

Università degli Studi di Camerino

SCUOLA DI SCIENZE E TECNOLOGIE

Corso di Laurea in Informatica (Classe L-31)



Applicazione web per l'interfacciamento ad un Real Time Location System

Laureando
Emanuele Guzzini
Matricola 093333

Relatore
Prof. Fausto Marcantoni

A.A. 2017/2018

Abstract

In un contesto sociale in cui la tecnologia 'e in grado di supportare e facilitare tutta una serie di attività quotidiane, questa deve essere il più possibile inclusiva. In questo lavoro viene sviluppato un'applicativo web in grado di interfacciarsi con un sistema di Real Time Localization per esporre, mediante una mappa ed in modo semplice e intuitivo, i dati che vengono raccolti dalle antenne e dai dispositivi. Il problema è stato affrontato mediante l'utilizzo del linguaggio html e di altri linguaggi limitrofi.

The most important property of a program is whether it accomplishes the intention of its user.

C.A.R. Hoare

Indice

Abstract	3
1 Introduzione	11
1.1 Motivazioni	11
1.2 Obiettivo	11
1.3 Struttura della Tesi	11
2 RTLS	13
2.1 Cosa è l'RTLS	13
2.2 Differenza tra RTLS e GPS	13
2.3 Standard ISO/IEC 24730-1:2014	14
3 Studio Progettuale	17
3.1 Client/Server	17
3.1.1 Cos'è un client e cos'è un server?	17
3.1.2 I fondamenti del protocollo HTTP	17
3.2 DBMS	17
3.2.1 MySQL	18
3.2.2 Alternative	18
3.3 Linguaggi utilizzati	18
3.3.1 HTML	18
3.3.2 CSS	19
3.3.3 Javascript	20
3.3.4 jQuery	21
3.3.5 Leaflet	21
4 Realizzazione del progetto	23
4.1 Front-end	23
4.2 Back-end	26
5 Conclusioni	29
5.0.1 Sviluppi futuri	29
6 Fonti	31
7 Ringraziamenti	33

Elenco dei Listati

2.1	Standard ISO IEC 24730-1:2014	14
4.1	Struttura della home	23
4.2	Struttura della mappa	24
4.3	Struttura del server	26

Elenco delle Figure

2.1	Funzionamento infrastruttura RTLS	13
3.1	Linguaggi utilizzati	18
4.1	Front-end mappa	26

1. Introduzione

1.1 Motivazioni

La tecnologia sta rapidamente entrando nella vita di tutti i giorni, amplificando gli orizzonti della realtà. Su quale sia lo scopo della tecnologia, l'idea maggiormente diffusa è quella che la vuole come supporto alle necessità dell'individuo. Sebbene la salute e la sicurezza siano considerabili necessità primarie, non bisogna fare l'errore di sottovalutare altri campi d'applicazione. Tra questi vi è l'interconnessione fra tutte le persone, uno dei metodi che vengono utilizzati di più al giorno d'oggi è il GPS (in inglese: Global Positioning System, a sua volta abbreviazione di NAVSTAR GPS, acronimo di NAVigation Satellite Timing And Ranging Global Positioning System). Questa tecnologia ha delle limitazioni, che vanno dal meteo agli ostacoli fisici. Focalizzando l'attenzione sugli impedimenti fisici, ad esempio gli edifici, ove all'interno il GPS non riesce a collegarsi ad i satelliti e rende impossibile la localizzazione. Si è pensato ad un sistema applicabile all'interno degli edifici, consultabile con qualsiasi dispositivo elettronico che supporti le tecnologie HTML ed abbia una connessione alla rete. Il portale sviluppato, consente a tutti gli utenti, di consultare la mappa dell'edificio, e localizzare tutti i dispositivi che vi sono all'interno, in maniera precisa ed in tempo reale.

1.2 Obiettivo

L'obiettivo del progetto è lo sviluppo di un applicativo web che consenta la facile consultazione dei dati ricevuti dal sistema di localizzazione. Questo portale dovrà consentire un tracciamento preciso di tutti gli oggetti che si trovano nel raggio d'azione del sistema, tutto ciò che si trova all'interno dell'edificio in questione dovrà comparire all'interno della mappa. La mappa dovrà fornire sufficienti informazioni riguardo la posizione delle ancore, degli oggetti e della conformazione dell'edificio. Dovrà permettere di rinominare i dispositivi, così da semplificarne il tracciamento, il controllo dello storico delle posizioni, così da poter ritracciare gli spostamenti di qualsiasi cosa venga tracciata.

1.3 Struttura della Tesi

Il Capitolo 1 introduce il lavoro svolto e riporta le motivazioni e l'obiettivo del progetto. Il Capitolo 2 introduce il sistema di RTLS e riporta le differenze con il GPS. Il Capitolo 3 mostra lo studio effettuato con tutte le tecnologie studiate ed utilizzate. Il Capitolo 4 spiega la vera e propria realizzazione del progetto, andando ad esplicitare la parte del front-end e del back-end nel dettaglio. Il Capitolo 5 contiene le conclusioni del progetto ed i possibili sviluppi futuri. Il Capitolo 6 contiene le fonti dove sono state prese definizioni e paragrafi che ritroveremo nella tesi. Il Capitolo 7 contiene i ringraziamenti.

