

Alma Mater Studiorum · Università di Bologna

---

**SCUOLA DI SCIENZE**  
**Corso di Laurea in Informatica**

**Sistema di navigazione indoor per edifici pubblici basato su  
punti di riferimento**

Dal filo al nodo d'Arianna

**Relatore: Chiar.mo**  
**Prof. LUCIANO BONONI**

**Presentata da:**  
**MARCELLO MAZZOLENI**

**Sessione**  
**Anno Accademico 2019/2020**



# Indice

1. Introduzione.....	5
1.1. Stato dell'arte.....	9
2. Progettazione.....	13
2.1. Il sistema .....	13
3. Implementazione .....	15
3.1. Descrizione funzionamento lato utente .....	16
3.1.1. L'homepage.....	16
3.1.2. La pagina del risultato .....	17
3.1.3. Il pannello di controllo .....	20
3.1.4. L'applicazione android.....	22
3.2. Descrizione del funzionamento lato Server .....	23
3.2.1. La homepage .....	23
3.2.2. Il pannello di controllo .....	24
3.2.3. La pagina del risultato .....	28
3.2.3.3. La selezione della lingua.....	31
3.3. Lo stile .....	33
4. Test E Validazione Sperimentale.....	34
4.1. Pregi: .....	34
4.2. Difetti: .....	34
5. Conclusioni e sviluppi futuri.....	35
6. Conclusioni:.....	35
7. Bibliografia .....	37



# Indice immagini

Figura 1 Mappa di una parte dell'aeroporto di Francoforte.....	8
Figura 2 Da: H. Hile et al, 2008, Landmark-Based Pedestrian Navigation from Collections of Geo-tagged Photos .....	9
Figura 3 Da: R. U. Thoranaghatte et al, 2008, "Landmark based Augmented Reality endoscope system for sinus and skull-base surgeries" .....	11
Figura 4 Edsger Wybe Dijkstra.....	14
Figura 5 Homepage versione desktop.....	16
Figura 6 Richiesta della password .....	17
Figura 7 Esempio di risultato in senza aiuti per disabilità.....	18
Figura 8 Alla pressione di questo tasto la pagina si sposta automaticamente all'istruzione successiva.....	18
Figura 9 .....	19
Figura 10 Tasto "nuova ricerca" per tornare alla pagina principale .....	20
Figura 11 L'opzione "aggiungi" permette di inserire un nuovo Landmark nel sistema ....	21
Figura 12 Pannello di controllo .....	21
Figura 13 Screenshot della homepage vista dall'app.....	22
Figura 14 L'opzione "elimina" cerca tutte le occorrenze del nodo selezionato e le cancella dai file di sistema.....	26
Figura 15 Opzione "rinomina nodo" .....	27
Figura 16 Opzione "modifica arco" .....	27
Figura 17 Il file Main.py crea il contenuto del risultato e lo scrive nel file cont.html. ....	28
Figura 18 Esempio di pagina del risultato in modalità "difficoltà visive" .....	29
Figura 19 Esempio di percorso alternativo in caso di difficoltà motorie.....	31
Figura 20.....	31
Figura 21 Esempio di homepage visualizzata in tedesco.....	32
Figura 22 Confronto tra pannello di controllo visualizzato in modalità desktop (a sinistra) e mobile (a destra).....	33
Figura 23.....	36



# 1. Introduzione

Spesso i visitatori di determinate edifici pubblici si trovano in una situazione di disorientamento, soprattutto se il luogo al quale accedono è a loro sconosciuto.

La funzione che tradizionalmente adempie un addetto alle informazioni, spesso non è sufficiente ad ottimizzare la visita che va dall'ingresso principale fino ad uno specifico punto dell'edificio (ufficio, ambulatorio, gate aeroportuale, sportello) e che faccia perdere il minor tempo possibile all'utente.

Mentre nel caso in cui ci trovassimo in una città a noi sconosciuta, la tecnologia già offre soluzioni affidabili per risolvere il problema, come ad esempio il GPS ormai presente su tutti gli smartphone in commercio, non esistono altrettante soluzioni quando siamo all'interno di un edificio.

Fiumi d'inchiostro sono stati versati per esprimere il disagio di chi si trova in una situazione labirintica, in cui arrivare alla soglia di un edificio è in realtà solo una parte della strada per raggiungere la propria meta.

Oggi tutti possiedono uno smartphone ed è quindi possibile creare delle soluzioni informatiche per risolvere il problema dell'orientamento anche all'interno degli edifici.

In questo documento tratterò di come l'utilizzo della tecnologia "Landmark-based navigation" o in italiano "navigazione per punti di interesse" possa risolvere questo problema.

Personalmente mi sono trovato in una situazione di questo genere nel 2018 quando per raggiungere la città di Strasburgo, partendo da Bologna, mi sono affidato alla compagnia Lufthansa, senza sapere che lo scalo previsto all'aeroporto di Francoforte avrebbe comportato una serie di problemi di non facile soluzione riguardanti lo spostamento all'interno del più grande aeroporto d'Europa.

Nel caso specifico la difficoltà consisteva nel prendere una coincidenza passando nell'arco di mezz'ora da un gate aeroportuale alla fermata di una corriera.

L'unico strumento messo a disposizione dall'aeroporto era una mappa cartacea che mi obbligava a delle soste frequenti per essere consultata. Inoltre la mappa non evidenziava correttamente la necessità di spostarsi tra un piano e un altro rendendo ancor più difficile la navigazione.

Constatata l'inefficacia della cartina, mi sono affidato alle informazioni fornite dal personale dell'aeroporto. Le informazioni fornitemi sfruttavano i punti di riferimento più visibili della struttura, quali bar, negozi, scale mobili ed altri elementi non segnati sulla mappa per guidarmi fino alla mia destinazione.

Da questa esperienza ho compreso la sostanziale differenza tra le indicazioni di tipo "classico": percorsi tracciati su mappe bidimensionali o il seguire i cartelli informativi, e le istruzioni basate sui punti di riferimento.

Mentre le prime creavano grosse difficoltà soprattutto a causa del fatto che sono organizzate come un unico filo conduttore, perso il quale non si ha più modo di procedere, le seconde mi permettevano di ritrovare la mia strada non appena individuato un punto di riferimento tra quelli indicati. Tali punti erano particolarmente efficaci quando ci si immetteva in uno spazio molto ampio, dove le indicazioni tradizionali non erano sufficientemente espressive.

