

UNIVERSITÀ DEGLI STUDI DI PADOVA

FACOLTÀ DI INGEGNERIA

CORSO DI LAUREA TRIENNALE IN INGEGNERIA DELLE TELECOMUNICAZIONI

---

**Progettazione e Realizzazione di  
funzionalità innovative basate sul VoIP  
per audioconferenze in un sistema vocale  
interattivo**

---

*Laureando:*  
Francesco BALDAN

*Relatori:*  
Prof. Claudio NARDUZZI  
Ing. Mauro FRANCHIN

Anno Accademico 2010-2011

Questa tesi riassume i risultati ottenuti durante il lavoro di tirocinio svolto presso un'azienda. L'intero lavoro è stato impostato ed eseguito in accordo con i criteri operativi usuali presso l'azienda MIDA Solution di Padova, dalla fase di analisi delle specifiche e progettazione iniziale fino alla realizzazione finale, portando il prodotto in uno stato di prototipo funzionante molto avanzato.

L'articolazione di questa tesi descrive e sottolinea nei vari capitoli l'importanza di un'efficace pianificazione globale prima di cimentarsi nello svolgimento di una qualsiasi attività, oltre a questo vengono illustrati nel dettaglio tutti gli strumenti a mia disposizione per raggiungere l'obiettivo che l'azienda mi ha posto innanzi. Lungo tutto il periodo del tirocinio ho appreso molte nozioni ed assorbito parte di quell'esperienza che ogni componente dell'azienda ha come bagaglio professionale, ritenuta fondamentale nel mondo del lavoro.

Tutto il materiale è stato supervisionato ed eventualmente migliorato, dal riferimento aziendale Ing. M. Franchin.

## *Ringraziamenti*

Desidero ringraziare prima di tutto i miei genitori che mi hanno sempre incoraggiato e sostenuto anche nei momenti difficili.

Un forte abbraccio va alle persone che mi sono sempre state vicine, hanno sempre creduto in me e che mi hanno spinto a tirare fuori il meglio.

Un ringraziamento particolare va anche al Prof. Claudio Narduzzi, all'Ing. Mauro Franchin e a tutto il team della Mida Solution.

# Indice

|          |   |           |
|----------|---|-----------|
| <b>1</b> | <b>Introduzione</b>   | <b>4</b>  |
| 1.1      | Profilo Aziendale - Mida Solutions . . . . .  | 5         |
| <b>2</b> | <b>Descrizione generale dei mezzi utilizzati</b>  | <b>6</b>  |
| 2.1      | Ambiente di sviluppo e conoscenze preliminari . . . . .                                     | 6         |
| 2.2      | Programmi per l'ingegnerizzazione del software . . . . .                                    | 6         |
| 2.3      | Linguaggio di programmazione Python . . . . .   | 7         |
| 2.4      | Comunicazione tra Database e Python . . . . .   | 8         |
| 2.5      | Strumenti di sviluppo per applicazioni basate sul protocollo IP e Beta<br>Testing . . . . . | 9         |
| 2.6      | Alcuni cenni sul Voip . . . . .   | 9         |
| 2.7      | Componenti utilizzati e Architettura della rete . . . . .                                   | 10        |
| <b>3</b> | <b>Analisi dei requisiti, Specifiche di progetto e Progettazione</b>                        | <b>12</b> |
| 3.1      | Analisi dei requisiti e Specifiche del progetto . . . . .                                   | 12        |
| 3.1.1    | Tipologie di Conferenze . . . . .   | 13        |
| 3.1.2    | Sistema di prenotazione di un'Audioconferenza . . . . .                                     | 14        |
| 3.1.3    | Attivazione di un'Audioconferenza . . . . .   | 15        |
| 3.2      | Progettazione . . . . .   | 15        |
| 3.3      | Multithreading e Programmazione Concorrente . . . . .                                       | 16        |
| <b>4</b> | <b>Diagrammi di flusso delle principali funzionalità implementate</b>                       | <b>17</b> |
| 4.1      | Funzionamento audioconferenza generale . . . . .  | 17        |
| 4.2      | Processo di identificazione . . . . .   | 19        |
| 4.3      | Attività della conference . . . . .   | 21        |
| 4.4      | Attività di gestione degli input . . . . .  | 23        |
| 4.4.1    | Cenni sul sistema DTMF . . . . .  | 25        |
| 4.5      | Attività di gestione del tempo della conferenza . . . . .                                   | 25        |
| 4.6      | Attività di conferenza automatica . . . . .   | 25        |
| <b>5</b> | <b>Progetto UML del database</b>  | <b>29</b> |
| <b>6</b> | <b>Pagine Web di configurazione e di prenotazione per l'utente finale</b>                   | <b>33</b> |
| <b>A</b> | <b>Appendice</b>  | <b>38</b> |