2. RTLS

2.1 Cosa è l'RTLS

RTLS (Real Time Location System) sono sistemi di posizionamento precisi che consentono non solo ad un utente di identificare e monitorare diversi tipi di oggetti chiave, compresi beni, strumenti, attrezzature, contenitori, oggetti work-in-progress e persone, ma anche fornire queste informazioni in tempo reale attraverso un feedback automatico e continuo. Una soluzione RTLS offre informazioni esatte e immediate relative a dove qualcosa è o è stata - e in alcuni casi ciò che ha fatto - con costante comunicazione tra il sistema e l'oggetto tracciato. Se necessario, i dispositivi sono in grado di determinare la propria posizione e possono comunicarla, grazie ad un tag e ad un device in grado di raccogliere le informazioni che vi sono contenute. Le informazioni memorizzate su questi dispositivi, insieme con i dati di posizionamento in tempo reale dell'oggetto, sono quindi comunicati ad un business system associato, dando quella visibilità in tempo reale che supporta funzioni di pianificazione, esecuzione e rendicontazione. Idealmente, i dati sono integrati in una soluzione ERP più ampia, fornendo aggiornamenti di processi aziendali chiave come la gestione del magazzino, la pianificazione della produzione, la pianificazione dei trasporti e altre applicazioni correlate. 2.1

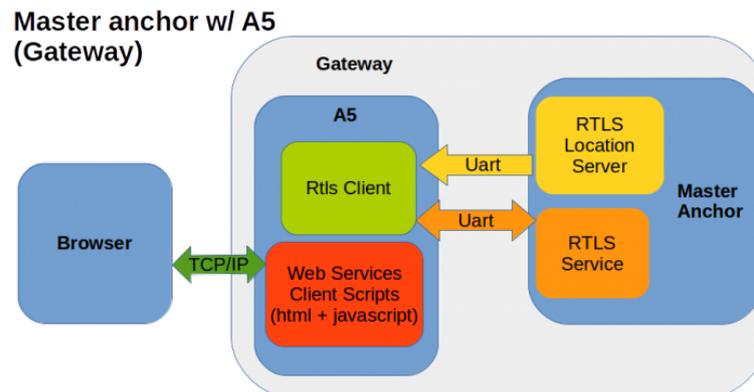


Figura 2.1: Funzionamento infrastruttura RTLS

2.2 Differenza tra RTLS e GPS

Anche se il GPS offre alcune capacità degli RTLS, i segnali GPS non possono penetrare più materiali da costruzione, lasciando strutture al coperto inaccessibili. Inoltre, i ricevitori GPS sono molto complessi, costosi e richiedono significativamente più potenza dei tag RTLS. I tag RTLS a banda ultralarga (UWB) hanno precisione superiore a 30

cm, e consumano poca energia. A seconda dell'applicazione RTLS, i tag contengono una durata della batteria che può arrivare fino a sette anni.

2.3 Standard ISOIEC 24730-1:2014

Lo standard ISO / IEC 24730-1: 2014 abilita gli applicativi software all'utilizzo di sistemi per la localizzazione consente alle applicazioni software di utilizzare un'infrastruttura di localizzazione in tempo reale (RTLS) per localizzare le risorse con trasmettitori RTLS ad esse collegate. Definisce un confine attraverso il quale il software applicativo utilizza le funzionalità dei linguaggi di programmazione per raccogliere le informazioni contenute nei lampeggiamenti dei tag RTLS ricevuti dall'infrastruttura RTLS. Il progetto, partendo dal sistema di tracciamento all'applicativo da me sviluppato, si basa interamente su questo standard, consentendo così sviluppi futuri anche da parte di terzi e consentendo sia una pubblicazione sia una possibile vendita standardizzata e sicura.