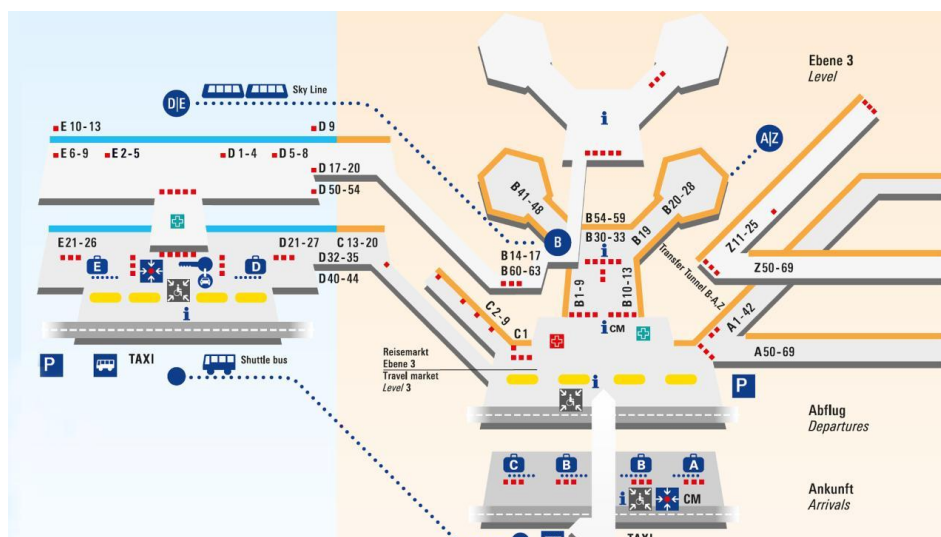


Figura 1  
Mappa di una parte dell'aeroporto di Francoforte



Se in quella situazione avessi avuto un sistema automatico di navigazione per punti di riferimento, avrei impiegato la metà del tempo per percorrere lo stesso tragitto e non avrei temuto neanche per un secondo di perdere la coincidenza.

## 1.1. Stato dell'arte

La navigazione per punti di interesse

è un modo di dare le indicazioni, sia ad un robot che ad una persona, che, invece di utilizzare istruzioni quali “svolta a destra sua via x” utilizza indicazioni basate su elementi dell’ambiente circostante, come ad esempio “svolta a destra dopo l’edicola”.

Questo modo di dare le indicazioni si rivela particolarmente importante quando ci si muove in un ambiente privo di vincoli: a differenza di quello che succede nel traffico veicolare dove ci si muove in ambienti predefiniti come le strade, è difficile invece indicare all’utente a piedi come raggiungere la destinazione desiderata.

L’idea di utilizzare un telefono cellulare per la Landmark based navigation, era stata già affrontata prima dell’avvento degli smartphone moderni [1].

Nell’articolo del 2008 “*Landmark-Based Pedestrian Navigation from Collections of Geo-tagged Photos*” veniva presentato un sistema che sfruttava una raccolta online di fotografie geo-localizzate per generare automaticamente istruzioni di navigazione per pedoni.

Il percorso veniva presentato all'utente come una sequenza di immagini relative a punti di riferimento, arricchite con istruzioni direzionali.

Come indicato da vari studi internazionali, l’utilizzo della navigazione per punti di riferimento, porta molteplici vantaggi: [2]–[8]

1) Più facili da seguire anche per le persone anziane



Figura 2

Da: H. Hile et al, 2008, *Landmark-Based Pedestrian Navigation from Collections of Geo-tagged Photos*

2) Riducono i tempi di navigazione

3) Riducono la confusione fornendo un feedback visivo istantaneo sul corretto avanzamento della navigazione

Motivato dagli innumerevoli vantaggi di questo tipo di navigazione, nell'articolo "A GIS data model for landmark-based pedestrian navigation" [9] si indica l'importanza dell'integrazione dei landmark nei modelli di navigazione pedonale.

La Landmark-based navigation viene utilizzata in svariati altri campi:

**Nell'industria** la navigazione basata su punti di riferimento di robot o veicoli mobili autonomi è stata ampiamente adottata. Tale strategia di navigazione si basa sull'identificazione e sul successivo riconoscimento di caratteristiche o oggetti distintivi dell'ambiente che sono noti a priori o estratti dinamicamente. Questo processo presenta difficoltà intrinseche nella pratica a causa del rumore del sensore e dell'incertezza ambientale.

Nell'articolo "*Landmark-based Navigation of Mobile Robots in Manufacturing*"[10] si affronta il problema di far muovere in modo autonomo, preciso ed economico, dei **robot** per la produzione industriale. I ricercatori hanno progettato un sistema che, utilizzando l'intelligenza artificiale unita ai sensori presenti sui robot, permetta loro di sapere sempre dove si trovano sulla mappa e di spostarsi utilizzando punti di riferimento dell'ambiente.

Similmente nell'articolo "Landmark-based autonomous navigation in sewerage pipes" [11] veniva affrontato il problema di far navigare autonomamente dei robot all'interno di tubature fognarie.

Anche in questo caso è stata necessaria l'integrazione di un'intelligenza artificiale per poter comprendere i dati acquisiti attraverso i sensori.

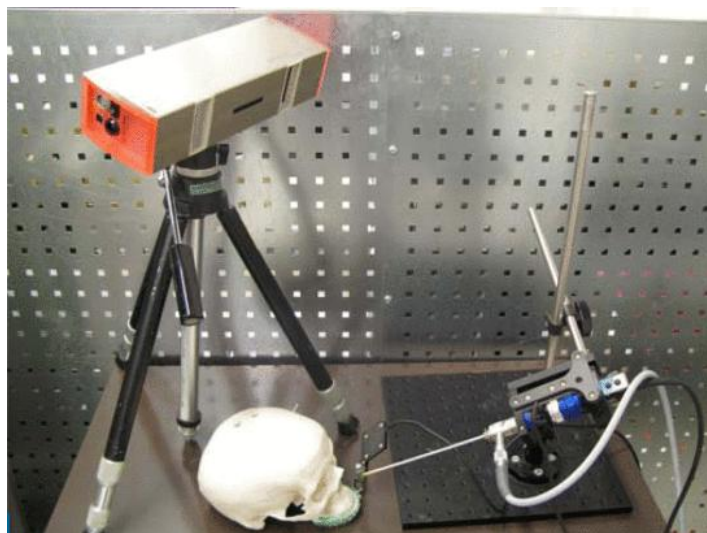
L'articolo "*Landmark-Based Indoor Positioning for Visually Impaired Individuals*"[12] affronta il tema dell'**aiuto ai disabili** visivi negli spostamenti in ambienti chiusi.

La soluzione proposta è quella di un dispositivo indossabile che scattando fotografie dell'ambiente ad una certa frequenza, riconosca i punti di riferimento analizzando l'immagine acquisita e confrontandola con la mappa pre-caricata.

Altri articoli invece propongono l'uso dei landmark per aiutare i non vedenti ad orientarsi sia dentro che fuori dagli edifici [13].

Il sistema da me sviluppato, si basa invece sul dare molte indicazioni estremamente semplici, ogni indicazione prevede un solo cambio di direzione, in tal modo, anche senza l'utilizzo di hardware specifico, un utente non vedente può, utilizzando il tatto e/o il bastone, completare il singolo passo di navigazione e completandoli tutti arrivare alla destinazione desiderata.

La navigazione per punti di riferimento viene utilizzata anche in **campo medico**. [14]  
In particolare negli interventi chirurgici al cranio eseguiti in endoscopia, l'utilizzo della landmark-based navigation, associata alla realtà aumentata, si è rivelata un valido aiuto nella precisione del movimento degli strumenti chirurgici, aiutando chi esegue l'intervento a trovare il punto da operare anche in condizioni di scarsa visibilità causati ad esempio dalla presenza di sangue o dalla differente anatomia della struttura dei nervi e delle vene tra paziente e paziente.



*Figura 3*

*Da: R. U. Thoranaghatte et al, 2008, "Landmark based Augmented Reality endoscope system for sinus and skull-base surgeries"*

Un altro interessante argomento riguarda il problema di come generare la mappa dell'interno di un edificio.

