una operazione di inversione degli archi per poter considerare percorsi in ogni direzione, essa è implementata da `invertEdge()`.

3.7 Funzione "naviga verso..."

Per poter avviare la navigazione l'utente deve aprire l'applicazione e nella landing page selezionare "Naviga verso..." successivamente vengono presentate tutte le possibili destinazioni. Un campo di ricerca permette di trovare rapidamente quella desiderata filtrando la lista in base all'input dell'utente.

```
1 filterList(text){
2     if(this.state.building.nodes && this.state.building.
3         nodes.length > 0){
4         var newNodesList = [];
5         for(var i=0; i < this.state.building.nodes.length
6             ; i++){
7             if(this.state.building.nodes[i].name.
8                 toLowerCase().indexOf(text.toLowerCase())
9                 >=0){
10                newNodesList.push(this.state.building.
11                    nodes[i])
12            }
13        }
14        this.setState({nodesFilteredList: newNodesList});
15    }
16 }
```

Il component, chiamato NavigationMenu, passa alla schermata successiva la destinazione scelta, per avviare la navigazione turn-by-turn. Il component pensato per la navigazione vera e propria si chiama NavigationScreen ed è la porzione di codice più consistente del progetto. Di seguito vengono presentate le principali caratteristiche di questa classe.

3.7.1 Calcolo Indicazioni

Il calcolo delle indicazioni, sia visive che audio, da fornire all'utente avviene ogni volta che il beacon con maggiore prossimità cambia. Esso si basa anche sull'orientamento del device, rilevato tramite la bussola interna, e sulla direzione in gradi dell'arco che deve essere percorso dall'utente. La funzio-

ne di seguito riportata, presa in ingresso la direzione dell'arco, restituisce la direzione in cui l'utente deve muoversi per percorrerlo. La scelta iniziale prevedeva 8 possibili valori di ritorno, poi ridotti a 4 (+1 di errore). Questa scelta è stata fatta per evitare che, nel caso in cui l'utente dovesse tenere il device in posizione non completamente ortogonale al suo corpo, vengano fornite indicazioni errate. La riduzione dei valori di ritorno e il conseguente allargarsi dei range di valori supportano quindi una maggiore tolleranza.

```
1  getDirection(arcDegrees){
2      var a = arcDegrees
3      var b = this.state.angle
4      var d = Math.abs(a - b) % 360;
5      var r = d > 180 ? 360 - d : d;
6      var sign = (a - b >= 0 && a - b <= 180) || (a - b <=-180
7          && a - b >= -360) ? 1 : - 1;
8      r *= sign;
9      if(r > -45 && r < 45) return "straight"
10     if(r > -135 && r < -45) return "left"
11     if(r > 45 && r < 135) return "right"
12     if(r > 135 || r < -135) return "back"
13     return "error"
14 }
```

Listing 3.7: Calcolo della direzione

Le indicazioni vengono formulate in base ad alcune condizioni che si possono verificare durante la navigazione, di seguito raggruppate in 3 categorie e vengono erogate sia in forma visiva (testo e mappa) che in forma di audio.

Prima indicazione

La prima indicazione viene fornita al momento del rilevamento del primo beacon ed essa si basa sull'orientamento rilevato dalla bussola con la funzione `getDirection()` (vista in precedenza). E' necessario utilizzare la bussola in quanto unico strumento per individuare l'angolo del device rispetto al nord.

```
1  getStartDirection(){
2      if(this.state.remainingArcs){
```

```
3     var arc = this.state.remainingArcs[0]
4     var direction = this.getDirection(arc.degrees)
5     return direction
6   }
7 }
```

Listing 3.8: Prima indicazione

Indicazioni durante il percorso

Durante la navigazione si può calcolare l'indicazione da erogare in modo più preciso considerando il beacon rilevato come vertice dell'angolo formato dall'arco appena percorso e da quella da percorrere. In questo modo è possibile conoscere la direzione che l'utente deve tenere con una semplice operazione tra angoli. E' possibile però che l'utente sbagli percorso e che il suo device rilevi un beacon non previsto. Ciò implica il ricalcolo del percorso e il conseguente utilizzo della bussola per fornire l'indicazione, rientrando così nel caso già discusso.

Ultima indicazione

Quando il device rileva il beacon corrispondente al Point-Of-Interest di destinazione diventa privo di significato il calcolo della direzione da seguire in quanto inesistente. Al contrario diviene utile l'indicazione riguardante l'orientamento del POI rispetto all'utente. Essa viene calcolata nuovamente con la funzione `getDirection()` e viene pronunciata dalla voce di sistema con la formula: "Sei arrivato. Destinazione alla tua destra.". Successivamente, sempre con l'utilizzo del text-to-speech vengono lette eventuali informazioni aggiuntive riportate nel JSON. Ad esempio: "Attenzione alla doppia porta.". Simultaneamente viene visualizzata la schermata di fine navigazione.

```
1 if(id == this.state.destinationItem.beacon){
2   this.state.beaconMonitor.stop()
3   var deg = this.getDirection(this.state.destinationItem.
   degrees);
```

```
4     var voice = this.t('you arrived');
5     if(deg == "right") voice += this.t('your destination is
6         on your right');
7     else if(deg == "left") voice += this.t('your destination
8         is on your left');
9     else if(deg == "straight") voice += this.t('your
10        destination is in front of you');
11    else if(deg == "back" ) voice += this.t('your destination
12        is behind you');
13    this.voice(voice + ', ' + this.state.destinationItem.info
14        );
15    this.props.navigator.push({screen: 'NavigationComplete'})
16 }
```

Listing 3.9: Ultima indicazione

3.7.2 Top bar

La top bar della schermata di navigazione non sfrutta lo stesso component utilizzato per il resto della applicazione. E' stata implementata infatti una barra caratterizzata dagli stessi colori e dimensioni, ma in grado di fornire informazioni per la navigazione. Essa è suddivisa in due sezioni. La prima, sulla sinistra, contiene indicazioni sotto forma di simboli, cioè frecce che indicano la direzione della svolta successiva. La seconda, di maggiori dimensioni, riporta in formato testuale la stessa informazione e, in piccolo, la successiva. Il contenuto della top bar viene aggiornato ad ogni beacon rilevato e si basa sulle indicazioni calcolate con le modalità precedentemente descritte.

3.7.3 Mappa

La mappa occupa la parte centrale della schermata di navigazione e corrisponde alla presentazione dell'immagine .JPG (Base64) dell'arco sul quale l'utente si trova. La localizzazione tramite proximity non è utilizzabile per visualizzare lo spostamento dell'individuo in tempo reale. La mappa viene aggiornata perciò ad ogni nodo (beacon). Ciò significa che la localizzazione è

più accurata in prossimità di svolte e di Point-Of-Interest. L'immagine della mappa è di forma circolare e ruota mantenendo la rappresentazione orientata verso il nord. Per fare ciò è stata utilizzata la bussola che aggiorna la posizione della mappa chiamando una callback ogni 5 gradi di rotazione del device.