Listato 2.1: Standard ISO IEC 24730-1:2014

```
1 {
2   {...}
3 1 area of use
4 This standard enables software applications.
5 use the RTLS system infrastructure to detect objects when
6 help transmitters. This standard defines the boundary layer
7 cuts which software application uses software language facilities
8 mirovaniya to collect information contained in the Blink package tags
9
10 Create PDF in your applications with the Pdfcrowd HTML to PDF API PDFCROWD
11
12 RTLS systems received from RTLS system infrastructure.
13 2 Normative references
14 This standard uses normative references to the following
15 following standards that must be considered when using it.
16 In the case of dated references, it is only necessary to use
17 edited by For undated references, use the latest
18 edited reference documents, including any amendments and changes to
19 by him.
20 ISO / IEC 15963 Information technology. Radio Frequency ID
21 Typification for managing items. Unique radio identification
22 frequency tags (ISO / IEC 15963, Information technology – Radio frequency
23 unique identification for RF tags).
24 ISO / IEC 19762 Information technology. Automation technologies
25 identification and data collection (AISD). Harmonized dictionary
26
27 (ISO / IEC 19762, Information technology – Automatic identification and
   data cap-
28 ture (AIDC) techniques – Harmonized vocabulary).
29
30 Create PDF in your applications with the Pdfcrowd HTML to PDF API PDFCROWD
31
32 GOST R ISO / IEC 24730-1-201_
33 IEEE standard. 48-bit extended usage guide
34 unique identifier (IEEE Guidelines for use of a 48-bit Ex
35 tended Unique Identifier (EUI-48 TM)).
36 IEEE standard. Guide to using 64-bit extension
37 unique unique identifier. Registration Authorities (IEEE Guidelines for
38 64-bit Global Identifier (EUI-64 TM Registration Authority).
39 Extensible Markup Language (XML) 1.0, (Third Edition), Recommendations
40 W3C, World Wide Web Consortium (W3C), February 4, 2004 (Ex
```

41 tensible markup language (XML) 1.0, (Third Edition), W3C Recommendation,
42 World Wide Web Consortium (W3C), 4 February 2004 one)).
43 SOAP Version 1.2 Part 1: A software platform for transmitting
44 Communication (Second Edition), W3C Recommendations, World Wide Consortium
45 the web (W3C), April 27, 2007 (SOAP Version 1.2 Part 1: Messaging
46
47 Framework (Second Edition), W3C Recommendation, World Wide Web Consor-
48 tium (W3C), April 27, 2007 2)).
49
50 3 Terms and definitions
51
52 Create PDF in your applications with the Pdfcrowd [HTML to PDF API](#) PDFCROWD
53
54 In this standard, the terms are used in accordance with the
55 ISO / IEC 19762 standards, as well as the following terms with the
corresponding
56 according to the following definitions:
57 3.1 field : a data record item that stores information
58 ration, containing one or more characteristics of the blink parcel label.
59 3.2 XML tag (XML tag): a descriptor (named tag, marker),
60 It defines content in an XML document.
61 3.3 persistent connection (persistent connection): Network compounds
62 the connection between the server and the client part, which remains open
to
63 data transfer at the application level or for a request to establish
64 unity even after sending an error message at the application level
65 connections.
66 one) <http://www.w3.org/TR/REC-xml/>
67 2) <http://www.w3.org/TR/2007/REC-soap12-part1-20070427/>
68
69 GOST R ISO / IEC 24730-1-201_.
70
71 Create PDF in your applications with the Pdfcrowd [HTML to PDF API](#) PDFCROWD
72
73 3.4 membership tag (tag status): mandatory fields containing a message
74 locating, and not including the source field
75 (source) and the field "Format" (format).
76 4 Abbreviations
77 The following abbreviations are used in this standard:
78 API – an application programming interface (Application Programming
79 Interface);
80 ASCII – American Standard Code for Information Interchange
81 (American Standard Code for Information Interchange);
82 CR is an ASCII set control character denoting the operation of
83 gate of the printing head (carriage) (ASCII Carriage Return);
84
85 EUI – extended unique identifier (Extended Unique Ident-
86 tifier);
87
88 JMS is a middleware standard for
89 message links allowing platform applications
90 Java EE, create, send, receive and read messages (Java Messaging
91 Service);
92 HTTP is an application-layer data transfer protocol (HyperText
93 Transfer Protocol);
94
95 Create PDF in your applications with the Pdfcrowd [HTML to PDF API](#) PDFCROWD
96
97 HTTPS is an HTTP protocol extension that supports encryption.
98 (HTTP Secure Protocol);
99 LF – ASCII set control character denoting the place of transfer

100 line, that is, the continuation of printing text from a new line or
already on the next
101 the next page (ASCII Line Feed);
102 OUI – unique identifier of the organization (Organizationally
103 Unique Identifier);
104 REST is a method of interaction between components of a distributed
application.
105 Representation in the global Internet (Representational State Transfer);
106 RTLS system – real-time positioning system (Real
107 Time Locating System);
108
109 { ... }
110 }

3. Studio Progettuale

3.1 Client/Server

3.1.1 Cos'è un client e cos'è un server?

Approcciamoci ai server partendo dal paradigma client-server, su cui si basa la comunicazione Web. “Secondo questo paradigma, la computazione è distribuita su almeno due nodi: uno, definito client, richiede un servizio a un altro nodo, definito server, utilizzando lo stesso linguaggio comune detto protocollo. Il server risponde [...] indicando l'esito della richiesta e allegando eventuali risorse fornite come risultato.” [6]. Il protocollo usato dal paradigma client-server per il Web è HTTP. I servizi consistono principalmente in richieste di pagine Web da parte del client, che si identifica nel browser Web (il software usato per la navigazione su Internet).