# 1 Introduzione

Lo scopo del progetto che mi è stato affidato è quello di realizzare un'applicazione che implementi e gestisca le audioconferenze definite anche conferenze telefoniche, mediante l'utilizzo di strumenti quali il linguaggio di programmazione Python e un pacchetto software, che permette la realizzazione di una vasta gamma di applicazioni basate sul protocollo IP locali o remote per la telefonia.

L'audioconferenza serve per collegare contemporaneamente tra loro più utenti sparsi nel mondo grazie ad una semplice telefonata, così i diversi partecipanti di una conferenza telefonica possono comunicare tra loro direttamente con la propria voce. Il sistema consente, in pratica, di estendere le caratteristiche della comunicazione telefonica ad un numero di persone superiore a due; gli usi di un'audioconferenza possono essere ludici, ad esempio parlare tra amici o parenti, o un modo per le aziende di ridurre le spese dei viaggi di lavoro e nello stesso tempo di condividere decisioni tra più persone in un breve lasso di tempo.

Questo porta alla definizione di diverse tipologie di conferenze che variano a seconda delle necessità dell'utilizzatore finale. Proprio per soddisfare queste particolari richieste ho progettato e realizzato quattro diverse tipologie di audioconferenza, che in seguito illustrerò nel dettaglio. Parallelamente all'applicazione che effettivamente esegue la connessione alla rete e la gestione delle audioconferenze ho progettato anche le interfacce grafiche di installazione e di configurazione dell'intero sistema e le tabelle del database su cui poggia tutto il programma.

L'applicazione garantisce un'ottima portabilità, in quanto grazie all'utilizzo di Python pu` essere eseguita sia su sistemi operativi Linux sia sotto sistemi Windows.

Il lavoro di realizzazione del sistema di AudioConferenza si è svolto principalmente in tre fasi ed ha visto la collaborazione tra un'azienda padovana (Mida Solutions) e l'Università degli Studi di Padova.

In un primo momento, ho svolto una fase di acquisizione delle conoscenze generali fondamentali per capire come agivano tra loro i vari tool di sviluppo a mia disposizione, durante la seconda fase ho partecipato ad alcune riunioni con il team di sviluppo interno per analizzare dettagliatamente le richieste da soddisfare e per redigere le specifiche del programma. Nell'ultima fase si è passati alla stesura dei diagrammi di flusso, dei diagrammi UML e alla scrittura del codice vero. Nel capitolo 2 verranno illustrati i tool di sviluppo e l'ambiente Python, nel capitolo 3 è presentato invece una breve descrizione dei componenti hardware del sistema e la descrizione della rete a cui sono collegati i vari congegni. Si proseguirà poi col capitolo 4 dove verrà fornito un rapporto completo e approfondito su tutte le varie fasi del lavoro svolto presso l'azienda, nel capitolo 5 verranno commentati e spiegati alcuni dei diagrammi di flusso più

importanti che mi hanno permesso di affrontare in maniera organizzata ed efficiente la stesura completa del programma finale. Nei capitoli finali 6 e 7 si darà spazio rispettivamente all'elenco dettagliato con breve descrizione delle tabelle costituenti il Database e alle interfacce grafiche di configurazione e di prenotazione dell'utente finale. L'appendice contiene tutti i restanti diagrammi di flusso, meno significativi, per dare visibilità e completezza all'intera attività lavorativa svolta nonché all'intero software.

