# CLIPS

Communication & Localization with Indoor Positioning Systems

---

## Università di Padova

## Manuale Utente 1.00


leaf.gruppo@gmail.com

| | |
|---|---|
| **Versione** | 1.00 |
| **Data Redazione** | 2016-05-08 |
| **Redazione** | Eduard Bicego |
| **Verifica** | |
| **Approvazione** | Davide Castello |
| **Uso** | Esterno |
| **Distribuzione** | Prof. Vardanega Tullio |
| | Prof. Cardin Riccardo |
| | Miriade S.p.A. |

# Diario delle modifiche

| Versione | Data | Autore | Ruolo | Descrizione |
|---|---|---|---|---|
| 0.01 | 2016-05-01 | Eduard Bicego | Amministratore | Aggiunta struttura documento |

# Indice

# Elenco delle figure

# 1 Introduzione

## 1.1 Panoramica generale

CLIPS è un'applicazione che offre funzionalità riguardanti la navigazione guidata all'interno di edifici, attraverso l'utilizzo di dispositivi mobile Android e dei beacon.
Tale applicazione è stata svilupparta per avere la possibilità di accedere alle informazioni per raggiungere una specifica area di interesse di un edificio offrendo indicazione testuali, sonore e visuali. Questo manuale ha lo scopo di illustrare le parti che compongono tale applicazione e il modo in tale applciazione è stata sviluppata.

# 2 Strumenti di sviluppo

## 2.1 SDK

Framework di sviluppo per applicazioni Android.

- Vesione SDK 24.4.1;
- Versione build tools 23.0.3;
- Versione target SDK 23;
- Versione minima SDK 19.

## 2.2 JDK

Insieme di strumenti di sviluppo per applicazioni Java.

- Vesione oraclejdk8.

## 2.3 Gradle

Sistema open source per l'automazione dello sviluppo basato su Groovy.

- Vesione oraclejdk8.

# 3 Componenti esterne

## 3.1 SQLite

Libreria che implementa un DBMS SQL transazionale senza la necessità di un server. Viene utilizzata per salvare e gestire le mappe scaricate e installate nel dispositivo e il relativo contenuto.

- Versione utilizzata 3.9.2

## 3.2 AltBeacon

Libreria che permette ai sistemi operativi mobile di interfacciarsi ai Beacon, offrendo molteplici funzionalità. Viene utilizzata per permettere la comunicazione tra l'applicativo Android e i Beacon.

- Versione utilizzata 2.02

## 3.3 JGraphT

Libreria Java che fornisce funzionalità matematiche per modellare grafi. Viene utilizzata per la rappresentazione delle mappe e per il calcolo dei percorsi.

- Versione utilizzata 0.9.1

## 3.4 Gson

Libreria Java che fornisce funzionalità per la gestione di oggetti JSON. Tale libreria è utilizzata la gestione del download delle mappe da remoto.

- Versione utilizzata 2.6.2

## 3.5 Dagger

Libreria Android utilizzata per effettuare la dependency injection. Viene utilizzata per la creazione dei singleton.

- Versione utilizzata 2.0

## 3.6 Picasso

Libreria per la gestione delle immagini in remoto. Viene utilizzata per scaricare le immagini utilizzate durante la navigazione.

- Versione utilizzata 2.5.2

# 4 Funzionalità

Nella presente sezione vengono spiegate nel dettaglio le componenti dell'applicazione e il loro scopo, presentate per funzionalità offerte dall'applicazione. Ogni funzionalità viene prima descritta in una sottosezione "Panoramica" e successivamente in sottosezioni che approfondiscono gli aspetti che compongono la funzionalità.

## 4.1 Localizzazione utente

### 4.1.1 Panoramica

L'applicazione offre la funzionalità di localizzare l'utente all'interno di un edifico in cui risiedono i beacon riconosciuti dall'applicazione e di mostrare semplici informazioni sull'edifico.