3.1.2 I fondamenti del protocollo HTTP

Prima di poter parlare di Web server, bisogna affrontare un'introduzione sul protocollo utilizzato per lo scambio di dati. L'Hyper Text Transfer Protocol (HTTP) è il protocollo sul quale lavora il Web. Ogni Web browser implementa questo protocollo per scambiare informazioni con il server. HTTP definisce 9 tipi di metodi, ma quelli comunemente utilizzati sono GET e POST. GET è il metodo con cui vengono richieste la maggior parte delle informazioni ad un Web server. È possibile inviare dei parametri in query string, cioè la parte di URL dopo il punto interrogativo. Es. `sito.com/pagina?id=123&page=3`. In questo caso, i parametri sono `id` e `page` e i rispettivi valori sono 123 e 3. Il metodo POST, invece, consente di inviare i dati senza mostrarli in query string. Il protocollo HTTP è un protocollo senza stato (stateless): la connessione viene chiusa una volta terminato lo scambio richiesta/risposta e non vengono mantenuti dati di sessione o altro.

3.2 DBMS

Il database (o base di dati, abbreviato anche in DB) è un insieme di dati omogeneo per contenuti e per formato, memorizzati in un elaboratore elettronico. Risulta essere il cuore dei moderni siti Web dinamici e si identifica con il DBMS (DataBase Management System), che rappresenta il sistema software per creare, manipolare e interrogare database. Il linguaggio standard de facto per la creazione, l'interrogazione e la manipolazione nei DBMS relazionali è SQL (Structured Query Language) e vi è una vasta gamma di prodotti software che implementano tale linguaggio. Inoltre, è necessario decidere se utilizzare un DBMS open source o acquistarne uno con licenza proprietaria. Di seguito saranno riportati alcuni tra i più famosi DBMS utilizzati al mondo.

3.2.1 MySQL

È doveroso cominciare parlando di MySQL12, il “DBMS open source più utilizzato al mondo” [7]. È un DBMS di tipo relazionale (RDBMS). La peculiarità dei RDBMS è la struttura del modello di memorizzazione dei dati. Questi, nei RDBMS sono memorizzati in tabelle aventi uno o più attributi o campi dato e un valore per ciascun attributo all’interno del dominio o tipo di dato consentito. Un concetto importante è la tupla (record o riga), che rappresenta l’insieme non ordinato di valori assunti dagli attributi. Rilasciato per la prima volta nel 23 maggio 1995, è divenuto proprietà della Sun Microsystems, che a seguito è stata acquisita dalla Oracle Corporation. È supportato da numerosissimi linguaggi di programmazione, come Java, .NET, PHP e Python (alcuni dei quali saranno discussi in seguito), ed è composto da un client a riga di comando e un server. Può essere installato su sistemi operativi Unix, Unix-like e Windows. Esistono vari fork di MySQL, come MariaDB13, Drizzle14 e Percona server15, ma risultano essere meno utilizzati rispetto al “genitore”.

3.2.2 Alternative

Come già detto, la scelta del DBMS può essere effettuata tra una grande varietà di prodotti e produttori. Per completezza è necessario accennare anche alla presenza di DBMS detti “non relazionali”, ovvero che non utilizzano la struttura classica a tabelle dei database relazionali. Tra questi, spicca MongoDB20, software libero che utilizza documenti in stile JSON (JavaScript Object Notation) ed è utilizzato prevalentemente per la memorizzazione di documenti. Per altre applicazioni particolari è possibile anche ricorrere a database basati sullo storage key-value. Il database di questo tipo più utilizzato è Redis21, open source con persistenza facoltativa.

3.3 Linguaggi utilizzati

Lo sviluppo di un sito web, implica molte volte l’utilizzo di add-on e plug-in esterni, per semplificare e migliorare aspetto e prestazioni dello stesso. Prendendo in esempio il sito realizzato per questo progetto, sono stati utilizzati bootstrap per lo sviluppo della parte grafica della homepage e per la gestione dei font, javascript e jquery per tutte le parti che richiedono dinamicità, ad esempio sliders, mappe, tabelle, connessioni a oggetti e software esterni etc, leaflet, un tool per l’implementazione e la gestione di una mappa custom all’interno del sito, con tutte le funzioni annesse per il corretto funzionamento di tutto il pacchetto mappa.



Figura 3.1: Linguaggi utilizzati

3.3.1 HTML

Quando si parla di Programmazione Web, nonostante non si tratti di un linguaggio di programmazione, è bene partire dall’HTML. Come si intuisce dal nome, l’HTML

(HyperText Markup Language) è un linguaggio di markup, vale a dire un insieme di regole che descrivono i meccanismi di rappresentazione del testo. Quindi, tramite una sequenza di caratteri si marcano gli elementi di un file di testo marcatori detti ‘tag’ per assegnare loro determinate caratteristiche o funzioni. Le regole sono stabilite dal World Wide Web Consortium (W3C), associazione che definisce gli standard del World Wide Web, e la versione attuale HTML5 è stata rilasciata dal W3C nell’ottobre 2014. Tale rilascio è avvenuto a causa della necessità di fornire direttamente le funzionalità che in precedenza erano fruibili tramite estensioni proprietarie all’esterno dei browser, come Adobe Flash. Sono state apportate molte modifiche rispetto ad HTML4, in particolare:

1. l’aggiunta di regole stringenti per la strutturazione del testo in capitoli, paragrafi e sezioni;
2. l’introduzione di elementi di controllo per i menu di navigazione;
3. l’introduzione di elementi specifici per il controllo di contenuti multimediali (tag `<video>` e `<audio>`);
4. settati come deprecati o eliminati alcuni elementi che hanno dimostrato scarso o nessun utilizzo effettivo;
5. estensione a tutti i tag di una serie di attributi, specialmente quelli finalizzati all’accessibilità;
6. viene supportata l’estensione Canvas che permette di creare animazioni e grafica bitmap;
7. introduzione della geolocalizzazione, dovuta ad una forte espansione di sistemi operativi mobili più diffusi;
8. sistema alternativo ai normali cookie, chiamato Web Storage, più efficiente, che consente un notevole risparmio di banda;
9. sostituzione del lungo e complesso doctype, con un semplice `<!DOCTYPE html>`

3.3.2 CSS

Il CSS (Cascading Style Sheets) è un linguaggio di programmazione utilizzato per definire la formattazione dei documenti HTML. La nascita del CSS è diventata un’esigenza quando ci si è resi conto che i produttori di browser sviluppavano tag propri per la formattazione del codice e questo era un problema per i webmaster, poiché la resa grafica era differente utilizzando client differenti e l’assenza di una standardizzazione non permetteva di poter sviluppare lo stesso codice per due browser differenti. Con la nascita dei palmari e degli smartphone aventi basse risoluzioni e differente rapporto d’aspetto dell’immagine, non era possibile visualizzare pagine create per essere mostrate su un computer. Inoltre, si sentiva la necessità di rendere il codice HTML più leggibile. Per questa ragione, W3C alla fine del 1996 rilascia le direttive (Recommendations) per comporre il CSS e separare il contenuto dalla sua formattazione. Usando i fogli di stile è possibile controllare bordi, colori, margini, allineamenti, font e tante altre proprietà per ottenere l’effetto visivo desiderato. La pratica migliore è posizionare i fogli di stile in un file esterno all’HTML che sarà richiamato dall’HTML stesso all’avvio del file, o tramite JavaScript (discusso in seguito) al verificarsi di un evento come il click del mouse. Inserendo i fogli di stile in file separati sarà possibile riutilizzare lo stesso CSS

in più pagine. Il foglio di stile può essere inserito anche all'interno della pagina HTML, ma è necessario copiare ed incollare il codice da una pagina all'altra per riutilizzarlo. Diventa quindi difficile la sincronizzazione e l'aggiornamento. Per questa ragione, il suddetto metodo è sconsigliato, a meno che non si tratti di modifiche grafiche da applicare solo alla pagina considerata. Il terzo modo per inserire i fogli di stile all'interno dell'HTML è detto inline. L'inserimento di fogli di stile inline è un modo per inserire il linguaggio dei fogli di stile direttamente all'interno del tag di apertura di un elemento HTML. Dopo aver detto che sarebbe sempre bene separare il contenuto dalla formattazione, può risultare strano sentirsi dire che il linguaggio CSS venga integrato nel contenuto, ma a volte risulta molto semplice e veloce l'inserimento del CSS inline per piccole modifiche che non saranno utilizzate in altre pagine. Gli elementi della pagina vengono individuati tramite l'utilizzo di "Selettori" che specificano uno o più elementi dell'HTML. Esistono 3 tipi di selettori:

1. selettori universali: utilizzati per gli elementi generici dell'HTML espressi nel codice html utilizzando tag, come `<h1>` o `<input>`
2. selettori di classe: identificano gli elementi HTML aventi il nome di classe specificato all'interno dell'attributo class. A titolo d'esempio `<input class="selettore_di_classe">`
3. selettori ID: permettono di identificare in modo univoco un singolo elemento tramite una stringa testuale `<input id="selettore_ID">`

I colori all'interno della sintassi CSS possono essere espressi o in formato esadecimale o sotto forma RGB (Red Green Blue). Esiste anche un altro modo di esprimere i colori ed è utilizzando il loro nome in inglese, ma sfortunatamente questo sistema funziona soltanto con 16 colori predefiniti. Bootstrap è un insieme di elementi grafici, stilistici, di impaginazione e Javascript pronti all'uso, nati in seno a Twitter ad opera degli sviluppatori Mark Otto e Jacob Thornton. Oggi Bootstrap è un progetto indipendente ed è stato messo a disposizione degli sviluppatori di tutto il mondo che sono liberi di utilizzare questo framework come base per i propri progetti web. Questo framework ci offre i mattoni con cui costruire pagine web HTML5, completamente responsive, coerenti e funzionali. L'utilità di Bootstrap è immediatamente evidente, soprattutto nella situazione attuale in cui le pagine web possono essere fruite su una miriade di dispositivi con caratteristiche diverse. Sarà Bootstrap ad occuparsi di mettervi a disposizione elementi di stile che permettono alla pagina di adattarsi al dispositivo utilizzando, al contempo, elementi di interfaccia comuni ai siti moderni, quelli cioè che l'utente si aspetta e di cui conosce comportamento e significato.