## **1.1 Profilo Aziendale - Mida Solutions**

Mida Solutions è un'azienda italiana che progetta, realizza e commercializza soluzioni software per Servizi Voce e Applicazioni a Valore Aggiunto, per il mondo della telefonia IP e tradizionale, al fine di dare un contributo significativo al miglioramento delle funzionalità telefoniche. L'azienda è stata fondata nel 2004, con la missione di fornire tecnologie innovative a valore aggiunto per le telecomunicazioni, puntando sia sul mercato nazionale che su quello internazionale.

Mida Solutions ha sede a Padova e che collabora strettamente con l'Università locale; il focus dell'azienda è sempre stato su tecnologie innovative e sviluppo delle risorse umane. I prodotti di Mida Solutions sono basati su tecnologie d'avanguardia nel mondo delle telecomunicazioni, come protocolli SIP e H323, Data Base ad alte prestazioni, motori di reportistica professionali, tool di RiconoscimentoVocale.

## 2 Descrizione generale dei mezzi utilizzati

Scopo di questo capitolo è dare una descrizione dettagliata di tutti gli strumenti utilizzati e delle tecniche apprese per la progettazione e l'implementazione di un'applicazione software personalizzata che viene impiegata nell'ambito delle telecomunicazioni basate sul protocollo IP.

### 2.1 Ambiente di sviluppo e conoscenze preliminari

La prima fase del tirocinio si è concentrata principalmente sull'acquisizione, l'apprendimento e la messa in pratica delle nozioni che vengono utilizzate per la realizzazione e il funzionamento di un sistema vocale interattivo; tutto questo ha permesso anche di ottenere un'ottima padronanza del sistema operativo Linux. Questi due passi iniziali sono stati fatti studiando e provando a modificare altri programmi di base già disponibili presso l'azienda svolgendo su di essi anche una piccola parte di Beta Testing (Cap 2.5).

### 2.2 Programmi per l'ingegnerizzazione del software

L'ingegnerizzazione del software è stata eseguita mediante l'utilizzo del programma Microsoft Office Visio il quale permette, attraverso una grafica semplice e lineare composta da vettori e altre strutture grafiche, la stesura oltre che di schemi UML rappresentanti le tabelle del database, di diagrammi di flusso che delineano i passi dell'algoritmo che poi dovrà essere implementato nel linguaggio di programmazione scelto, nel mio caso Python. Questa parte definisce quindi un insieme di processi, ovvero sequenze di fasi che individuano tappe specifiche nella realizzazione di un sistema software tutte documentate e ispezionabili, che offrano in sostanza perfetta visibilità alla diversa tipologia degli utenti del sistema, per il controllo dei singoli prodotti e/o per l'eventuale manutenzione. Tutta la documentazione creata verrà commentata nei capitoli 5, 6 e 7. Il tool Visio si avvale di elementi grafici che aiutano a definire in modo immediato le funzioni. Gli elementi grafici utilizzati sono molti: di seguito verranno elencati quelli principali dandone una breve descrizione allo scopo di facilitare la comprensione dei diagrammi presentati nel successivo cap.5;

- ⊗ • : questi sono i simboli rispettivamente di inizio e fine di un diagramma di flusso;
-  : all'interno del rettangolo viene specificata l'azione che viene compiuta dal processo;

-  : questo simbolo è definito come blocco decisionale, nel suo interno viene scritta una domanda e in base alla risposta si prende un percorso diverso;
-  : questo simbolo definisce una Attività ed implica, nel momento in cui si arriva su di esso, la creazione di un altro diagramma di flusso che viene richiamato dal nome scritto al suo interno(es.Attività di identificazione nella fig1 di pagina);
-  :all'interno di questo blocco viene inserito il testo o il nome del messaggio, nel nostro caso audio, che verrà riprodotto quando il processo arriva nel punto in cui è situato;
-  :questo blocco rappresenta un ritardo che viene introdotto per permettere all'utente di inserire il comando che vuole eseguire;

Il prodotto finale di questo processo di formalizzazione è costituito da un insieme di schede che documentano i processi che sono stati definiti.