Nell'articolo *"ALIMC: Activity Landmark-Based Indoor Mapping via Crowdsourcing"* [15] si affronta l'argomento utilizzando gli utenti come elementi attivi nella costruzione e manutenzione delle mappe.

Per fare ciò, i ricercatori si affidano al crowdsourcing, cioè la collaborazione degli utenti nella raccolta dei dati.

Il sistema ALIMC utilizza i sensori degli smartphone e i segnali wifi captati, per capire sia quali attività gli utenti stanno svolgendo, sia dove la stanno svolgendo generando così una mappa in modo automatico.

A differenza di ciò che viene proposto in quest'ultimo articolo, il mio sistema utilizza mappe precostituite generate da un amministratore. Ciò, seppur risultando più lento come metodo, garantisce una maggiore accuratezza e precisione delle indicazioni che verranno date all'utente e, data la generale lentezza con la quale l'interno di un edificio cambia nel tempo, risulta preferibile rispetto alle altre soluzioni.

Ho immaginato di risolvere il problema di un utente che si trovi a dover localizzare un determinato ufficio all'interno di un edificio di più piani.

Ho immaginato anche che questa persona possa essere italiana o straniera, che abbia degli handicap fisici oppure no; in tutti questi casi l'app deve risultare facilmente utilizzabile e adattabile alle esigenze dell'utente.

Per la soluzione di questo problema ho pensato ad un sistema che, sfruttando i grafi e la ricerca del percorso più breve tra due nodi, metta l'utente nelle condizioni di trovare al primo colpo la strada ottimale tra due punti di un edificio senza perdere tempo e senza dover chiedere informazioni a nessuno.

## 2. Progettazione

A differenza di una città, che si sviluppa solitamente in orizzontale, all'interno di un edificio l'esistenza di più piani crea difficoltà nel disegno di una mappa.

Una mappa, graficamente indica il percorso da seguire, ma in presenza di spostamenti su altri livelli, può confondere l'utente.

L'approccio adottato nel mio progetto è quello di una navigazione passo-passo che, di fatto, risolve le ambiguità esistenti nella consultazione delle carte tradizionali.

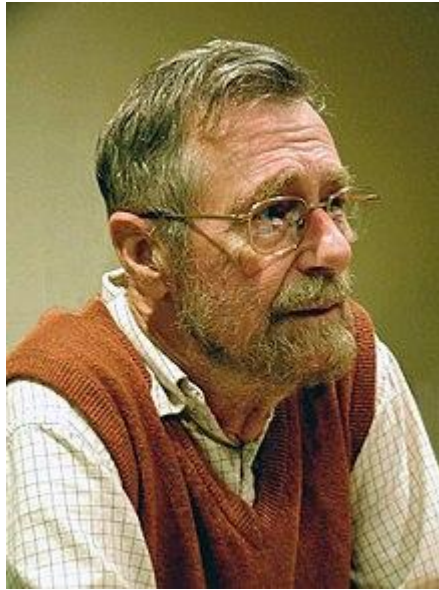
Inoltre l'utente, una volta raggiunta una prima meta, potrebbe volersi muovere da quel punto verso altri luoghi; consultando una cartina tradizionale, però, potrebbe essere difficile ritrovare addirittura il punto dal quale si sta nuovamente partendo.

Il sistema da me progettato, invece, richiede solo di indicare un punto di riferimento come ad esempio il nome o il numero scritto sulla porta più vicina a noi per dare al sistema un punto di partenza.

### 2.1. Il sistema

Il sistema consiste in un sito web responsive raggiungibile anche tramite l'apposita applicazione android.

Si basa principalmente sull'algoritmo di Dijkstra, sviluppato da Edsger Dijkstra nel 1956 allo scopo di cercare i cammini minimi in un grafo.



*Figura 4*  
*Edsger Wybe Dijkstra*

In questo caso l'algoritmo viene utilizzato per calcolare il percorso migliore per muoversi da un qualsiasi punto di un edificio ad un altro.

L'utente deve solo specificare dove si trova e dove vuole andare per ricevere come risultato una lista di semplici indicazioni di spostamento da seguire.

È inoltre possibile visualizzare il sistema in altre lingue e richiedere aiuto in caso di disabilità visiva e/o motoria.

Un edificio, o una serie di edifici vicini, viene trasformato in un grafo dove ogni stanza è un nodo ed ogni percorso da una stanza all'altra è un arco.

La conversione viene fatta semplicemente inserendo i dati dall'interfaccia utente del pannello di controllo, il quale è accessibile solo tramite password da un amministratore.

I dati che vengono richiesti per convertire due stanze collegate in due nodi connessi di un grafo:

- 1) A quale nodo esistente si collegherà il nuovo nodo
- 2) Il nome del nuovo nodo

- 3) La distanza in metri tra i due nodi
- 4) L'azione da compiere per raggiungere il nuovo nodo (ad esempio entrare in ascensore, salire le scale o semplicemente camminare)
- 5) La direzione nella quale ci si muove espressa in punti cardinali (nord, est, sud, ovest)
  - È stata fatta questa scelta per poter calcolare in modo automatico cosa dire all'utente: girare a destra, a sinistra, continuare dritto o tornare indietro
- 6) Un file di tipo immagine che ritragga la nuova stanza che si sta aggiungendo al sistema
- 7) Una descrizione testuale dell'immagine
  - Questa verrà caricata al posto dell'immagine nel caso l'utente dichiari di essere non vedente
- 8) Dichiarare se il percorso descritto per arrivare alla nuova stanza sia fruibile per utenti con disabilità motorie
  - Ad esempio se sono presenti rampe di scale, bisognerà selezionare "NO"

I dati inseriti verranno salvati su un file testuale presente sul server

Inoltre il nome della nuova stanza verrà aggiunto ad un secondo file di testo che conterrà il nome di tutte le stanze

Infine il file immagine verrà caricato su di un'apposita cartella sul server e rinominato <nome della stanza esistente><nome della nuova stanza>

### **3. Implementazione**

È stato realizzato sia un sito web responsive, sia un'app per connettersi al servizio.

Il sistema per funzionare ha bisogno di un server. Ho deciso di implementare un server locale utilizzando l'utility XAMPP che permette di creare un server apache in grado di eseguire codice PHP e python senza pagare abbonamenti.

## 3.1. Descrizione funzionamento lato utente

### 3.1.1. L'homepage

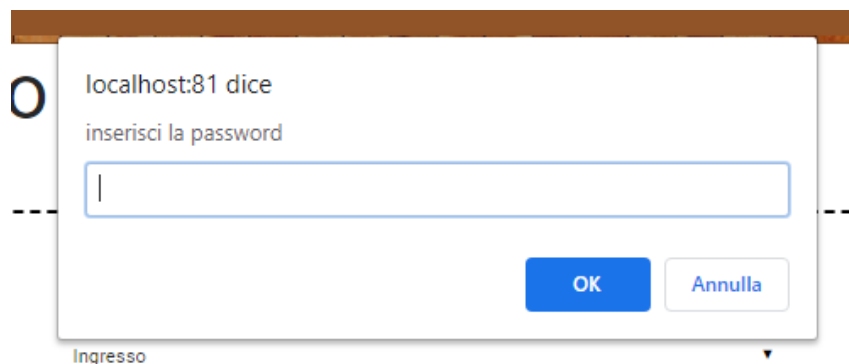
L'utente che accede alla homepage o che avvia l'app si trova di fronte ad una pagina in cui gli viene chiesto di specificare.