3.7.4 Bottone "vicino"

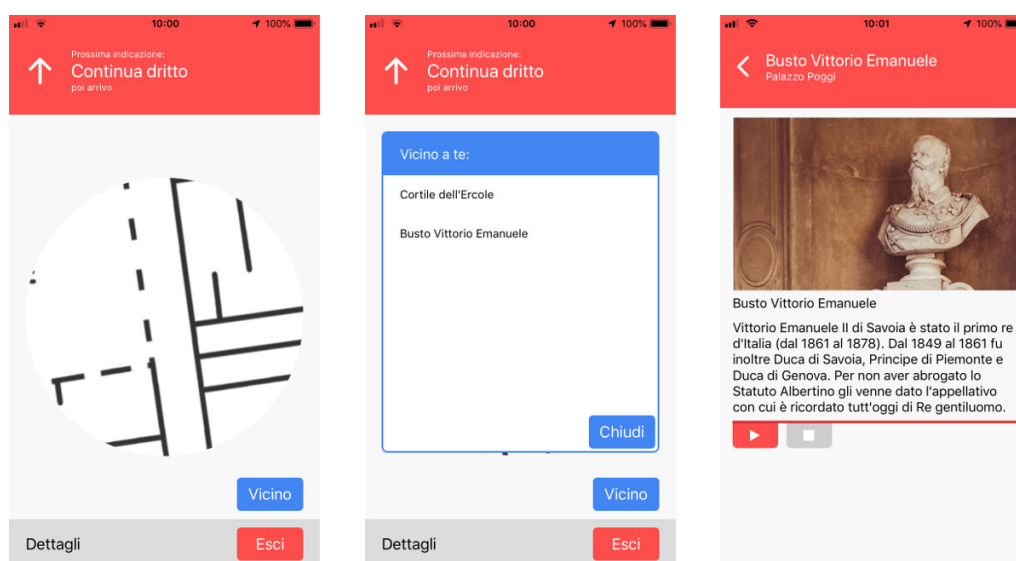


Figura 3.2: Presentazione POI turistico/culturale

Quando il device rileva, durante la navigazione, un beacon al quale sono associati anche Point-Of-Interest di tipo storico/culturale viene presentato a lato della mappa un bottone di colore azzurro che, se premuto, mostra l'elenco dei suddetti POI. Ciò non interrompe la navigazione, ma fa apparire un pop-up nella zona centrale dello schermo che può essere nascosto con l'utilizzo del bottone "chiudi". La pressione su uno degli elementi elencati nel pop-up attiva la transizione verso una schermata dedicata al POI selezionato. Tale schermata presenta la foto ed una breve descrizione dell'oggetto. La funzione di text-to-speech, utilizzata per fornire indicazioni in fase di navi-

gazione, è qui sfruttata per consentire all'utente di ascoltare il testo come se si trattasse di una audio-guida museale. La pressione del tasto back (fisico o non) riporta in primo piano la schermata di navigazione consentendo la prosecuzione dell'itinerario.

3.7.5 Bottom bar

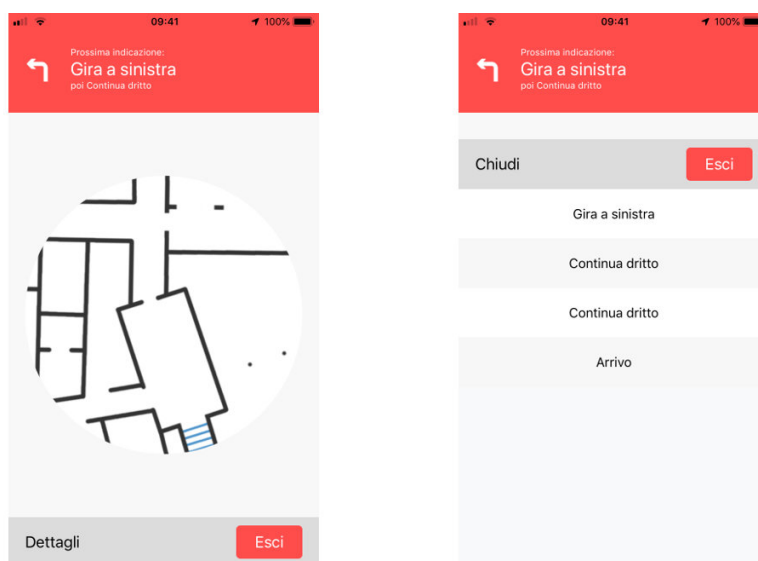


Figura 3.3: Funzionamento della bottom bar

Nella parte bassa dello schermo è stata inserito un component con una duplice funzionalità. Al suo interno si trova il bottone "Esci" che termina la navigazione ed il resto del component, anch'esso cliccabile, mostra i dettagli della navigazione. La visualizzazione dei dettagli consiste in una sezione scorrevole che appare dal basso verso l'alto e che contiene tutte le indicazioni rimanenti per il raggiungimento della destinazione. L'elenco completo viene calcolato sfruttando l'orientamento in gradi di tutti gli archi che separano l'utente dall'arrivo con il codice proposto di seguito. La lista viene aggiornata ad ogni beacon rilevato.

```
1 getRemainingDirections(){
```



```
2   var array = []
3   if(this.state.remainingArcs){
4       for(var i=1; i<this.state.remainingArcs.length; i++){
5           if(this.state.remainingArcs[i-1].degrees == this.
6               state.remainingArcs[i].degrees) array.push(
7               this.t('keep going straight on'))
8           else if((this.state.remainingArcs[i-1].degrees
9               *1+90)%360 == this.state.remainingArcs[i].
10              degrees) array.push(this.t('turn right'))
11          else if((this.state.remainingArcs[i-1].degrees
12              *1+270)%360 == this.state.remainingArcs[i].
13              degrees) array.push(this.t('turn left'))
14          else if((this.state.remainingArcs[i-1].degrees
15              *1+180)%360 == this.state.remainingArcs[i].
16              degrees) array.push(this.t('turn back'))
17      }
18      array.push(this.t('arrived'))
19  }
20  return array
21 }
```

Listing 3.10: Calcolo percorso completo

3.8 Funzione "intorno a te"

La funzione "intorno a te" è selezionabile dalla landing page e ha lo scopo di presentare i POI che si trovano nelle immediate vicinanze del device. Un component specifico denominato ExploreScreen si occupa di tale compito. Non appena un beacon viene rilevato, il component mostra tutti i POI associati a tale emettitore. Ogni voce della lista è caratterizzata dal nome e da una icona che suggerisce la tipologia dell'elemento, cioè storico culturale (icona di una macchina fotografica) o non (marker stile maps).

3.8.1 Lettura contenuto

Ogni voce dell'elenco è selezionabile e, alla pressione, viene caricata la scheda del POI scelto all'interno del component `ExploreSingleItem`. Si tratta della stessa schermata utilizzata per la presentazioni delle informazioni turistico/culturali durante la navigazione alla pressione del bottone "Vicino". Anche in questo caso viene visualizzata l'immagine, se disponibile e il testo estratto dal file `.JSON`. Sono disponibili, sotto al testo, due bottoni, uno per avviare la lettura tramite `tts` del contenuto e l'altro per interromperla. La lettura può avvenire, per chi ne fa uso, anche tramite screen reader essendo tale elemento accessibile. La funzione di lettura automatica del testo è stata infatti implementata per consentire a tutti, anche senza screen reader attivo, di ascoltare le nozioni come se si trattasse di una audio-guida museale.

3.9 Funzione "tour"

La funzione "tour" consiste nell'erogazione di indicazioni per consentire all'utente di spostarsi all'interno dell'edificio, lungo un percorso prestabilito, con la finalità di fargli incontrare una serie di POI legati tra loro da un tema comune, sia esso artistico, storico o di altra natura. I tour sono specificati sul file `.JSON` e corrispondono ad un array di ID riferiti ai nodi che si vogliono far raggiungere all'utente. Ad ogni POI del tour l'utente riceverà sullo schermo la scheda culturale associata al nodo stesso con la possibilità, terminata la lettura, di proseguire con la visita. Per realizzare questa funzione è stato utilizzato il component di navigazione, opportunamente modificato per consentire la navigazione verso una serie di destinazioni anziché verso una singola. E' possibile selezionare la modalità tour dalla landing page solo nel caso in cui siano stati specificati percorsi per l'edificio in cui l'utente si trova. In caso contrario il bottone viene nascosto.

3.10 Impostazioni

La schermata dedicata alle impostazioni è accessibile direttamente dalla landing page e consente la modifica di tutte preferenze della app:

1. E' possibile richiedere solo percorsi accessibili per utenti con disabilità motorie che fanno uso di sedia a rotelle.
2. La voce che fornisce le indicazioni durante la navigazione può essere disattivata.
3. E' possibile disabilitare la funzione di rilevamento automatico dell'edificio tramite GPS, rendendo obbligatorio specificarlo manualmente.
4. Il tema scuro per la navigazione in condizioni di scarsa luminosità può essere attivato.
5. Esistono due bottoni per accedere alle sezioni "credits" e "about" della applicazione.