## 2.3 Linguaggio di programmazione Python

L'ambiente di sviluppo utilizzato è Python, un linguaggio di programmazione ad alto livello, orientato agli oggetti, adatto, tra gli altri usi, per sviluppare applicazioni distribuite, scripting, computazione numerica e system testing. Python è un linguaggio multi-paradigma, che fa della dinamicità, semplicità e flessibilità i suoi principali obiettivi. Supporta il paradigma object oriented, la programmazione strutturata e molte caratteristiche di programmazione funzionale e riflessione. Le caratteristiche più immediatamente riconoscibili di Python sono le variabili non tipizzate e l'uso dell'indentazione per la definizione dei blocchi. Altre caratteristiche distintive sono l'overloading di operatori e funzioni tramite delegation, la presenza di un ricco assortimento di tipi e funzioni di base e librerie standard, sintassi avanzate quali slicing e list comprehension.

Il controllo dei tipi è comunque forte (strong typing) e viene eseguito al runtime (dynamic typing). In altre parole una variabile è un contenitore al quale viene associata un'etichetta (il nome) che però, durante il suo tempo di vita, può essere associata a diversi contenitori anche di tipo diverso. Il linguaggio usa un garbage collector per la liberazione automatica della memoria. Python ha qualche similarità con Perl, ma i suoi progettisti hanno scelto la via di una sintassi più essenziale e uniforme, con l'obiettivo di aumentare la leggibilità del codice. Come Perl spesso è classificato come linguaggio di scripting, ma pur essendo utile per scrivere script di sistema, la

grande quantità di librerie disponibili e la facilità con cui questo linguaggio permette di scrivere software modulare favoriscono anche lo sviluppo di applicazioni molto complesse. Python, deriva il suo sistema di indentazione dal meno noto linguaggio di programmazione Occam: invece di usare parentesi o parole chiave, usa l'indentazione stessa per indicare i blocchi nidificati.

Come detto sopra, un altro punto di forza di Python è la disponibilità di elementi che facilitano la programmazione funzionale. Le funzioni sono considerate degli oggetti e sono dunque utilizzabili alla stregua di qualsiasi altro oggetto, ad esempio inserendole in collezioni, o utilizzandole direttamente come parametri per altre funzioni. Gli elementi di programmazione funzionale, insieme a costrutti specifici per la manipolazione di contenitori, rendono ancora più comodo operare con liste o altri tipi contenitore.

Python supporta e usa estensivamente la gestione delle eccezioni come mezzo per segnalare e controllare eventuali condizioni di errore, incluse le eccezioni generate dagli errori di sintassi. Python viene in genere considerato un linguaggio interpretato. In realtà però il codice sorgente non viene direttamente convertito in linguaggio macchina, ma passa prima da una fase di pre-compilazione in bytecode; lo stesso bytecode viene quasi sempre riutilizzato dopo la prima esecuzione del programma, evitando così di dover ogni volta interpretare il sorgente ed incrementando di conseguenza le prestazioni. Inoltre è possibile distribuire programmi Python direttamente in bytecode, saltando a piè pari la fase di interpretazione da parte dell'utilizzatore finale, e permettendo anche di avere programmi Python a sorgente chiuso.