3.3.3 Javascript

JavaScript® (spesso abbreviato con JS) è un linguaggio di programmazione, interpretato, orientato agli oggetti con first-class functions, ed è meglio conosciuto come linguaggio di scripting client-side per pagine web, nonostante questo è anche utilizzato in molte applicazioni non solamente basate sul web. Il linguaggio è prototype-based, è dinamico, imperativo, e il offre uno stile di programmazione funzionale.

JavaScript viene eseguito direttamente lato "client-side" della pagina web e può essere utilizzato per dare un design e stabilire il comportamento delle pagine web quando viene scatenato una particolare evento da parte dell'utente. JavaScript è semplice da apprendere e nello stesso tempo rappresenta un linguaggio che permette un controllo quasi totale sulla pagina web.

3.3.4 jQuery

jQuery è una libreria JavaScript per applicazioni web. Nasce con l'obiettivo di semplificare la selezione, la manipolazione, la gestione degli eventi e l'animazione di elementi DOM in pagine HTML, nonché implementare funzionalità AJAX.

Le sue caratteristiche permettono agli sviluppatori JavaScript di astrarre le interazioni a basso livello tra interazione e animazione dei contenuti delle pagine. L'approccio di tipo modulare di jQuery consente la creazione semplificata di applicazioni web e versatili contenuti dinamici. È un software libero, distribuito sotto i termini della Licenza MIT. Al 2018, jQuery risulta la libreria JavaScript più utilizzata su Internet.

3.3.5 Leaflet

Leaflet è la principale libreria JavaScript open source per le mappe interattive ottimizzate per dispositivi mobili. Con un peso di circa 38 KB di JS, ha tutte le funzionalità di mappatura di cui la maggior parte degli sviluppatori ha mai avuto bisogno. Leaflet è stato progettato con semplicità, prestazioni e usabilità in mente. Funziona in modo efficiente su tutte le principali piattaforme desktop e mobili, può essere esteso con molti plugin, ha un API funzionale, facile da usare, ben documentata e con un codice sorgente semplice e leggibile.

4. Realizzazione del progetto

Nella realizzazione del progetto si è voluto procedere con una soluzione step-by-step, andando ad affrontare una situazione alla volta. Si è partiti con lo studio della soluzione per poi passare alla fase vera e propria di sviluppo. Per affrontare questa fase sono partiti dal front-end, per poi procedere alla fase di sviluppo del database, continuando con il back-end e concludendo con l'interconnessione tra le varie parti del progetto e la rifinitura finale.

4.1 Front-end

Per front-end si intende quella parte di progetto che viene esposta all'utente finale, ovvero ciò che l'utente, digitando l'url sulla barra di ricerca, si troverà a visualizzare e con la quale dovrà interagire. Le caratteristiche principali di un front-end ben realizzato sono leggerezza, velocità e semplicità di utilizzo; nel primo caso, la leggerezza, si deve ricercare il modo meno pesante possibile per lo sviluppo, in quanto non possiamo sapere se tutti dispongono di una connessione veloce ad internet, e non possiamo sapere neanche che tipo di dispositivo verrà utilizzato per la visualizzazione, per questo è sempre sconsigliato l'utilizzo di immagini molto grandi, pagine con troppe immagini e molto grandi, perchè appesantirebbero il caricamento e renderebbero l'esperienza di navigazione meno bella e più difficile. In secondo luogo abbiamo la velocità che è strettamente legata al fattore leggerezza, in quanto un sito veloce, pronto e facilmente raggiungibile e navigabile con ogni connessione e dispositivo, è molto più fruibile e convincente, e permette l'utilizzo ad una fascia di utenti molto più ampia. In ultimo ma non per importanza la facilità di utilizzo, che deve essere a parere mio, il primo "must have" di ogni sito internet, in quanto un sito dove non si capisce dove trovare una determinata cosa, un sito dove non si riesce ad accedere in maniera semplice e veloce, non è un sito fruibile da tutti. Lo sviluppo deve essere minimalistico, seppur molto completo ed esteticamente ben rifinito, deve avere tutto "a portata di click", in quanto non è l'utente a dover cercare le sezioni, ma è il programmatore stesso a dover mettere l'utente nella posizione di avere tutto sotto mano e sotto controllo. Fatta questa premessa, andrò ora ad esplicitare il lavoro svolto nel front-end. In questa sezione, appena effettuato l'accesso al sito troveremo la pagina principale, ovvero la Home, dove l'utente potrà trovare delle informazioni utili riguardo il progetto completo, la possibilità di registrarsi al sito e poi di effettuare consequenzialmente il log-in. Questa sezione è stata sviluppata interamente sfruttando le potenzialità di bootstrap, di conseguenza stili, font e sezioni sono interamente sviluppati con questo tool.