- 1) La lingua nella quale vuole utilizzare il sistema
- 2) In quale punto dell'edificio si trova
- 3) In quale punto dell'edificio vuole arrivare
- 4) Se ha handicap visivi o motori

The image shows a web form titled "Landmark Based Navigation". At the top right, there is a language selection dropdown labeled "Seleziona lingua" with a red arrow and the number "1" pointing to it. The main form area is enclosed in a dashed black box. Inside this box, there are two dropdown menus: "Partenza:" (departure) and "Arrivo:" (arrival), both currently set to "Ingresso". A red arrow and the number "2" point to the "Partenza:" dropdown, and another red arrow and the number "3" point to the "Arrivo:" dropdown. Below these, there is a section titled "Agevolazioni per utenti disabili:" (Accessibility for disabled users) with two checkboxes: "Difficoltà motorie" (motor difficulties) and "Difficoltà visive" (visual difficulties). A red arrow and the number "4" point to the "Difficoltà motorie" checkbox. At the bottom left of the dashed box is a teal button labeled "CERCA" with a right-pointing arrow. Below the dashed box is a teal button labeled "PANNELLO DI CONTROLLO" with a key icon.

Figura 5  
Homepage versione desktop





*Figura 6*  
*Richiesta della password*

È presente infine un bottone per accedere al pannello di controllo. Il pannello di controllo è accessibile solo dall'amministratore di sistema dotato di password.

Una volta che l'utente avrà selezionato i 4 parametri della homepage, selezionando "cerca" verrà reindirizzato alla pagina del risultato.

### **3.1.2. La pagina del risultato**

Una volta che l'utente ha completato la richiesta di indicazioni, verrà reindirizzato alla pagina del risultato.

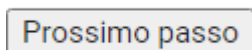
Nella pagina del risultato è descritto come arrivare al luogo desiderato passo per passo.

Ogni passo è composto da una istruzione a parole es: "cammina per 10 metri ed entra nella stanza x" e da un'immagine del punto di riferimento descritto.



*Figura 7*  
*Esempio di risultato in senza aiuti per disabilità*

È anche presente un tasto che fa scorrere automaticamente la pagina al passo successivo, si può in alternativa semplicemente scorrere la pagina verso il basso.



*Figura 8*  
*Alla pressione di questo tasto la pagina si sposta automaticamente all'istruzione successiva*

La pagina del risultato visualizzerà le istruzioni anche tenendo conto degli handicap segnalati dall'utente.

### 3.1.2.1. Aiuti ai disabili

#### 3.1.2.1.1. Aiuto in caso di disabilità visiva

L'utente che dichiara di avere una disabilità visiva non riconosce i landmark dal loro aspetto ma dalle informazioni che può trarre tramite il tatto o l'udito.

Per venire incontro a queste esigenze, la pagina del risultato in questo caso non visualizzerà le immagini dei landmark bensì una loro descrizione a parole.

L'idea è quella di permettere ai software text-to-speech attivi negli smartphone degli utenti non vedenti di leggere la descrizione del punto di riferimento.

Un utente con problemi visivi troverà d'aiuto ad esempio l'informazione che una porta deve essere fatta scorrere piuttosto che spinta, oppure che la porta che gli interessa aprire sia in legno piuttosto che in altri materiali così da poterla riconoscere al tatto.



Figura 9

*Scena tratta dal film "Il favoloso mondo di Amélie", 2002*

*In questa scena l'efficace descrizione dell'ambiente circostante fatto della protagonista al non vedente permette all'uomo di "vedere" tramite le parole.*

### 3.1.2.1.2. Aiuto in caso di disabilità motoria

L'utente che dichiara di avere difficoltà motorie ha bisogno di un percorso privo di barriere architettoniche.

In questa modalità il sistema privilegia gli archi appositamente indicati come agibili per utenti con disabilità motorie presentando all'utente il percorso più breve possibile tra quelli adatti alle sue esigenze.

L'idea è quella di ottenere un percorso che, anche se più lungo in termini di metri da percorrere, escluda scale, corridoi troppo stretti per le sedie a rotelle e ostacoli simili. Ovviamente non sempre esistono percorsi con queste caratteristiche. In questo caso sarà comunque proposto all'utente un percorso ma questo sarà preceduto da un avviso della presenza di barriere architettoniche.

Alla fine di ogni percorso, oltre alla scritta "sei arrivato a <destinazione>" viene anche calcolato il totale dei metri percorsi.

È infine presente un bottone "nuova ricerca" che reindirizza l'utente alla homepage.



*Figura 10*  
*Tasto "nuova ricerca" per tornare alla pagina principale*

### 3.1.3. Il pannello di controllo

Il sistema permette anche di **aggiungere, eliminare, rinominare o modificare** i parametri di ogni nodo del grafo dall'interfaccia grafica.

Queste opzioni sono però accessibili solo dopo l'inserimento di una password che solo l'amministratore conosce.

La password di amministratore è salvata come variabile di sessione. Questo significa che non verrà più chiesta fino a che il browser resterà aperto. Ciò rende molto più agevole il lavoro in caso di modifiche multiple.

Al pannello di controllo si accede dalla homepage tramite il tasto “pannello di controllo”.

### Pannello di controllo

← TORNA A HOMEPAGE

#### Aggiungi Nodo

A quale nodo vuoi collegarti?: Ingresso ▼

Nome nuovo nodo: Nome nuovo nodo

Distanza tra i nodi: Distanza tra i nodi

Azione da compiere: Azione da compiere

In che direzione ti muovi?: Nord ▼

Descrizione immagine: Descrivi l'immagine per gli utenti non vedenti

Agibile per utenti con difficoltà motorie?: ☒ Sì/No

Invia questo file:  Nessun file selezionato

INVIA ➤

*Figura 11*

*L'opzione “aggiungi” permette di inserire un nuovo Landmark nel sistema*

Ognuna delle 4 funzioni del pannello di controllo è un form che una volta compilato ed “inviato” andrà a modificare i file sul server come richiesto.

Tutti i menu a tendina contengono dei valori che sono stati inseriti automaticamente leggendo il file di sistema dove sono salvati i nomi di tutti i Landmark presenti nel grafo.

Naturalmente se ci si accorge di avere acceduto al pannello di controllo per errore, esiste un apposito tasto per tornare alla homepage.

### Pannello di controllo

← TORNA A HOMEPAGE

*Figura 12*  
*Pannello di controllo*

Dopo ogni modifica effettuata dal pannello di controllo, si verrà riportati al pannello di controllo all'altezza dell'ultima modifica fatta, ciò viene fatto grazie all'utilizzo di link interni (anchor link).

### 3.1.4. L'applicazione android

Ho inoltre sviluppato una semplice app per utilizzare più comodamente il sistema.

L'app è composta da un unico elemento "webview" nel quale verrà visualizzata la versione mobile del sito.

L'app permette di fare tutto quello che si può fare da PC, ma con più semplicità per gli utenti mobile.

Si tratta è una sorta di browser web dedicato al sistema. Il compito dell'app android è quindi solo quello di scaricare l'intera pagina web, completa di interfaccia grafica e di visualizzarla correttamente sullo smartphone dell'utente.

La scelta di un sistema così semplice permette di vedere applicate all'app tutte le eventuali modifiche grafiche fatte al sistema senza dover aggiornare l'applicazione.