La localizzazione utente avviene seguendo le seguenti fasi:

1. l'utente avvia l'applicazione;
2. l'applicazione avvia il monitoring per poter rilevare i beacon circostanti;
3. l'applicazione reperisce l'identificativo major;
4. l'applicazione si accerta che i beacon rilevati siano pertinenti all'applicazione attraverso un confronto tra major rilevato e major dei beacon nel database locale;
5. se il beacon è riconosciuto ed esiste un match nel database locale:
   - viene costruito il grafico dal database;
   - vengono mostrate all'utente semplici informazioni sull'edificio;
6. se il beacon è riconosciuto e non esiste un match nel database locale:
   - viene segnalato all'utente che la mappa dell'edificio non è scaricata nel device;
   - l'utente se lo desidera è reindirizzato alla gestione delle mappe;
7. se il beacon non è riconosciuto:
   - viene ignorato.

## 4.2 Interfaccia grafica

### Componenti interne

**Package:**

**Classi:**

### Componenti esterne

**Classi SDK:**

??? Immagine dell'applicazione nella home vuota
??? Immagine dell'applicazione nella home con le informazioni dell'edificio
??? Immagini dell'applicazione nella home vuota con messaggio di avviso mappa non disponibile nel device

## 4.3 Presenter

??? Activity

### 4.3.1 Rilevamento beacon

Per il rilevamento interno

### Componenti interne

**Package:** `model`, `beacon`, `dataaccess`;

**Classi:** BeaconManagerAdapter

### Componenti esterne

**Classi JDK:** `PriorityQueue`;

**Classi SDK:** `Intent`, `LocalBroadcastManager`, `Service`, `Binder`, `LocalBroadcastManager`;

**Classi AltBeacon:** `BeaconManager`, `BootstrapNotifier`, `BeaconConsumer`, `RangeNotifier`, `Region`, `BeaconParser`, `DistanceCalculator`, `Beacon`.

La classe BeaconManagerAdapter si occupa di effettuare il ranging e il monitoring dei beacon circostanti. Essa estende un Service (vedi appendice )per cui presenta un ciclo di vita così gestito:
I principali metodi sono:

- 
- 
- 

Classe AltBeacon: ??? Classe AltBeacon: ???

## 4.4 Costruzione grafo

## 4.5 Gestione preferenze

Settings utilizzo

## 4.6 Gestione delle mappe

## 4.7 Navigazione

### 4.7.1 Panoramica

La funzionalità di navigazione è resa disponibile dalle componenti dei package:

- `navigator`;
- `beacon`;
- `compass`;
- `dataaccess`;
- `setting`.

Esse permettono di guidare l'utente all'interno di un edificio. La navigazione è gestita attraverso queste fasi:

1. l'utente interagendo con l'interfaccia grafica avvia la navigazione;
2. la business logic dell'applicazione costruisce un grafo;
3. alle componenti di `navigator` viene passato il grafo;
4. viene calcolato il percorso utilizzando la libraria `JgraphT`;
5. vengono restituite le informazioni necessarie per guidare l'utente verso la destinazione da lui scelta;
6. l'interfaccia mostra all'utente le informazioni.

### 4.7.2 Interfaccia grafica

## 4.8 Presenter

### 4.8.1 Costruzione grafo

Componenti coinvolte:

**Package:** `dataacess`, `dao`, `service`;

**Classi:** ;

**Componenti esterne:** .

La costruzione del grafo avviene MapGraph
dal database parti del grafo spiegare package graph

### 4.8.2 Calcolo percorso

### 4.8.3 Bussola

**Componenti coinvolte**

**Package:** `compass`;

**Classi:** `Compass`.

**Componenti esterne**

**Classi SDK:** `SensorManager`, `Sensor`, `SensorEventListener`.

### 4.8.4 Navigazione

**Componenti interne**

**Package:** `navigator`, `graph`, `edge`, `vertex`, `area`;

**Classi:** .

**Componenti esterne**

**Classi JGraphT:** `SimpleDirectedWeightedGraph`, `DijkstraPathFinder`, `DefaultWeightedEdge`;

**Classi JDK:** `Exception`.

## 4.9 Area sviluppatore

i

# A Fondamenti di Android

## A.1 Ciclo di vita di un'applicazione

## A.2 Activity

### A.2.1 Ciclo di vita

## A.3 Service

Tipologie di service

unbind service

## A.4 Ciclo di vita