## 2.4 Comunicazione tra Database e Python

L'applicazione sviluppata richiede l'accesso ad un database in cui sono contenuti tutti i dati fondamentali che caratterizzano le varie conferenze presenti nel sistema, attive e non. Per ottenere i dati sono state usate API specifiche di JAVA delle Java DataBase Connectivity(JDBC). Queste consentono l'accesso alle basi di dati da qualsiasi programma scritto con il linguaggio di programmazione Java, indipendentemente dal tipo di DataBase Management System utilizzato. Esse sono raggruppate nel package java.sql, servono ai client per connettersi a un database e forniscono tutti i metodi per interrogare e modificare i dati in esso presenti. Il software scritto in Python per poter utilizzare le suddette API deve poterle incorporare: questo viene permesso dall'utilizzo del programma JPyte, dopo aver fatto l'import del pacchetto precedentemente nominato, esse vengono richiamate all'interno del programma con la classica sintassi dei metodi. Le prestazioni restano pressochè invariate nonostante l'invocazione necessaria per attivare la Java Virtual Machine(JVM). Un semplice

programma per vedere come usare JPype è riportato qui sotto:

```
from jpye import *
startJVM( d:/tools/j2sdk/jre/bin/client/jvm.dll ; -ea ) lancio della JVM
java.lang.System.out.println( hello world ) stampa a video il messaggio
shutdownJVM() chiusura della JVM
```

## 2.5 Strumenti di sviluppo per applicazioni basate sul protocollo IP e Beta Testing

Gran parte del software in questione è stato prodotto utilizzando API ad alto livello, fornite direttamente dalla casa madre dei componenti hardware e software che vengono impiegati per l'implementazione dell'intero sistema vocale interattivo. Queste sono un'insieme di procedure disponibili al programmatore, di solito raggruppate a formare un set di strumenti specifici per un determinato compito. L'uso delle API permette di ottenere un'astrazione, di solito tra l'hardware e il programmatore, o tra software a basso ed alto livello.

L'azienda presso la quale ho svolto il tirocinio svolge anche una rigorosa attività di Beta Testing, che è una fase di prova e collaudo di un software non ancora pubblicato, con lo scopo di trovare eventuali errori (bug). Durante la fase iniziale di apprendimento ho svolto in prima persona questo lavoro di beta-tester cercando errori presenti nelle API, comunicandoli prontamente ai responsabili in Mida e, successivamente, all'azienda fornitrice.

## 2.6 Alcuni cenni sul Voip

Nelle telecomunicazioni con il termine Voice over IP (Voce tramite protocollo Internet), acronimo VoIP, si intende una tecnologia che rende possibile effettuare una conversazione telefonica sfruttando una connessione Internet o un'altra rete dedicata che utilizza il protocollo IP senza connessione per il trasporto dati. Più specificamente con VoIP si intende l'insieme dei protocolli di comunicazione di strato applicativo che rendono possibile tale tipo di comunicazione. Grazie a numerosi provider VoIP è possibile effettuare telefonate anche verso la rete telefonica tradizionale (PSTN). Il vantaggio principale di questa tecnologia sta nel fatto che essa elimina l'obbligo di riservare della banda per ogni telefonata (commutazione di circuito), sfruttando l'allocazione dinamica delle risorse, caratteristica dei protocolli IP (commutazione di pacchetto). Mediante un'applicazione vengono instradati sulla rete pacchetti di dati