Il costo è, però, quello di un maggiore utilizzo di dati causato dal download dell'intera interfaccia grafica.

Dall'app è possibile caricare le foto scegliendole direttamente dalla galleria del telefono.

È presente un sistema che evita di uscire dall'applicazione nel caso in cui si tocchi inavvertitamente il tasto "indietro" di android.



*Figura 13  
Screenshot della homepage vista  
dall'app*

Alla pressione del tasto indietro, viene visualizzato un messaggio “toast” che spiega all’utente che per uscire è necessario premere il tasto ancora una volta. Per uscire dall’applicazione è infatti necessario premere il tasto indietro per 2 volte in rapida sequenza.

## 3.2. Descrizione del funzionamento lato Server

### 3.2.1. La homepage

La homepage si compone di 2 select cioè menu a tendina (partenza e arrivo) e due checkbox per segnalare eventuali handicap dell’utente (handicap motori / visivi).

I menù a tendina vengono popolati automaticamente con i nomi di tutte le stanze presenti nel sistema.

Questo viene fatto grazie ad un codice PHP all’interno della pagina web che legge un file denominato *nodi.txt* contenente i nomi di tutte le stanze. Il file “nodi” viene acceduto in lettura dalla homepage ed in scrittura dal pannello di controllo quando si inserisce una nuova stanza.

Quando l’utente ha compilato la sua richiesta, alla pressione del tasto “cerca” viene lanciato il codice python *main.py*, questo codice prende in input i seguenti parametri:

*start* (luogo dove l’utente si trova)

*finish* (luogo dove l’utente vuole andare)

*moto* (avrà il valore “yes” se l’utente ha segnalato di avere problemi motori)

*vista* (avrà il valore “yes” se l’utente ha segnalato di essere non vedente)

Il codice main a sua volta procederà a richiamare i seguenti file:

*Grafo.py*: questo codice leggerà il file *grafo.txt* e con esso caricherà in memoria l’intero grafo corrispondente all’edificio.

*Dijkstra.py*: questo codice utilizzerà i parametri start, finish e moto per calcolare, tramite l’algoritmo di Dijkstra, il percorso più breve per muoversi dal punto di

partenza a quello di arrivo.

*Direzione.py*: questo file contiene il metodo “orienta” che trasforma due coordinate cardinali in un’istruzione che permetta all’utente di capire dove svoltare.

Ad esempio se ci si muove da sud ad ovest, si otterrà l’indicazione “svolta a sinistra”.

Quando il codice Dijkstra avrà restituito il vettore contenente tutti i passi da percorrere, questi saranno impaginati indicando: la svolta da prendere, l’azione da compiere, i metri da percorrere e il punto di riferimento intermedio da raggiungere, allo stesso modo le immagini verranno impaginate dinamicamente sfruttando il titolo del file (se l’istruzione indica di spostarsi da stanza1 a stanza2, verrà inserita l’immagine dal titolo “stanza1stanza2.jpg”).

Tutti i dati così ottenuti saranno scritti nel file “cont.html” il quale verrà caricato dalla pagina “risultato.html” alla quale verremo reindirizzati così da poter essere formattato correttamente grazie un file css e al framework “materialize”.

### 3.2.2. Il pannello di controllo

Il pannello di controllo si trova nel file “admin.php” e contiene:

La richiesta della password: la password, è salvata come variabile di sessione quindi sarà chiesta una sola volta fino alla chiusura del browser. Se viene inserita la password errata, oppure ci si rifiuta del tutto di inserirla, si viene reindirizzati alla pagina precedente, cioè la homepage in quanto il pannello di controllo è accessibile solo tramite l’apposito tasto presente nella home.

Di seguito sono presenti 4 form separati per le 4 funzioni del pannello: Aggiungi nodo, Elimina Nodo, Rinomina Nodo e Modifica arco.



### 3.2.2.1. Aggiungi nodo

È la funzione più complessa perché comprende anche l'upload di un file di immagine.

Dopo aver riempito tutti i campi del form, infatti, perché la richiesta sia accettabile è necessario caricare un file di tipo immagine.

Alla pressione del tasto "invia" i dati inseriti verranno inviati al file "upload\_file.php".

Il compito del codice è quello di inserire il file di immagine nella cartella "img" presente nel percorso "C:/xampp/htdocs/test/Progetto/img/" e di rinominarlo seguendo lo schema "<vecchio nodo><nuovo nodo>.jpg" questo permetterà di richiamare l'immagine concatenando i nomi di due nodi. L'immagine così rappresenterà in modo univoco l'arco che li collega.

È degno di nota un particolare bug che mi ha creato non pochi problemi durante lo sviluppo.

Se la foto che si sta caricando è stata scattata in modalità *portrait* da uno smartphone, il file di immagine risultante potrebbe essere in realtà in *landscape*, cioè con la base più larga dell'altezza ma avere un metadato che ne corregge l'orientamento quando viene aperto. Il browser google chrome, a differenza di altri non riesce ad interpretare correttamente l'orientamento della fotografia e il risultato è un'immagine ruotata di 90 o 180 gradi.

Per risolvere il problema alla radice, in fase di upload viene richiamato il metodo "correctImageOrientation" che legge il metadato EXIF del file e sovrascrive l'immagine ruotandola *effettivamente*. Il browser così non deve più leggere o interpretare alcun metadato e la visualizzazione risulta sempre corretta.

Il form aggiungi nodo, però, conteneva molti altri dati. Per semplificare il debug e la futura manutenzione del programma, i 7 parametri inseriti nel form (nodo a cui ci colleghiamo, nome nuovo nodo, distanza tra i nodi, azione da compiere, direzione, descrizione dell'immagine immagine, agibilità per utenti con difficoltà motorie) sono inviati al file *NuovoNodo.py* questo file inserisce tutti i parametri in una nuova riga del

file *grafo.txt* separati da una virgola e, dopo aver controllato che non sia già presente, inserisce solo il nome del nuovo nodo nel file *nodi.txt*.

Terminata tale operazione, così come per tutte le funzioni del pannello di controllo, si viene reindirizzati alla pagina del pannello di controllo, all'altezza dell'ultima funzione utilizzata.

### 3.2.2.2. Elimina nodo

La seconda funzione è **elimina nodo**, la sua funzione è quella di eliminare tutte le occorrenze del Landmark che si desidera eliminare.

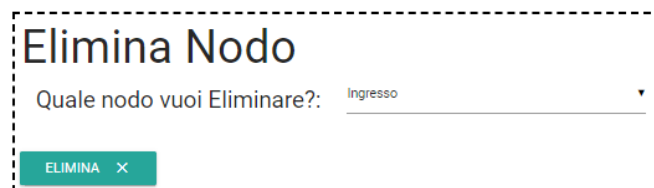


Figura 14

*L'opzione "elimina" cerca tutte le occorrenze del nodo selezionato e le cancella dai file di sistema*

Una volta selezionata la voce da eliminare e "inviato" il form, viene lanciato *CancellaNodo.py* la prima operazione che viene effettuata è quella di cercare file di immagine che nel titolo presentano il nome del Landmark da eliminare, che sia all'inizio o alla fine del file.

Successivamente viene letto il file *grafo.txt* e, se una riga contiene la parola cercata, quella riga viene eliminata per intero. Infine viene letto il file *nodi.txt* ed eliminata la riga corrispondente alla parola cercata.