Listato 4.1: Struttura della home

```
1 {  
2 <body>
```

```
3 <nav class="navbar navbar-expand-lg navbar-dark navbar-custom ">
4 <div class="container">
5 <a class="navbar-brand" href="#">RTLS</a>
6 <button class="navbar-toggler" type="button"
7     data-toggle="collapse" data-target="#navbarResponsive"
8     aria-controls="navbarResponsive" aria-expanded="false"
9     aria-label="Toggle navigation">
10 <span class="navbar-toggler-icon"></span>
11 </button>
12 <div class="collapse navbar-collapse" id="navbarResponsive">
13 <ul class="navbar-nav ml-auto">
14 <li class="nav-item">
15 <a class="nav-link" href="#">Sign Up</a>
16 </li>
17 <li class="nav-item">
18 <a class="nav-link" href="Map.html">Log In</a>
19 </li>
20 </ul>
21 </div>
22 </div>
23 </nav>
24
25 <header class="masthead text-center text-white">
26 <div class="masthead-content">
27 <div class="container">
28 <h1 class="masthead-heading mb-0">RealTime Location System</h1>
29 <a href="#" class="btn btn-primary btn-xl rounded-pill
30     mt-5">Learn More</a>
31 </div>
32 </div>
33 { ... }
34 }
```

Effettuando poi l'accesso al sito, verremo portati nella sezione vera e propria del sito, dove possiamo trovare una mappa sulla parte sinistra dello schermo, separata mediante uno slider, dalla tabella che contiene le informazioni riguardo i dispositivi e le ancore presenti, oltre a fornire una visione generale dell'edificio, permette il posizionamento delle ancore mediante click sulla mappa, e verranno visualizzati tutti i dispositivi in movimento, ognuno con il proprio id/mac. Nella realizzazione di questa sezione del front-end, si è utilizzato Leaflet per lo sviluppo e la gestione di tutte le possibilità della mappa, e per lo slider e tutte le altre funzionalità e oggetti si è usato il tool jqx che si avvale di jquery.

Listato 4.2: Struttura della mappa

```
1 {
2 <div id='bodymap'>
3 <div style="border: none" id='splitter '>
4 <div>
5 <div id="mapid">
6 <script type="text/javascript">
7
8 var mymap = L.map(" mapid").setView([67, 280], 2);
9 var bounds = [[0,0], [100,500]];
10 var image = L.imageOverlay('polo.jpg', bounds).addTo(mymap);
11 var mapx = 160;
12 var mapy = 55;
13 var devicemarker = L.marker([mapy, mapx], {icon:
14     DeviceIcon}).addTo(mymap);
```

```

14   let i=0;
15
16   mymap.on('click', function(ev) {
17     if (confirm(ev.latlng)) {
18       var anx = ev.latlng.lat;
19       var any = ev.latlng.lng;
20       post(anx,any);
21       var anchormarker = L.marker(ev.latlng, {icon:
           AnchorIcon}).addTo(mymap);
22     }
23
24   });
25
26   </script>
27 <script type="text/javascript">
28 function xmlParser(xml){
29   (xml).find("PushEvents").each(function()post()devicemarker.setLatLng([
           (xml).find("y").text(), 160]).update()
30
31   });
32 }
33 //<button type="button" name="start" onclick="Start();">Start</button>
34 </script>
35 </div>
36
37 <div id="consolelog">
38
39 </div>
40 </div>
41   <div>
42     <div id='jqxTabs'>
43       <ul style='margin-left: 20px;'>
44         <li>Anchors</li>
45         <li>Devices</li>
46       </ul>
47     </div>
48
49     <div id='jqxWidget' class="anchorgrid" >
50       <div id="jqxgrid" class="anchorgrid"></div>
51     </div>
52   </div>
53   <div>
54     <div id='jqxWidget' class="devicegrid" >
55       <div id="jqxgriddevice" class="devicegrid"></div>
56     </div>
57   </div>
58
59   </div>
60 </div>
61 </div>
62 </div>
63 </body>
64
65 { ... }
66 }