contenenti le informazioni vocali, codificati in forma digitale, e ciò solo nel momento in cui è necessario, cioè quando uno degli utenti collegati sta parlando. Le conversazioni VoIP non devono necessariamente viaggiare su Internet, ma possono anche usare come mezzo trasmissivo una qualsiasi rete privata basata sul protocollo IP, per esempio una LAN all'interno di un edificio o di un gruppo di edifici. I protocolli usati per codificare e trasmettere le conversazioni VoIP sono solitamente denominati Voice over IP protocols. Uno dei vantaggi di questa tecnologia è che permette di fare leva su risorse di rete preesistenti, consentendo una notevole riduzione dei costi in ambito sia privato che aziendale, specialmente per quanto riguarda le spese di comunicazione interaziendali e tra sedi diverse. Una rete aziendale, infatti, può essere sfruttata anche per le comunicazioni vocali, permettendo di semplificare l'installazione e il supporto e di aumentare il grado di integrazione di uffici dislocati sul territorio, ma collegati tramite l'infrastruttura di rete. La tecnologia VoIP richiede due tipologie di protocolli di comunicazione in parallelo, una per il trasporto dei dati (pacchetti voce su IP), ed una per la segnalazione della conversazione (ricostruzione del frame audio, sincronizzazione, identificazione del chiamante, etc). Nella grande maggioranza delle implementazioni VoIP, per il trasporto dei dati, viene adottato il protocollo RTP (Real-time Transport Protocol). Invece per la seconda tipologia di protocolli necessari alla telefonia via Internet, il processo di standardizzazione non si è ancora concluso. Al momento, sono coinvolti tre enti internazionali di standardizzazione: ITU (International Telecommunications Union), IETF (Internet Engineering Task Force) ed ETSI (European Telecommunication Standard Institute) con alcuni consorzi (per esempio, Softswitch, H.323ORG, Vivida ecc.). La gestione delle chiamate voce sulla rete IP, al momento, è indirizzata verso due differenti proposte, elaborate in ambito ITU e IETF, che sono rispettivamente H.323 e SIP (Session Initiation Protocol).

## 2.7 Componenti utilizzati e Architettura della rete

In questo paragrafo vengono brevemente discussi l'architettura della rete ed i componenti informatici ed elettronici a cui vengono connessi i vari sistemi. Uno schema semplificato della rete è rappresentato nella figura 2.1.

Come illustrato dalla Fig.2.1 i componenti che caratterizzano la rete sono un server, sul quale viene installata una scheda di voce virtuale ovvero che permette la comunicazione VoIP con il sistema, chiamata Prosody S. All'interno dello stessa macchina dev'essere presente anche un altro programma definito AMS Server il quale, attraverso le API messe a disposizione, permette di interagire con la scheda Prosody eseguendo varie applicazioni telefoniche VoIP o comunque basate sul protocollo IP.

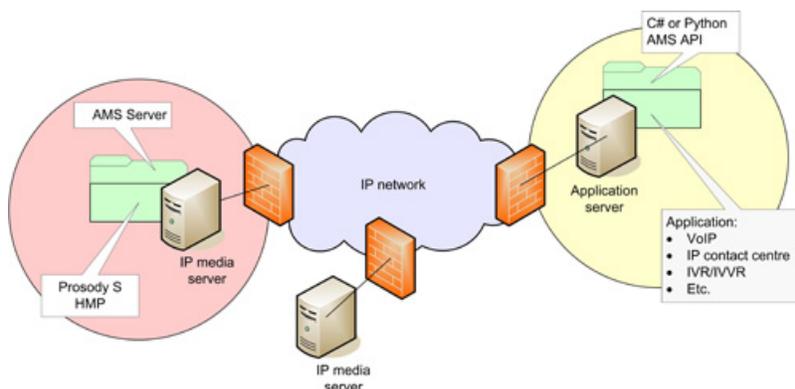


Figura 2.1: Componenti e Architettura

La principale peculiarità del sistema utilizzato è quello di avere un'unico server in cui è presente la scheda vocale virtuale e di avere in altri server remoti, ovviamente anch'essi collegati alla rete, applicazioni che sfruttano la Prosody e l'AMS Server. In pratica applicazioni multiple remote possono ottenere risorse messe a disposizione da più Prosody o AMS Server, probabilmente in esecuzione su altrettanti computer anche distanti tra loro. I vantaggi per le aziende sono chiari, poichè invece di acquistare un server costoso e poco flessibile è possibile produrne uno su misura rispondendo alle proprie esigenze portando ad un notevole risparmio di risorse economiche che possono essere indirizzate altrove all'interno dell'azienda. È proprio questo l'obiettivo posto durante il mio periodo di tirocinio.