### 3.2.2.3. Rinomina nodo

La funzione successiva è **rinomina nodo**, il suo funzionamento è del tutto simile a quello di elimina nodo. Lo scopo è quello di

rinominare tutte le occorrenze del nome selezionato. Come per

l'eliminazione si inizia cercando i file di immagine che contengono nel nome la parola cercata e si procede a rinominarli. Successivamente viene aperto in lettura il file *grafo.txt* modificando le linee di file contenenti il nome del landmark da rinominare. Infine viene acceduto il file *nodi.txt* e una volta trovata la riga corrispondente, la voce viene modificata come richiesto dall'utente.

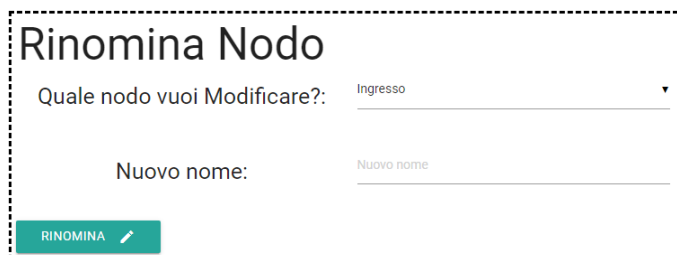


Figura 15  
Opzione "rinomina nodo"

### 3.2.2.4. Modifica arco

Infine la funzione **Modifica arco**: il menu a tendina in questo caso contiene le intere righe lette dal file *grafo.txt* per indicare all'utente quale arco si sta modificando. Viene richiesto di inserire una nuova distanza tra i due nodi, una nuova azione e una nuova direzione (a scelta solo tra nord, sud, ovest ed

est). Inviato il form, i dati vengono passati al file *ModificaNodo.py* dove accedendo in scrittura al file *grafo.txt* alla riga già indicata dall'utente, vengono rimpiazzati i parametri 2, 3 e 4 come richiesto dall'utente. Per distinguere dove finisce ogni parametro della riga, si usa il carattere “,” come separatore.



Figura 16  
Opzione "modifica arco"

### 3.2.3. La pagina del risultato

La pagina del risultato si trova all'indirizzo <http://localhost:81/test/Progetto/risultato.html>.

Il file *risultato.html* ha solo il compito di caricare e formattare correttamente il contenuto che si trova nel file *cont.html* che a sua volta è stato scritto da *main.py*

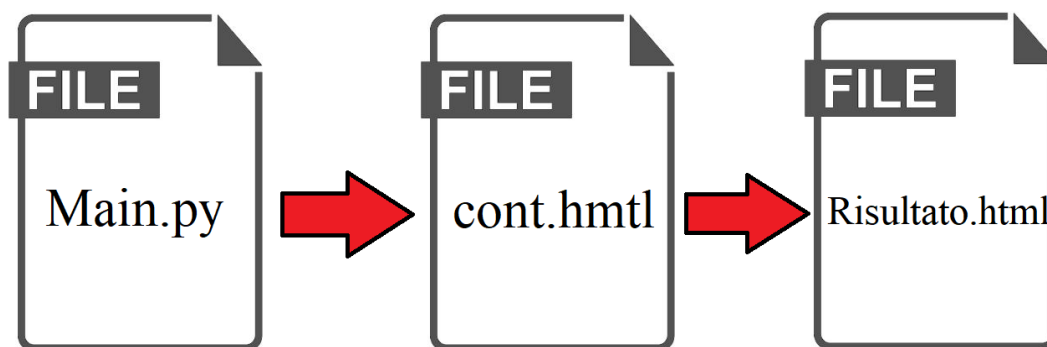


Figura 17

*Il file Main.py crea il contenuto del risultato e lo scrive nel file cont.html.*

*Il file risultato.html carica il file cont.html al suo interno e lo formatta in modo corretto utilizzando framework, css e script*

#### 3.2.3.1. In caso di disabilità visiva

Per aiutare gli utenti con disabilità visiva il sistema non visualizzerà le immagini dei landmark ma solo la loro descrizione testuale.

Per questo motivo alla creazione di un nuovo nodo, oltre ad un file di immagine è richiesta anche una descrizione. Questa sarà sempre inserita nell'attributo alt dell'elemento immagine.

L'utente che indica di avere una disabilità visiva genererà un link al risultato del tipo seguente: <http://localhost:81/test/Progetto/risultato.html?&vista=yes>

In coda è presente il valore della variabile *vista* impostato a "yes". Grazie a questo parametro, in fase di caricamento del risultato si potrà eseguire correttamente il seguente script:

```

<script>
    var url_string = window.location.href;
    var url = new URL(url_string);
    var vista = url.searchParams.get("vista");
    $(document).ready(function(){
        $('#pagina').load("cont.html");
    });
    if(vista=="yes"){
        setTimeout(function(){
            $('responsive-img').attr('src',"");
        }, 1000);
    }
</script>

```

Lo script legge il valore della variabile vista. Se il valore è yes vengono rimosse le immagini lasciando solo il parametro alt contenente la descrizione

La descrizione ora visibile sarà letta più agevolmente dai software di text-to-speech integrati ormai in tutti gli smartphone moderni ed essenziali per gli utenti con problemi di vista.

## Cammina per 2 metri fino a Porta scorrevole



descrizione:

Porta  
scorrevole  
bianca a  
vetri

sei arrivato a Porta  
scorrevole  
Distanza percorsa:

Figura 18

*Esempio di pagina del risultato in modalità "difficoltà visive"*

### **3.2.3.2. In caso di disabilità motoria**

L'utente che segnala di avere difficoltà motorie, imposta tramite la checkbox il parametro "moto" a "yes", ciò renderà tutti gli archi precedentemente segnalati come non adatti ad utenti con difficoltà motorie, più "pesanti" di 999 nel calcolo del percorso.

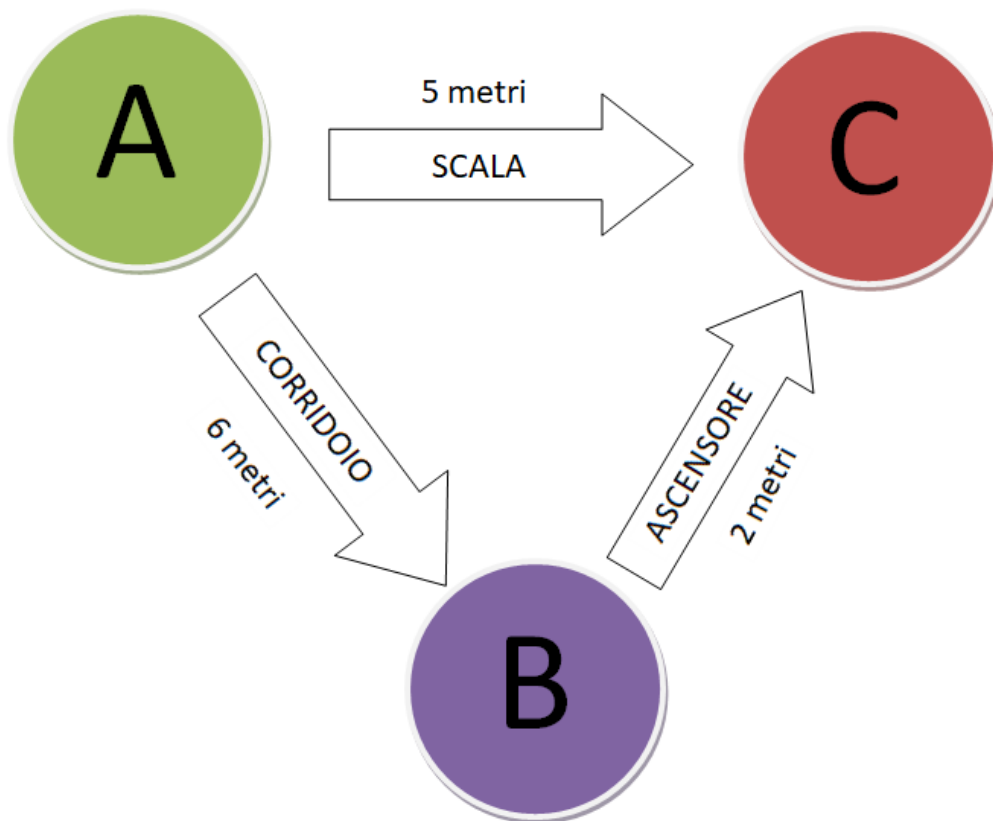
Questo privilegerà percorsi adatti ai disabili anche se più lunghi in termini di distanza, come ad esempio la scelta di arrivare fino ad un ascensore piuttosto che prendere le scale.