Non sono presenti impostazioni riguardanti l'accessibilità per utenti con disabilità visive in quanto è sufficiente che lo screen reader del dispositivo sia acceso per poter utilizzare la applicazione sfruttando tale strumento.

3.11 Server

L'architettura del sistema prevede un server con una triplice funzione: conservare i dati sugli edifici mappati, inviarli ai device che ne fanno richiesta e consentirne la modifica via web. L'implementazione è stata fatta in linguaggio php col supporto del framework Symfony e la creazione del file .JSON da inviare ai device che ne fanno richiesta è demandata alla classe JsonController.

```
1 private function prepareBuilding(Building $building, Request
   $request, UploaderHelper $uploader): array {
2     $data = [
```

```
3     'name' => $building->getName(),
4     'latitude' => $building->getLatitude(),
5     'longitude' => $building->getLongitude(),
6     'proximityUUID' => $building->getUuid(),
7     'beacons' => [],
8     'nodes' => [],
9     'arcs' => [],
10 ];
11 foreach($building->getBeacons() as $beacon) {
12     $data['beacons'][] = $this->prepareBeacon($beacon);
13     foreach($beacon->getNodes() as $node) {
14         $data['nodes'][] = $this->prepareNode($node,
15             $request, $uploader);
16     }
17     foreach ($beacon->getOutgoingArcs() as $arc) {
18         $data['arcs'][] = $this->prepareArc($arc,
19             $request, $uploader);
20     }
21 }
22 return $data;
23 }
```

Listing 3.11: Costruzione file JSON

3.11.1 Database

Il database è stato realizzato in linguaggio SQL e mette a disposizione una tabella per ogni entità del dominio: edifici, beacon, nodi, archi e immagini. Gli archi vengono salvati così come appaiono nel file, cioè una sola volta e con una sola direzione, lasciando al device il compito di calcolare gli archi con direzione invertita.

3.11.2 Interfaccia web

L'interfaccia consente l'inserimento la modifica e l'eliminazione di tutte le entità. Esiste una home page che presenta il progetto mediante testo e

screenshot ed una pagina dedicata ad ogni entità, ad eccezione delle immagini (mappe) che vengono gestite nella pagina dedicata agli archi essendo ad essi direttamente associate. In fase di creazione degli archi è possibile specificare, oltre al beacon di origine e a quello di destinazione, l'accessibilità. I colori dell'interfaccia sono il bianco e il nero per garantire semplicità e facilità di lettura. Il colore rosso è stato utilizzato per evidenziare i bottoni.

Capitolo 4

Test di interazione con l'utente

4.1 Human-Computer Interaction

Negli anni settanta con il termine "ergonomia" si indicavano due diversi concetti. Il primo identificava gli studi sul design degli oggetti della vita quotidiana, mentre il secondo, anche detto Man-Machine Interaction, riguardava l'usabilità degli oggetti di lavoro come macchine, computer, ecc. Negli anni ottanta, con l'assunzione di un ruolo sempre più centrale da parte dei computer nel lavoro e nella vita quotidiana, un nuovo termine è nato per identificare gli studi riguardanti nello specifico la progettazione di software ed hardware, l'Human-Computer Interaction. L'HCI si focalizza sugli utenti che lavorano ed interagiscono con i computer, piuttosto che con altri tipi di macchine o prodotti di design. L'interazione è una attività di comunicazione iniziata dall'utente e finalizzata ad un preciso scopo. L'interfaccia invece è il luogo dell'interazione, lo strumento abilitante che funge da punto di contatto. L'ACM (Association for Computing Machinery) definisce l'Human-Computer Interaction come *"una disciplina che si occupa della progettazione, valutazione e implementazione di sistemi informatici interattivi per uso umano e con lo studio dei principali fenomeni che li circondano."* [39]. L'HCI si occupa dunque della ricerca nel campo del design e dell'utilizzo di tecnologie informatiche, con attenzione al tema delle interfacce tra

utente e computer e si colloca a cavallo tra varie discipline come: la computer science, le scienze comportamentali, il design ed altre. La maggior parte della ricerca e delle pubblicazioni in tale ambito si concentra sul miglioramento dell'HCI tramite un incremento dell'usabilità delle interfacce trattando, tra le altre cose:

1. Metodi per la progettazione di nuove interfacce ottimizzate.
2. Metodi per l'implementazione delle interfacce progettate.
3. Metodi per la valutazione e il confronto delle interfacce rispetto alla loro usabilità.

Dalla nascita dell'HCI ad oggi numerose metodologie per la progettazione, spesso basate su un modello di interazione tra utente e sistema, si sono delineate. In principio esse consideravano prevedibili i processi cognitivi degli individui incoraggiando i progettisti a sviluppare le interfacce basandosi sulle evidenze delle scienze cognitive. In tempi più recenti i modelli sono nati a partire dai feedback dell'utente finale, coinvolto in un dialogo con i progettisti, dando origine a sistemi realizzati a partire dal tipo di esperienza che l'utilizzatore desidera e si aspetta. Le principali metodologie sono:

1. Activity theory: delinea e studia il contesto ove l'interazione dell'uomo con il computer avviene. Fornisce un framework per ragionare e valutare le attività che in questo contesto si svolgono, oltre a strumenti di analisi come liste di situazioni ed elementi che devono essere considerati nello studio e nella progettazione dell'interazione, con un focus costante sulle attività [40].
2. Principles of user interface design: si tratta di una serie di principi che possono essere considerati durante la progettazione di una interfaccia utente proposti dagli ingegneri Larry Constantine e Lucy Lockwood [41]. La loro applicazione non è vincolata ad un particolare ordine e riguardano concetti come la semplicità, la tolleranza, la consistenza, la struttura il riuso e i feedback.

3. Value sensitive design (VSD): il valore denota l'importanza di una qualche cosa o di un qualche comportamento col fine di determinare quale sia la migliore azione da compiere. Il value sensitive design è quindi metodo per costruire tecnologie che tengono conto dei valori delle persone che usano direttamente la tecnologia e di coloro che dalla tecnologia sono influenzati. Il processo tipico di progettazione di VSD è iterativo e prevede una indagine concettuale, una empirica e una tecnica [42]. L'indagine concettuale mira a delineare e comprendere i vari soggetti interessati dalla tecnologia ed i loro valori. L'indagine empirica racchiude le ricerche atte a facilitare la comprensione, da parte dei progettisti, delle esigenze e dei valori degli utenti. Infine l'indagine tecnica riguarda il design di sistemi a supporto dei valori identificati in precedenza.
4. User-centered design (UCD): è una metodologia moderna e tra le più utilizzate, basata sul principio che l'utente sia il fulcro della progettazione di un sistema informatico. L'utente è parte attiva assieme a progettisti e sviluppatori e lavora con essi aiutandoli ad individuare le esigenze e le problematiche dell'utilizzatore finale.

4.1.1 User-Centered Design

L'User-Centered Design può essere considerato un framework che pone l'usabilità, l'ambiente, i task e soprattutto le caratteristiche dell'utente al centro dell'attenzione durante tutti gli stadi della progettazione. L'UCD consiste in un processo di problem-solving in più fasi che richiede ai progettisti di analizzare e ipotizzare le modalità con cui gli utenti utilizzeranno il prodotto e, successivamente, di confermare le ipotesi ad esempio tramite dei test. Questi test sono previsti in ogni fase, dalla raccolta dei requisiti fino a dopo l'implementazione per garantire che il focus rimanga sull'utente finale e sulle sue esigenze portando alla realizzazione di sistemi utili e usabili. La sostanziale differenza tra l'User-Centered Design e le altre metodologie consiste nell'ottimizzazione del prodotto a partire da ciò che all'utente serve, interes-

sa, desidera e si aspetta durante l'uso del sistema. Tale metodologia evita situazioni in cui l'utente sia costretto a modificare significativamente il suo comportamento per adattarsi al software. Il termine User-Centered Design è stato utilizzato per la prima volta da Donald A. Norman, diventando famoso grazie alla pubblicazione intitolata "User-Centered System Design: New Perspectives on Human-Computer Interaction" [43]. Oggi è globalmente condiviso un processo comune per l'UCD e prevede fasi di analisi, progettazione, valutazione da svolgersi in maniera iterativa.