```

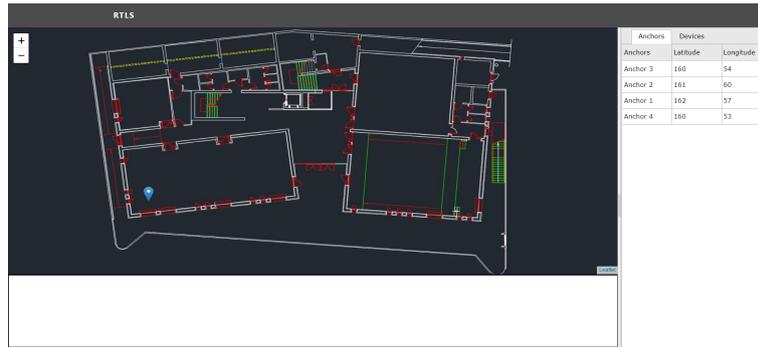


Figura 4.1: Front-end mappa

4.2 Back-end

Per back-end si intende il lato server dell'applicativo, ovvero tutto ciò che non viene visto dall'utente che visita il sito, è il cuore pulsante di tutto il progetto, ed in questo caso parliamo del database e delle funzioni di recupero ed invio dei movimenti dei dispositivi. Il Database del sistema è sviluppato in sql, contiene all'interno le tabelle relative agli utenti, quindi i dati richiesti in fase di registrazione dell'utente, ed i dati appunto di login per poter accedere al sito, nonché il livello di permessi di ogni utente, oltre a questo vi sono le tabelle contenenti i vari mac address ed id dei dispositivi, con tutte le varie posizioni e lo storico a seguire, così da poter tenere traccia degli spostamenti e poterli usare per ricostruire un ipotetico percorso eseguito dall'oggetto o dalla persona. Nel server poi, gira uno script che va a "leggere" la richiesta http con il body composto da xml che è stata inviata dall'API dell'RTLS contenente i vari dati relativi al dispositivo, comprese le coordinate attuali dei dispositivi. Ognuna di queste richieste può essere formata da uno o più tag xml che possono essere relativi a uno o più dispositivi contemporaneamente. Una volta parsato l'xml, il risultato sarà un oggetto di tipo json, che tramite delle apposite funzioni in javascript posso richiamare ed andare a prendere tutti i contenuti dei vari tag xml. A questo punto una volta prese le coordinate, si andranno ad inserire nelle funzioni di leaflet per creare il movimento del device sulla mappa del front-end.

Listato 4.3: Struttura del server

```

1 {
2 var express = require('express');
3 var mongoose = require('mongoose');
4 var bodyParser = require('body-parser');
5
6
7 var app = express();
8 app.use(bodyParser.urlencoded({
9   extended: true
10 }));
11 app.use(bodyParser.json());
12
13 // Istruisco il server su quale cartella usare come radice
14 var mainroot = _dirname + '/app/';
15 app.use(express.static(mainroot));
16
17 //app.use(app.router);
18 app.use(function(req, res) {

```

```
19     res.sendFile(__dirname + '/app/index.html');
20   });
21
22   // Inizializzo il server
23   var server = app.listen(process.env.PORT || 8080, function() {
24     var port = server.address().port;
25     console.log('Server in ascolto sulla porta', port);
26   });
27
28
29   {...}
30 }
```

5. Conclusioni

La premessa del progetto era creare un applicativo web in grado di interfacciare un software e hardware di Real Time Positioning con qualsiasi utente. Sviluppare quindi un sito in grado di rendere semplice e veloce il movimento di un qualsiasi oggetto/persona all'interno di un determinato edificio, ovvero dove il gps non riesce a funzionare e dove la precisione svolge un lavoro molto importante. L'obiettivo del progetto è stato raggiunto, fornendo una mappa dettagliata dell'edificio e la possibilità di andare a mappare gli spostamenti di questi oggetti nell'arco del tempo. Le soluzioni studiate sono state molte e si è giunti alla conclusione di utilizzare quelle descritte sopra, come leaflet,js,query e bootstrap, aggiunte alla base di html.

5.0.1 Sviluppi futuri

Come ogni progetto, anche questo è al suo stadio primordiale, la cosiddetta versione 1 del software, utilizzabile e fruibile in tutte le sue sezioni e caratteristiche, ma ancora molto migliorabile e con infinite possibili implementazioni. Al momento della stesura di questo documento, le possibili implementazioni future possono essere:

1. Storico dei movimenti in un determinato lasso di tempo;
2. Possibilità di separare le stanze degli edifici in modo da gestirle separatamente;
3. Incrementare la precisione lato software di visualizzazione con una gestione differente dei movimenti;
4. Sviluppare una sezione per gli amministratori del software.

6. Fonti

1. <https://developer.mozilla.org>
2. <https://it.wikipedia.org/>
3. <https://leafletjs.com/index.html>
4. <https://www.w3.org/html/>
5. <https://www.w3.org/Style/CSS/Overview.en.html>
6. Documento dello standard isoiec 24730-1:2014

7. Ringraziamenti

I miei ringraziamenti vanno anzitutto al Prof. Fausto Marcantoni per avermi dato la possibilità ed i mezzi per lavorare a questa tesi. Ringrazio Michele Biondi e Andrea Salvatori, i due ragazzi che hanno sviluppato l'RTLS per l'aiuto che ho ricevuto per la sincronizzazione e la spiegazione del funzionamento dello stesso. Dedico la tesi a tutti quelli che mi hanno supportato in questo percorso, in primis i miei genitori e la mia famiglia, a mio nonno che è venuto a mancare di recente e mi ha sempre spronato a dare il meglio di me, ai miei amici di tutti i giorni ed i compagni di corso, che ci sono stati sempre e mi hanno aiutato ad andare avanti in questo percorso universitario. Grazie a tutti. Emanuele