Nel caso in cui, per mancanza di alternative dovesse essere comunque utilizzato un passaggio non adatto ai disabili, la lunghezza in metri di quel tratto verrà visualizzata correttamente e non aumentata di 999.

Dalla homepage LMBN.php viene selezionata la checkbox. Il form contenente la richiesta viene inviata a main.py, Main.py richiama, tra gli altri, anche grafo.py ha il compito di leggere il file grafo.txt e trasformarlo in un grafo utilizzabile dall'algoritmo di Dijkstra.

Se il parametro moto è uguale a yes, in fase di caricamento ad ogni arco verrà aggiunto il valore 0 se l'arco rappresenta un passaggio adatto agli utenti con disabilità oppure 999 se il passaggio non è adatto per gli utenti suddetti.

Questa somma influisce solo durante il calcolo del percorso, in quanto in fase di presentazione del risultato, viene utilizzato il dato originale del peso dato all'arco.



*Figura 19*  
*Esempio di percorso alternativo in caso di difficoltà motorie*

Esempio: Un utente con disabilità motorie che volesse spostarsi dal punto A al punto C verrà indirizzato prima al punto B e poi da B a C per evitare la scala.

Il sistema, in modalità di assistenza per problemi motori vedrà l'arco A – C come se pesasse  $5 + 999 = 1004$  metri e sarà portato automaticamente a scegliere il percorso alternativo più breve A – B – C di 8 metri in totale.

### 3.2.3.3. La selezione della lingua

Il componente “seleziona lingua” è presente in tutte le pagine del sistema. Permette di tradurre il testo delle pagine nelle seguenti lingue: inglese, tedesco, spagnolo e francese.



*Figura 20*

Il componente strutta la funzione `googleTranslateElementInit()` fornita da google proprio a questo scopo.

```
<script type="text/javascript">

    function googleTranslateElementInit() {

        new google.translate.TranslateElement({pageLanguage: 'it',
        includedLanguages: 'it,en,de,es,fr', layout:
        google.translate.TranslateElement.InlineLayout.SIMPLE},
        'google_translate_element');

    }

</script>

<script type="text/javascript"
src="//translate.google.com/translate_a/element.js?cb=googleTranslateElementInit">
</script>
```

In fase di debug è stato necessario indicare solo le aree della pagina da NON tradurre utilizzando `<span class="nottranslate">`

Una volta selezionata la lingua, il sistema rimane tradotto su tutte le pagine.

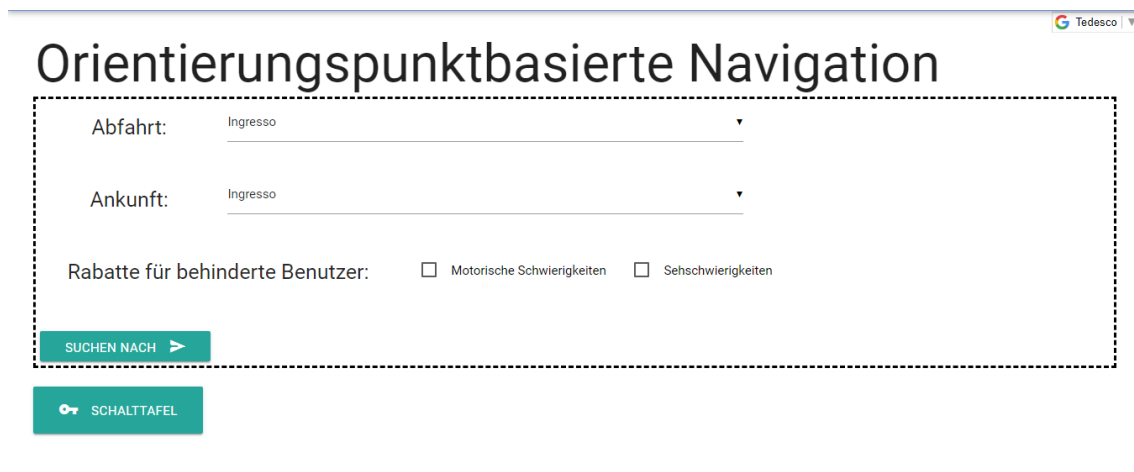


Figura 21  
*Esempio di homepage visualizzata in tedesco*



### 3.3. Lo stile

Per l'interfaccia grafica ho scelto di usare il framework "materialize" per dare uno stile *material design* agli elementi grafici come i menu a tendina, i bottoni e le checkbox.

Ho creato anche un file css per definire colori, font e posizionamento degli elementi nella pagina sia in versione desktop che mobile.

I menu select quindi eseguono una piccola animazione quando vengono estesi, così come i bottoni quando vengono cliccati e le checkbox quando vengono selezionate

Inoltre le immagini e il testo vengono scalati di dimensione in rapporto alla grandezza della pagina web o dello schermo del dispositivo su cui vengono visualizzate.

Sotto la larghezza di 400px, si entra in modalità mobile: le immagini sono centrate anziché messe a sinistra dello schermo, le tabelle nella homepage e nel pannello di controllo passano da 2 a 1 colonna e lo sfondo diventa azzurro.



Figura 22

Confronto tra pannello di controllo visualizzato in modalità desktop (a sinistra) e mobile (a destra)

## 4. Test E Validazione Sperimentale

Il progetto potrebbe essere utilizzato come mappa interattiva per matricole dell'università di Bologna. Si potrebbe caricare, ad esempio la mappa dell'interno di tutti gli edifici che compongono l'università per aiutare gli studenti a non perdersi.

L'attenzione rivolta agli utenti con disabilità lo rende utilizzabile in qualunque contesto e da un più ampio numero di utenti.

### 4.1. Pregi:

- 1) Il sistema è estremamente semplice sia da utilizzare che da mantenere, risolve un problema non indifferente che sistemi più blasonati quali google maps non hanno ancora risolto.
- 2) L'utente che usa il sistema via mobile, richiede il calcolo e il relativo download del risultato dal server in un unico momento. Questo si rivela molto utile nel caso in cui il percorso preveda zone senza segnale, eventualità frequente in edifici antichi con pareti spesse che non permettono il passaggio del segnale internet.
- 3) L'app android composta da una semplice webview non ha bisogno di essere aggiornata neanche in caso di grossi aggiornamenti grafici del sistema.