User analysis

In questa fase ci si occupa di delineare un contesto di utilizzo, identificando le principali categorie di utenti. Per ognuna di esse è necessario comprendere perché utilizzerebbe il prodotto, come, e di cosa necessita.

Durante tale processo ci si può servire di *personas*. Una *persona* rappresenta un utente ed è utilizzata come modello per semplificare i processi decisionali riguardanti funzionalità, interazione e design del prodotto. In molti casi vengono estratte da una serie di interviste a persone reali, poi sintetizzate e riportate in un breve testo descrittivo del comportamento, degli obiettivi, delle capacità e dell'ambiente in cui queste ultime si trovano. Alcuni dettagli di fantasia vengono aggiunti per donare realismo al profilo realizzato.

Il concetto di *personas* fu proposto da Alan Cooper, un noto sviluppatore software, il quale cominciò ad utilizzarne una prima versione nel 1983. Cooper sfruttava i dati raccolti da interviste informali di 7/8 possibili utenti. Successivamente ha spostato il focus su come uno specifico utente si sarebbe comportato, piuttosto che un gruppo, nel rapportarsi con il software [44]. Questa tecnica è diventata popolare a partire dal 1999, anno in cui fu pubblicato il libro "The Inmates are Running the Asylum". In questa pubblicazione Cooper presenta le best practices per la creazione e l'uso di *personas* da parte dei progettisti [45].

Generalmente si delinea un piccolo set di *personas* per ogni prodotto (o per ogni funzionalità significativa), una di queste è, solitamente, il focus principale del progetto. Le altre sono chiamate *secondary personas* e identificano individui che non sono il target principale, ma che possono trarre beneficio dalla realizzazione del progetto e le cui necessità devono essere soddisfatte ove possibile. Le *secondary personas* aiutano ad analizzare alcuni aspetti che possono essere tralasciati anche se per la *persona* principale sono soddisfatte tutte necessità. Esiste anche un concetto di anti-persona che delinea l'utente per il quale il prodotto non è pensato.

Utilizzare *personas* significa contribuire alla creazione di una idea definita di gruppo di utenti attorno al quale il prodotto deve essere realizzato. Aiutano inoltre a capire quali funzionalità siano indispensabili per l'utente e quali esulino dal contesto principale. Disegnano un volto dell'utente riassumendo le caratteristiche raccolte con le interviste e consentono l'immedesimazione ed una qualche forma di empatia da parte dei progettisti. Sebbene questi artefatti possono facilmente ricadere in forme stereotipate di utenti con caratteristiche poco accentuate, se ben realizzati consentono di prendere decisioni di design oculate ed in tempi brevi semplificando il processo intero.

Gli aspetti più importanti delle *personas* sono quindi:

1. Condivisione all'interno di un team di lavoro di una idea comune di utente. Questo fattore semplifica la comprensione del problema e facilita l'interazione e la comunicazione tra i membri del team stesso dando origine a sistemi coerenti.
2. Soluzioni proposte sempre in relazione alle esigenze degli utenti, tema chiave dell'User-Centered Design. Le varie funzionalità possono essere valutate in base al numero di esigenze soddisfatte.
3. Delineare un volto umano aumentando l'empatia verso le persone rappresentate dai dati raccolti.

Questo progetto ha fatto uso di tre *personas*, riportate nei capitoli precedenti. Tale scelta ha consentito di sviluppare un software coerente con le richieste e che non presenti funzionalità sovrabbondanti.

Design

Un approccio molto utilizzato nel design di progetti che vogliono mantenere in primo piano gli aspetti riguardanti l'interazione uomo-macchina e quello del design partecipativo (Participatory Design). Utenti reali vengono coinvolti attivamente nel processo, spesso raggruppati per caratteristiche comuni come età, interessi, ecc... Alcune metodologie a supporto della progettazione sono:

1. Focus group: è una metodologia utilizzata per evidenziare punti di forza e di debolezza nelle varie funzionalità della applicazione. I focus group vengono condotti con gruppi di utenti di 8/15 persone ai quali vengono proposte le varie funzionalità, eventualmente mediante l'uso di storyboard. I partecipanti sono liberi di discutere i temi proposti e di fornire feedback. Le sedute durano massimo due ore e si concludono con l'analisi dei risultati e la stesura di un report. Questa tecnica è utilizzata sia per definire aspetti di design e funzionalità che per il marketing.
2. Experience prototyping: si tratta di una metodologia utile ad evidenziare, agli occhi dei progettisti, il punto di vista degli utenti tramite una esperienza in un contesto reale. Consiste in una simulazione realistica che avviene con l'uso di una demo o di disegni e sketch. Questa metodologia è usata principalmente nel mondo mobile. Si svolge con una simulazione di come gli utenti userebbero il prototipo per concludere il task assegnato e termina con l'analisi dei risultati e la stesura di un report.
3. Interface Development principles: sono principi da considerare nell'implementazione di interfacce utente e garantiscono un prodotto finale

semplice e usabile. Il principio KISS (Keep It Simple Stupid), ad esempio, indica la necessità di scrivere codice minimale, senza complicazioni non necessarie. Il principio KISS è valido per la stesura di codice sia riguardante interfacce che non. Un altro principio è Less-Is-More. E' fortemente legato a KISS e precisa che per la realizzazione di una buona interfaccia grafica è necessario rimuovere tutti gli elementi che non sono indispensabili all'interazione con l'utente. Altri principi sono, Responsive Design, Mobile First, ecc...

Focus group ed experience prototyping sono stati studiati e applicati dall'autore di questa tesi nell'ambito del progetto "ilocalapp" [47], al quale ha partecipato come unico sviluppatore [50, 51]. Tale progetto nato all'interno del programma Erasmus+ e supportato dall'Unione Europea, ha previsto la progettazione e la realizzazione di una applicazione per dispositivi mobili con la finalità di agevolare la permanenza di studenti stranieri nella città di Bologna (e in altre 3 città europee), fornendo loro informazioni pratiche sulla vita quotidiana e universitaria, unendo aspetti di apprendimento incidentale della lingua italiana (finlandese, portoghese e polacca). Le università coinvolte sono: Alma Mater Studiorum Università di Bologna come capo progetto, Lapin Yliopisto University of Lapland, Centre for Social Studies Universidade de Coimbra, Adam Mickiewicz University Poznan. Tale progetto ha inoltre utilizzato meccanismi di localizzazione GPS e tenuto in primo piano gli aspetti di interazione uomo/macchina. Ciò ha consentito il riutilizzo di alcune nozioni apprese durante lo sviluppo, avvenuto in collaborazione con gli utenti finali, semplificando il processo di design di questa applicazione. Inoltre i principi KISS e Less-Is-More sono stati adottati e hanno guidato l'implementazione dell'interfaccia grafica.

Evaluation

1. Heuristic evaluation: si tratta di uno strumento di controllo dell'usabilità nei sistemi software. L'interfaccia viene analizzata e valutata in base ad alcuni principi di usabilità (heuristics). Questo metodo è

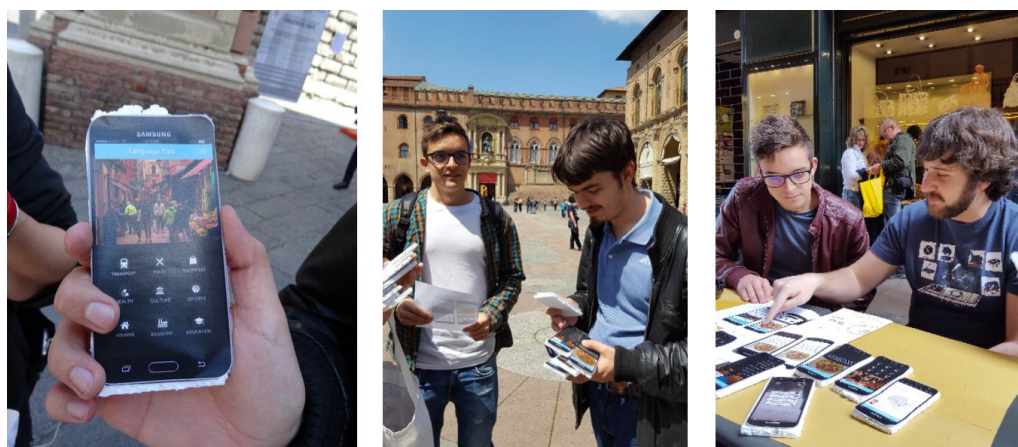


Figura 4.1: Experience prototyping durante il progetto ilocalapp

stato sviluppato da Jakob Nielsen ed ha lo scopo di rilevare qualunque problema legato all'interfaccia utente. Non sono necessari utenti per svolgere questo tipo di valutazione, basta seguire uno o più dei set di euristiche come, ad esempio, quello proposto da Nielsen stesso per l'analisi delle interfacce.

2. Usability testing: per valutare una interfaccia, i relativi meccanismi di interazione ed i servizi offerti da una applicazione ci si può servire di test di usabilità. Questi test coinvolgono utenti reali i quali utilizzano il sistema svolgendo una serie di task all'interno di uno scenario stabilito in modo da creare una situazione realistica. Ulteriori informazioni possono essere estratte tramite l'uso di questionari pre/post test.

Nell'ambito di questo progetto sono stati effettuati test di usabilità in più contesti e con differenti categorie di utenti per poter valutare la qualità dell'interazione tra individuo e sistema. I dettagli dei test sono riportati nelle sezioni seguenti.

4.2 Venues

I test si sono svolti in due edifici dell'Università di Bologna.

1. Il primo è Palazzo Poggi, sede centrale dell'università, ospita il rettore e numerosi musei come il Museo di palazzo Poggi, il Museo della Specola, il Museo delle Navi e delle Antiche Carte Geografiche. Si tratta di un edificio ubicato in via Zamboni 33 a Bologna, costruito a metà del '500, abitazione di Alessandro Poggi. Al suo interno sono presenti un'aula intitolata a Giosuè Carducci, dove il poeta stesso tenne lezioni per 40 anni, e la famosa statua dell'Ercole scolpita da Angelo Piò nel 1730. L'edificio è caratterizzato da ampi corridoi e scalinate. La pianta è complessa a causa di svariati interventi di ampliamento e modifica avvenuti nei secoli. Attualmente al suo interno si trovano portineria, segreteria, uffici, aule, musei e servizi. Inoltre le opere d'arte ospitate sono numerose.
2. Il secondo edificio è la sede della Scuola di Lettere e Beni Culturali dell'Università di Bologna. Si trova in via Zamboni 38 a Bologna e ospita anche la biblioteca Felice Battaglia e il dipartimento di Filosofia. Interventi recenti lo hanno dotato di ascensori e rampe per motivi di accessibilità, ma i 4 piani e i numerosi ambienti rendono la pianta complessa. Non tutti gli ascensori sono adatti ad ospitare una sedia a rotelle. Gli ambienti sono privi di opere d'arte di rilievo.

I due edifici sono stati ritenuti adatti per ospitare i test sia per la loro conformazione tipica degli edifici storici, quindi ampia e complessa, sia per le diverse tipologie di ambienti ed opere ospitate. Nel primo edificio si sono svolti infatti i test con particolare riguardo per le tematiche turistico/culturali mentre nel secondo la maggior parte dei test hanno mirato agli aspetti di navigazione pura per verificare la validità del prodotto nel caso di utilizzo da parte di studenti e personale.

4.3 Categorie di utenti

I test sono stati condotti su 3 categorie di utenti:

1. Studenti dell'università di Bologna sia originari della città che non. Nei test è stato incluso uno studente non vedente.
2. Personale docente dell'università di Bologna e figure di governance esperte di accessibilità.
3. Personale tecnico/amministrativo con ruoli di portineria direttamente interessati in quanto possibili promotori del servizio offerto dall'applicazione e beneficiari del suo utilizzo in quanto alleggeriti da parte dei compiti di erogazione di indicazioni agli utenti.

4.4 Tipologia di valutazione

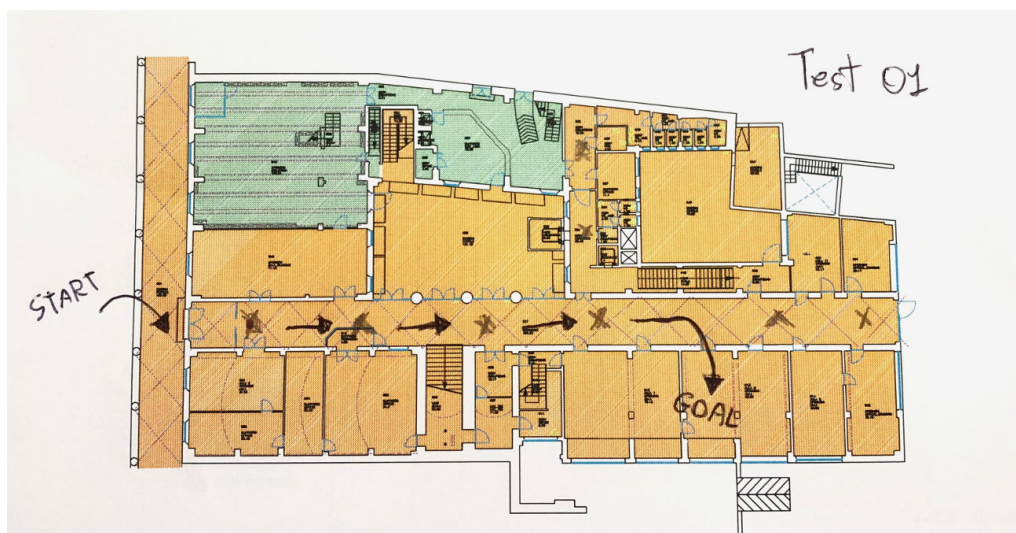


Figura 4.2: Esempio di percorso di test

E' stata effettuata una valutazione di tipo qualitativo mediante test di usabilità con specifici task [46]. La funzionalità oggetto principale dei test è

stata quella di navigazione comprensiva di opzioni turistico/culturali. Sono state proposte alcune destinazioni agli utenti i quali hanno autonomamente impostato il luogo di arrivo ed hanno seguito le indicazioni erogate. Presso palazzo Poggi, lungo il percorso, è stata testata la possibilità di conoscere informazioni turistico/culturali in fase di navigazione. Il numero di utenti coinvolti è 8. Secondo gli studi proposti da Nielsen Norman Group, 5 utenti sono sufficienti per cogliere l'80% degli aspetti di usabilità rilevanti. 8 utenti portano la percentuale al 90% circa [48, 49]. Questi valori sono stati elaborati nel corso degli anni a partire dagli studi di Jakob Nielsen. I suoi lavori presentano anche il numero di utenti necessari per eseguire test quantitativi anziché qualitativi e riportano le formule per calcolare il livello di accuratezza che un test può raggiungere. La necessità di partire da test di usabilità qualitativi è messa in primo piano, lasciando l'utilizzo di test quantitativi alle fasi successive.

4.5 Feedback

I feedback sono stati raccolti con think-aloud protocol unitamente ad una intervista strutturata post test. I feedback sono stati organizzati per temi come segue:

1. Interfaccia grafica ed interazione: La totalità degli utenti si è detta soddisfatta dell'aspetto dell'interfaccia grafica e dei meccanismi di interazione proposti. In particolare la semplicità e la chiarezza delle informazioni e lo stile minimal con colori Unibo sono stati apprezzati.
2. Accessibilità dell'interfaccia: L'interfaccia è risultata idonea per un utilizzo tramite screen reader, senza presentare difformità rispetto a ciò che l'utente prevedeva.
3. Goal/purpose dell'applicazione: Gli utenti concordano sulla utilità della applicazione per quanto riguarda gli aspetti turistico/culturali, ma sottolineano che la funzionalità chiave sia quella di navigazione.