### 4.2. Difetti:

- 1) Il sistema si appoggia ad un web server locale fornito da XAMPP, per cui è accessibile solo nella rete locale in cui si trova il server.

Questo problema è però facilmente risolvibile utilizzando un server pubblico che esegua correttamente i codici python e php, nel caso di questo progetto ho scelto di utilizzare un server locale non solo perché gratuito ma anche perché, essendo un progetto sperimentale, ho ritenuto che tenerlo fuori dalla rete mondiale fosse la cosa migliore.

- 2) Per funzionare correttamente è fondamentale una perfetta conversione dell'edificio in grafo. Ad esempio un ascensore per essere rappresentato

correttamente dovrà essere convertito in più archi che collegano i piani (es: ascensore che va da 0 a 1, da 0 a 2 e da 0 a 3).

## **5. Conclusioni e sviluppi futuri**

Una feature importante da aggiungere sarebbe quella di permettere di far scegliere all'utente la mappa dell'edificio d'interesse tramite l'interfaccia utente scegliendo da una lista di mappe memorizzate.

L'utente che volesse richiedere più volte lo stesso percorso potrebbe risparmiare dati se fosse implementata una funzione di salvataggio della ricerca sulla memoria del dispositivo.

Similmente sarebbe interessante aggiungere la funzione di salvataggio dell'intera mappa di un edificio così da poter fare ricerche offline.

Per rendere il sistema appetibile per il mercato, si dovrebbe implementare un sistema più rapido di conversione in grafo. Ad esempio sfruttando le immagini raccolte da uno smartphone, permettendo così di generare la mappa semplicemente facendo camminare un addetto lungo i corridoi di un edificio.

Aggiungere una voce guida aiuterebbe gli utenti ad utilizzare il sistema anche senza guardare lo schermo del dispositivo.

Il sistema troverebbe un'utile applicazione anche all'interno di un grosso centro commerciale dove spesso il visitatore è costretto ad interpellare il personale già molto impegnato in altre attività solo per trovare un prodotto.

## **6. Conclusioni:**

In una città come Bologna, città universitaria, dalla geografia medievale, con palazzi storici adibiti ad uffici amministrativi e musei, è necessario spesso adattare le strutture preesistenti ad usi moderni. Tutto ciò, però, può creare delle difficoltà di orientamento da parte di un utente occasionale e ciò crea un danno all'immagine stessa di quell'istituzione.

Organizzare un sistema di orientamento infonde nell'utente una maggiore fiducia nelle capacità organizzative di tutto il sistema, sia esso preposto alla cultura come nel caso dell'università, alla giustizia, come nel caso dei tribunali o alle amministrazioni, come nel caso di uffici comunali, regionali o statali del territorio.

Risulta quindi particolarmente indicato un sistema di landmark based navigation, che guidi l'utente in modo semplice ma efficace all'interno di questi edifici.

Il sistema da me proposto in questa tesi mira a risolvere esattamente questo problema dando all'utente l'impressione di essere affiancato da una guida umana che gli indica la strada migliore da seguire.



*Figura 23*

## 7. Bibliografia

- [1] H. Hile *et al.*, "Landmark-based pedestrian navigation from collections of geotagged photos," in *MUM'08 - Proceedings of the 7th International Conference on Mobile and Ubiquitous Multimedia*, 2008, doi: 10.1145/1543137.1543167.
- [2] A. K. Beeharee and A. Steed, "A natural wayfinding - Exploiting photos in pedestrian navigation systems," in *ACM International Conference Proceeding Series*, 2006, doi: 10.1145/1152215.1152233.
- [3] A. Millonig and K. Schechtner, "Developing landmark-based pedestrian navigation systems," in *IEEE Conference on Intelligent Transportation Systems, Proceedings, ITSC*, 2005, doi: 10.1109/ITSC.2005.1520046.
- [4] T. H. Kolbe, "Augmented Videos and Panoramas for Pedestrian Navigation," in *Proceedings of the 2nd Symposium on Location Based Services & TeleCartography 2004, 28-29th of January 2004 in Vienna*, 2004.
- [5] J. Goodman, P. Gray, K. Khammampad, and S. Brewster, "Using landmarks to support older people in navigation," *Lect. Notes Comput. Sci. (including Subser. Lect. Notes Artif. Intell. Lect. Notes Bioinformatics)*, 2004, doi: 10.1007/978-3-540-28637-0\_4.
- [6] T. Ross, A. May, and S. Thompson, "The use of landmarks in pedestrian navigation instructions and the effects of context," *Lect. Notes Comput. Sci. (including Subser. Lect. Notes Artif. Intell. Lect. Notes Bioinformatics)*, 2004, doi: 10.1007/978-3-540-28637-0\_26.
- [7] B. Elias and V. Paelke, "3 User-Centered design of landmark visualizations," in *Lecture Notes in Geoinformation and Cartography*, 2008.
- [8] A. Furlan, T. Baldwin, and A. Klippel, "Landmark classification for route directions," 2007, doi: 10.3115/1654629.1654632.
- [9] Z. Fang, Q. Li, X. Zhang, and S. L. Shaw, "A GIS data model for landmark-based pedestrian navigation," *Int. J. Geogr. Inf. Sci.*, 2012, doi: 10.1080/13658816.2011.615749.

- [10] H. Hu and D. Gu, "Landmark-based navigation of mobile robots in manufacturing," in *IEEE Symposium on Emerging Technologies and Factory Automation, ETFA*, 1999, doi: 10.1109/etfa.1999.815346.
- [11] J. Hertzberg and F. Kirchner, "Landmark-based autonomous navigation in sewerage pipes," in *Proceedings of the 1st Euromicro Workshop on Advanced Mobile Robots, EUROBOT 1996*, 1996, doi: 10.1109/EURBOT.1996.551883.
- [12] Y. Bai, W. Jia, H. Zhang, Z. H. Mao, and M. Sun, "Landmark-based indoor positioning for visually impaired individuals," in *International Conference on Signal Processing Proceedings, ICSP*, 2014, doi: 10.1109/ICOSP.2014.7015087.
- [13] J. Balata, Z. Mikovec, P. Bures, and E. Mulickova, "Automatically generated landmark-enhanced navigation instructions for blind pedestrians," in *Proceedings of the 2016 Federated Conference on Computer Science and Information Systems, FedCSIS 2016*, 2016, doi: 10.15439/2016F135.
- [14] R. U. Thoranaghatte, J. G. Giraldez, and G. Zheng, "Landmark based Augmented Reality endoscope system for sinus and skull-base surgeries," in *Proceedings of the 30th Annual International Conference of the IEEE Engineering in Medicine and Biology Society, EMBS'08 - "Personalized Healthcare through Technology,"* 2008, doi: 10.1109/iembs.2008.4649094.
- [15] B. Zhou, Q. Li, Q. Mao, W. Tu, X. Zhang, and L. Chen, "ALIMC: Activity Landmark-Based Indoor Mapping via Crowdsourcing," *IEEE Trans. Intell. Transp. Syst.*, 2015, doi: 10.1109/TITS.2015.2423326.