4. Target user suggeriti/individuati: I target user individuati in fase di analisi sono stati confermati in fase di test. Utenti con disabilità visive e motorie risultano essere i principali beneficiari del prodotto. Inoltre è stata fatta notare l'utilità della applicazione nel caso di studenti provenienti da altre città e appena trasferiti a Bologna.
5. Possibili miglioramenti: La applicazione risulta completa. Un miglioramento suggerito è quello di integrare il software con strumenti per la navigazione outdoor in modo da rendere possibile una navigazione indoor/outdoor uniforme.



Figura 4.3: Test presso Lettere e Beni Culturali

4.6 Altri test

Altri test interni sono stati eseguiti durante lo sviluppo della applicazione e non hanno coinvolto utenti finali. Si tratta principalmente di test sul segnale emesso dai beacon con la finalità di individuare la migliore strategia di localizzazione dell'utente. Inoltre è stato necessario studiare tramite test

quale sia il miglior modo per fissare gli emettitori con particolare attenzione all'altezza da terra.

Capitolo 5

Sviluppi futuri

In questo capitolo vengono presentati alcuni possibili sviluppi del sistema. Si tratta di una lista non esaustiva e da considerarsi come suggerimento per futuri ampliamenti e migliorie. Tale elenco viene espresso come insieme di interventi che si sono dimostrati, nel corso dell'implementazione, possibili ed interessanti in quanto migliorativi del prodotto.

5.1 Mappe SVG

L'adozione di mappe in formato .png codificate Base64 è stata, come precedentemente indicato, una scelta dettata dalla necessità di trattare formati semplici e utilizzabili da personale senza specifiche competenze informatiche. Il formato .png è infatti tra i più diffusi per quanto riguarda le immagini digitali ed è supportato dalla quasi totalità dei software. Inoltre spesso le piantine degli edifici sono già disponibili in tale formato e possono essere direttamente utilizzate come mappe all'interno del sistema. L'eventuale transizione dal formato .png al formato .svg, porta uno svantaggio (mitigabile) in termini di usabilità da parte di chi caricherà le mappe, ma implica alcuni importanti vantaggi:

1. Il formato .svg consente un notevole risparmio di spazio di archiviazione se paragonato a .png. Ciò significa minore spazio occupato sul server

e trasmissione di file JSON più piccoli verso i device.

2. E' inoltre possibile erogare immagini con risoluzione maggiore senza peggiorare le performance del sistema.
3. In un'ottica di coerenza dei formati, .svg è la naturale rappresentazione di immagini in un ambiente che utilizzi i linguaggi tipici del web e risulta facilmente gestibile da codice JavaScript.
4. Trattandosi di testo in formato XML è integrabile all'interno dei file JSON senza necessità di conversioni Base64 o simili.

La gestione di file in formato .svg da parte di chi caricherà le mappe potrebbe essere semplificata fornendo la possibilità di caricare direttamente il file nel giusto formato oppure in formati alternativi (.jpg, .png) per poi eseguire una conversione automatica.

5.2 Here Maps

Una possibile alternativa alla transizione da .png a .svg è quella di integrare un sistema di navigazione esistente. Tra quelli analizzati nell'ambito di questo progetto è risultato adatto allo scopo il framework "Here Maps". Include API per Android e iOS, tool per il caricamento di mappe indoor e la possibilità di sfruttare il sistema di rilevamento beacon compreso nel pacchetto. L'adozione di tale framework garantisce vantaggi quali l'utilizzo di codice sviluppato a livello aziendale e costantemente migliorato, semplificazione del codice JavaScript, ecc.

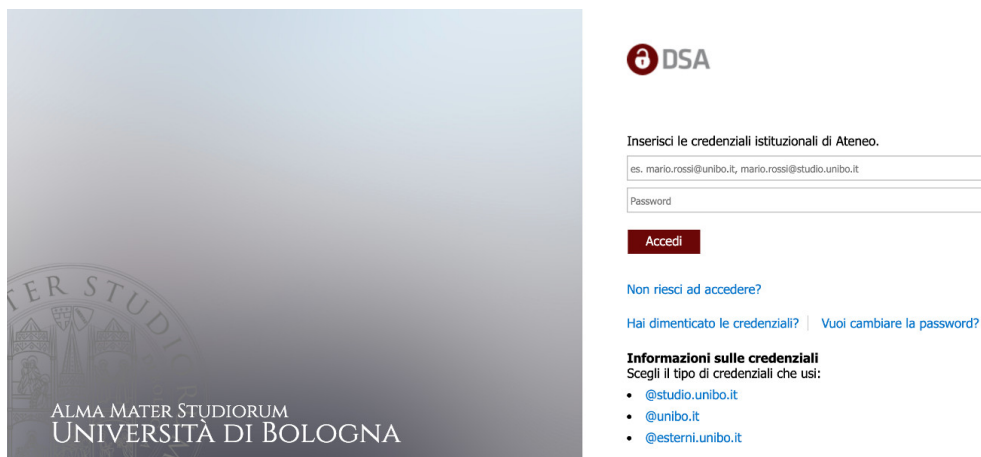
Sono però da considerare alcuni fattori che potrebbero rivelarsi controproducenti come l'impossibilità di personalizzare nel dettaglio il comportamento della app e la dipendenza da scelte e politiche altrui. Inoltre Here Maps non fornisce un component per React Native compatibile con entrambi i sistemi operativi. Ciò implica la necessaria scrittura di codice nativo per realizzarne uno ad hoc.

5.3 Report

L'accessibilità dei percorsi è in larga parte influenzata dalla presenza di oggetti lungo il tragitto che possono fungere da ostacoli o da facilitatori. Tali oggetti possono essere spostati, aggiunti, rimossi in tempi brevi modificando così l'accessibilità degli edifici. L'aggiunta di una rampa può rendere accessibile un corridoio, così come un mobile o un nuovo muro possono intralciare il transito. Per questo motivo trova utilità l'implementazione di un sistema per la segnalazione di situazioni di accessibilità discordanti tra i dati forniti dalla applicazione e la realtà.

Ciò può essere realizzato con l'utilizzo di un bottone apposito che attivi la trasmissione al server delle informazioni necessarie a risalire al preciso luogo in cui è stata rilevata l'anomalia.

5.4 Login Unibo



DSA

Inserisci le credenziali istituzionali di Ateneo.

es. mario.rossi@unibo.it, mario.rossi@studio.unibo.it

Password

Accedi

Non riesci ad accedere?

[Hai dimenticato le credenziali?](#) | [Vuoi cambiare la password?](#)

Informazioni sulle credenziali
Scegli il tipo di credenziali che usi:

- @studio.unibo.it
- @unibo.it
- @esterni.unibo.it

Figura 5.1: Form di login dell'Università di Bologna

Il form di login dell'Università di Bologna può trovare utilità per quanto riguarda l'accesso all'interfaccia web per l'esecuzione di operazioni CRUD sul database. Oltre agli ovvi vantaggi in termini di sicurezza garantiti da un

sistema di gestione degli accessi sviluppato per l'intera università, l'integrazione di tale servizio consente l'utilizzo delle credenziali Unibo per il personale interno senza necessità di ulteriori registrazioni di account appositi.

Il form, se utilizzato anche all'interno della applicazione, potrebbe consentire per gli utenti in possesso di un account di effettuare l'accesso, con il fine di memorizzare le proprie preferenze sui server Unibo consentendo il cambio del device senza la perdita delle proprie impostazioni.

5.5 Analisi statistica

Una ulteriore funzionalità implementabile, abilitata dalla presenza di un server, è quella della rilevazione statistica di dati legati al posizionamento degli utenti all'interno degli edifici. L'invio di dati anonimi raccolti dal device potrebbe aiutare nello studio di innumerevoli comportamenti frequenti tenuti dagli utenti come:

1. Destinazioni più richieste.
2. Edifici più frequentati.
3. Funzionalità più utilizzate.
4. Preferenze e impostazioni più rilevanti.

E' possibile che tale funzionalità implichi l'obbligo di richiesta del consenso al trattamento dei dati personali dell'utente in forma anonima.

5.6 Integrazione con software per la navigazione outdoor

E' contemplabile l'integrazione con ulteriore software di navigazione specializzato in spostamenti outdoor per poter erogare un servizio utilizzabile anche negli spostamenti tra un edificio ed un altro. Un prodotto adatto a

tale scopo è il progetto realizzato dal dottor De Vita [25]. Il software indicato utilizza già il framework Here Maps, è accessibile ed è specializzato negli spostamenti tra i vari edifici universitari. Sono da verificare l'effettiva necessità di tale integrazione e la compatibilità tra i due prodotti.

Conclusioni

E' stato sviluppato un sistema di navigazione indoor per smartphone. Esso fa uso di emettitori bluetooth detti beacon per rilevare la posizione dell'utente consentendo l'erogazione di indicazioni precise e puntuali. Il processo di design si è sviluppato a partire dallo studio delle tecnologie di posizionamento e wayfinding esistenti con particolare attenzione per l'accessibilità. Sono state adottate metodologie tipiche dell'User-Centered Design come l'uso di textitpersonas e l'esecuzione di test di usabilità. E' stata implementata una applicazione disponibile sia per dispositivi Android che iOS mediante l'utilizzo del framework React Native con l'ausilio di componenti esistenti. Il prodotto finale è un software dalle prestazioni native in grado di guidare gli utenti all'interno degli edifici mappati. Sono stati inclusi aspetti turistico/culturali ritenuti utili in edifici storici quali quelli dell'Università di Bologna. Il sistema comprende anche un server con relativo database e una interfaccia web per l'inserimento di nuove mappe. Il lavoro ha riguardato lo studio dei vari approcci alla localizzazione ed ha adottato lo schema basato su proximity. L'erogazione dei contenuti avviene in maniera visiva e sonora. Gli aspetti di accessibilità sono stati valutati e realizzati per consentire piena compatibilità con gli screen reader garantendo così l'usabilità da parte di utenti con disabilità visive. Gli utenti con disabilità motorie, invece, dispongono di una impostazione dedicata che esclude i passaggi non accessibili dai tragitti. Per questi motivi il prodotto è stato presentato all'ufficio del servizio per gli studenti con disabilità dell'Università di Bologna che ha seguito con interesse il processo di sviluppo e di test. I test eseguiti sono di tipo qualitativo e

costituiscono una valida overview sulle caratteristiche di usabilità della applicazione, dimostrando inoltre l'effettiva efficienza del sistema. Si tratta di test di usabilità eseguiti in diversi ambienti e con 3 differenti categorie di utenti rispettando le linee guida Nielsen. Ulteriori test di tipo quantitativo sono auspicabili, così come alcuni sviluppi futuri in accordo con i feedback ricevuti in fase di test e con le possibilità offerte dal mercato. Nuovo codice per la navigazione outdoor può essere integrato e numerose modifiche possono essere apportate per assecondare esigenze future grazie alla struttura basata su componenti del codice prodotto. Il team si dice soddisfatto del risultato ottenuto e gli sviluppatori hanno rilevato errori di localizzazione conformi con le specifiche. Tempi e richieste sono stati rispettati.

Bibliografia

- [1] Hideyuki Iwahashi. *Toward white wave - Story of Seiichi Miyake*. Traffic Safety Research Center. 1983.
- [2] Madoka Nakajima and Shinichiro Haruyama. *New indoor navigation system for visually impaired people using visible light communication*. EURASIP Journal on Wireless Communications and Networking. 2013.
<https://doi.org/10.1186/1687-1499-2013-37>
- [3] Dragan Ahmetovic , Cole Gleason, Chengxiong Ruan, Kris Kitani, Hironobu Takagi, Chieko Asakawa. *NavCog: a navigational cognitive assistant for the blind*. Proceedings of the 18th International Conference on Human-Computer Interaction with Mobile Devices and Services. 2016.
<https://dl.acm.org/citation.cfm?id=2935361>
- [4] Dragan Ahmetovic , Cole Gleason, Kris Kitani, Hironobu Takagi, Chieko Asakawa. *NavCog: turn-by-turn smartphone navigation assistant for people with visual impairments or blindness*. Proceedings of the 13th Web for All Conference. 2016.
<https://dl.acm.org/citation.cfm?doid=2899475.2899509>
- [5] Michele A. Williams, Amy Hurst, Shaun K. Kane. *Pray before you step out: describing personal and situational blind navigation behaviors*. Proceedings of the 15th International ACM SIGACCESS Conference on Computers and Accessibility. 2013.
<https://dl.acm.org/citation.cfm?id=2513449>

- [6] Jobke Wentzel, Eric Velleman, Thea van der Geest. *Wearables for all: development of guidelines to stimulate accessible wearable technology design*. Proceedings of the 13th Web for All Conference. 2016.
<https://dl.acm.org/citation.cfm?id=2899496>
- [7] Jan B. F. Van Erp. *Presenting directions with a vibrotactile torso display*. Ergonomics. 2007.
<https://www.tandfonline.com/doi/abs/10.1080/0014013042000327670>
- [8] Roger W. Cholewiak, J. Christopher Brill, Anja Schwab. *Vibrotactile localization on the abdomen: Effects of place and space*. Perception & Psychophysics. 2004.
<https://link.springer.com/article/10.3758/BF03194989>
- [9] Jan B. F. Van Erp, Hendrik A. H. C. Van Veen, Chris Jansen, Trevor Dobbins. *Waypoint navigation with a vibrotactile waist belt*. ACM Transactions on Applied Perception. 2005.
<https://dl.acm.org/citation.cfm?id=1060585>
- [10] Wilko Heuten, Niels Henze, Susanne Boll, Martin Pielot. *Tactile wayfinder: a non-visual support system for wayfinding*. Proceedings of the 5th Nordic conference on Human-computer interaction: building bridges. 2008.
<https://dl.acm.org/citation.cfm?id=1463179>
- [11] German Flores, Sri Kurniawan, Roberto Manduchi, Eric Martinson, Lourdes M. Morales, Emrah Akin Sisbot. *Vibrotactile Guidance for Wayfinding of Blind Walkers*. IEEE Transactions on Haptics. 2015.
<https://ieeexplore.ieee.org/abstract/document/7060731>
- [12] Jizhong Xiao, Samleo L. Joseph, Xiaochen Zhang, Bing Li, Xiaohai Li, Jianwei Zhang. *An assistive navigation framework for the visually impaired*. IEEE Transactions on Human-Machine Systems. 2015.
<https://ieeexplore.ieee.org/abstract/document/7010003>

- [13] Thomas Gallagher, Elyse Wise, Binghao Li, Andrew G. Dempster, Chris Rizos, Euan Ramsey-Stewart. *Indoor positioning system based on sensor fusion for the Blind and Visually Impaired*. 2012 International Conference on Indoor Positioning and Indoor Navigation (IPIN). 2012.
<https://ieeexplore.ieee.org/abstract/document/6418882>
- [14] U.S. government, *GPS Accuracy*.
<https://www.gps.gov/systems/gps/performance/accuracy/>
- [15] Rick W. Sturdevant. *NAVSTAR, the Global Positioning System: A Sampling of Its Military, Civil, and Commercial Impact* History of Nasa Chapter 17. 1994.
<https://history.nasa.gov/sp4801-chapter17.pdf>
- [16] Catia Prandi, Valentina Nisi, Paola Salomoni, Nuno Jardim Nunes. *From gamification to pervasive game in mapping urban accessibility* Proceedings of the 11th Biannual Conference on Italian SIGCHI. 2015.
<https://dl.acm.org/citation.cfm?id=2808449>
- [17] Clare Southerton. *Zombies, Run!: Rethinking immersion in light of nontraditional gaming contexts* Transmedia Practice: A Collective Approach. 2014.
<https://www.researchgate.net/publication/304656926>
- [18] OSM Foundation. *OpenStreetMap Foundation Main Page*.
https://wiki.osmfoundation.org/wiki/Main_Page
- [19] P. Leach, M. Mealling, R. Salz. *A Universally Unique Identifier (UUID) URN Namespace*. 2005.
<https://tools.ietf.org/pdf/rfc4122.pdf>
- [20] Sakmongkon Chumkamon, Peranitti Tuvaphanthaphiphat, Phongsak Keeratiwintakorn. *A Blind Navigation System Using RFID for Indoor Environments*. 5th International Conference on Electrical Engineering/Electronics, Computer, Telecommunications and Information Tech-

- nology. 2008.
<https://ieeexplore.ieee.org/abstract/document/4600543>
- [21] José Faria, Sérgio Lopes, Hugo Fernandes, Paulo Martins, João Barroso. *Electronic white cane for blind people navigation assistance*. World Automation Congress. 2010.
<https://ieeexplore.ieee.org/abstract/document/5665289>
- [22] Fazli Subhan, Halabi Hasbullah, Azat Rozyyev, Sheikh Tahir Bakhsh. *Indoor positioning in Bluetooth networks using fingerprinting and lateration approach*. International Conference on Information Science and Applications. 2011.
<https://ieeexplore.ieee.org/abstract/document/5772436>
- [23] Fazli Subhan, Halabi Hasbullah, Azat Rozyyev, Sheikh Tahir Bakhsh. *Easily Guiding of Blind: Providing Information and Navigation - SmartNav*. International Wireless Internet Conference. 2014.
https://link.springer.com/chapter/10.1007/978-3-319-18802-7_18
- [24] Ye-Sheng Kuo, Pat Pannuto, Ko-Jen Hsiao, and Prabal Dutta. *Luxapose: indoor positioning with mobile phones and visible light*. Proceedings of the 20th annual international conference on Mobile computing and networking. 2014.
<https://dl.acm.org/citation.cfm?id=2639109>
- [25] Alessio De Vita. *Geolocalizzazione e wayfinding indoor e outdoor per utenti con disabilità*.
- [26] OpenstationMap. *OpenstationMap 0.9.0*.
<http://www.openstationmap.org/>
- [27] Giovanni Delnevo, Lorenzo Monti, Francesco Vignola, Paola Salomoni, Silvia Mirri. *AlmaWhere: A prototype of accessible indoor wayfinding and navigation system*. 15th IEEE Annual Consumer Communications

- & Networking Conference. 2018.
<https://ieeexplore.ieee.org/abstract/document/8319242>
- [28] Tadeu Zagallo. *Bridging in React Native. An in-depth look into React Native's core.* 2015.
<https://tadeuzagallo.com/blog/react-native-bridge/>
- [29] *A short Story about React Native.* JobNinja. 2018.
<https://jobninja.com/blog/short-story-react-native/>
- [30] Erwan Datin (MacKentoch). *react-native-beacons-manager.*
<https://github.com/MacKentoch/react-native-beacons-manager>
- [31] Viktor Nilsson (vnil). *react-native-simple-compass.*
<https://github.com/vnil/react-native-simple-compass>
- [32] Alberto Restifo (albertorestifo). *node-dijkstra.*
<https://github.com/albertorestifo/node-dijkstra>
- [33] Anton Krasovsky (ak1394). *react-native-tts.*
<https://github.com/ak1394/react-native-tts>
- [34] Wix.com (wix). *react-native-navigation.*
<https://github.com/wix/react-native-navigation>
- [35] Treccani *Accessibilità.*
<http://www.treccani.it/enciclopedia/accessibilita/>
- [36] Vincenzo Rubano. *L'(in)accessibilità degli articoli scientifici sul Web e l'uso di RASH e EPUB.*
<https://amslaurea.unibo.it/12281/>
- [37] Giacomo Mambelli *Progetto e realizzazione di una applicazione mobile a supporto di iniziative ed eventi culturali.*
<https://amslaurea.unibo.it/10985/>

- [38] Sapere.it *L'orientamento*. DeAgostini.
<http://www.sapere.it/sapere/strumenti/studiafacile/scienza/La-Terra/Il-pianeta>
- [39] Thomas T. Hewett , Ronald Baecker, Stuart Card, Tom Carey, Jean Gasen, Marilyn Mantei, Gary Perlman, Gary Strong, William Verplank. *ACM SIGCHI Curricula for Human-Computer Interaction*. 1992.
<https://dl.acm.org/citation.cfm?id=2594128>
- [40] Victor Kaptelinin. *Activity Theory*. The Encyclopedia of Human-Computer Interaction, 2nd Ed, Chapter 16.
<https://www.interaction-design.org/literature/book/the-encyclopedia-of-human-c>
- [41] Larry Constantine, Lucy Lockwood. *Software for use: a practical guide to the models and methods of usage-centered design*. 1999.
<https://dl.acm.org/citation.cfm?id=301248>
- [42] Batya Friedman, Peter H. Kahn, Jr. Alan Borning, Alina Huldtgren. *Value Sensitive Design and Information Systems*. Early engagement and new technologies: Opening up the laboratory. 2013.
https://link.springer.com/chapter/10.1007/978-94-007-7844-3_4
- [43] Donald A. Norman *User-Centered System Design: New Perspectives on Human-Computer Interaction*. 1986.
<https://www.taylorfrancis.com/books/e/9780367807320>
- [44] Alan Cooper. *The origin of personas*.
https://www.cooper.com/journal/2008/05/the_origin_of_personas
- [45] Alan Cooper. *The Inmates are Running the Asylum*. Macmillan Publishing Company Inc. 1999.
- [46] Jakob Nielsen. *Quantitative Studies: How Many Users to Test?*. Nielsen Norman Group. 2006.
<https://www.nngroup.com/articles/quantitative-studies-how-many-users/>

- [47] *Incidentally Learning Other Cultures And Languages through an APP*.
<http://www.ilocalapp.eu/>
- [48] Jakob Nielsen. *Why You Only Need to Test with 5 Users*. Nielsen Norman Group. 2000.
<https://www.nngroup.com/articles/why-you-only-need-to-test-with-5-users/>
- [49] Jakob Nielsen, Thomas K. Landauer. *A mathematical model of the finding of usability problems*. Proceedings of the INTERACT '93 and CHI '93 Conference on Human Factors in Computing Systems. 1993.
<https://dl.acm.org/citation.cfm?id=169166>
- [50] Antonella Valva, Giacomo Mambelli. *Travelling with the UniOn App: Itineraries of Language and Culture Through the City of Bologna*. Almatourism. 2018.
<https://almatourism.unibo.it/article/view/8438>
- [51] Silvia Mirri, Marco Rocchetti, Paola Salomoni, Giacomo Mambelli, Antonella Valva. *On the design of an app for foreign languages incidental learning*. IEEE Symposium on Computers and Communications (ISCC). 2017.
<https://ieeexplore.ieee.org/abstract/document/8024514>

Ringraziamenti

Alla fine di questo percorso desidero scrivere due righe per esprimere alcuni sentiti e dovuti ringraziamenti. Lo farò sfruttando uno dei miei doni: quello della sintesi.

Ringrazio la professoressa Silvia Mirri, relatrice di questa tesi di laurea, per gli insegnamenti e per l'aiuto che ho ricevuto durante gli anni presso l'università, per la disponibilità, la correttezza e per aver permesso la realizzazione di questo progetto.

Grazie a Paola Salomoni (prorettrice tecnologie digitali), Rabih Chattat (delegato del rettore per la disabilità), Elena Trombini (prorettrice studenti) per l'interesse manifestato nei confronti del progetto e per la partecipazione attiva in sede di test.

Un ringraziamento va ai miei genitori che mi hanno sostenuto moralmente ed economicamente, contribuendo alla mia formazione e permettendomi di raggiungere questo traguardo.

Un grazie anche ai miei amici, che mi hanno consentito di compiere questo percorso con serenità e che hanno condiviso con me le esperienze più belle di questi ultimi anni. Infine a tutti coloro che sono stati parte attiva della mia vita e che mi hanno supportato sia in ambito universitario che nella quotidianità.