## 3 Analisi dei requisiti, Specifiche di progetto e Progettazione

Nel seguente capitolo vengono descritte in maniera dettagliata ed esaustiva tutte le fasi di progettazione e realizzazione del software, l'analisi dei requisiti fondamentali che il sistema deve possedere e le specifiche caratterizzanti l'intero piano di lavoro.

### 3.1 Analisi dei requisiti e Specifiche del progetto

Il software agisce come un voice pager, l'amministratore può definire uno o più numeri di servizio, che possono essere anche di una Private Branch Exchange(PBX), i quali corrispondono a differenti stanze per audioconferenze: Il termine stanza o room indica l'astrazione di una specifica audioconferenza nella quale accedono solo i partecipanti invitati ad essa. I requisiti sono forniti principalmente dalle necessità che ha l'utente; queste sono il punto di partenza dal quale si è sviluppato l'intero progetto, arrivando alla definizione rigorosa delle specifiche di progetto. In particolare, si sono individuate le diverse modalità di audioconferenza da realizzare e le caratteristiche delle stanze virtuali. Ogni stanza perciò dovrà possedere:

- Una lista di riferimenti telefonici che devono essere chiamati in caso di attivazione(Active DNIS);
- Un codice PIN univoco usato per attivare la chiamata di gruppo e utilizzato anche per identificare l'amministratore;
- Messaggi pre-registrati che devono essere riprodotti quando l'utente risponde alla chiamata o esegue dei comandi;
- Una lista di comandi che permettere all'amministratore di gestire la conferenza;
- Possibilità di registrare per motivi legali o comunque utili all'interno dell'azienda l'intera audioconferenza;
- A seconda del tipo di conferenza dovrà prevedere o meno la possibilità di riservare delle risorse;

Per ogni utente è riservato e verrà utilizzato un canale voce quando questo deciderà di chiamare il servizio e di conseguenza utilizzare il sistema. Il numero dei partecipanti, chiamante incluso, di un'audioconferenza non può essere superiore al numero di canali voce disponibili nel server, bisogna quindi tenere conto delle risorse disponibili all'atto della prenotazione di una nuova conferenza, per evitare accavallamenti

tra più conferenze. A seconda della modalità selezionata i canali voce permetteranno una comunicazione full-duplex simultaneo, cioè viene consentito a tutti gli utenti connessi di parlare e ascoltare contemporaneamente, o in alternativa una comunicazione full-duplex simultaneo solo per l'amministratore della chiamata di gruppo, mentre al resto degli utenti viene consentito il solo ascolto. Una variante di quest'ultima modalità prevede che i partecipanti ospiti<sup>1</sup> possano intervenire solamente dopo aver inoltrato una richiesta all'amministratore e aver ricevuto da esso il permesso per interagire.

### 3.1.1 Tipologie di Conferenze

Il programma implementa quattro diverse tipologie di audioconferenza ognuna con diverse e innovative funzionalità:

1. **Conference Bridge**: detta anche conferenza passiva con prenotazione. A tutti i possibili partecipanti deve essere comunicato il PIN d'accesso, la data e l'ora in cui è stata fissata la conferenza telefonica. Tutti i partecipanti sono abilitati a parlare e ad ascoltare contemporaneamente, per accedere alla conferenza ogni singolo partecipante deve chiamare obbligatoriamente il numero telefonico a cui è abbinato il servizio, in caso di caduta improvvisa della telefonata qualsiasi utente può rientrare semplicemente ricomponendo il numero del servizio e identificandosi nuovamente.
2. **VoiceCast<sup>2</sup>**: detta anche conferenza attiva. Questa seconda modalità di funzionamento non ha bisogno di prenotare le risorse, utilizza i canali liberi di cui ha bisogno al momento della telefonata per attivare il servizio. Quando viene attivata questa modalità il programma mediante l'AMS Server fa partire le telefonate verso tutti i numeri inseriti in un'apposita lista, precedentemente configurata; se il chiamato risponde lo si connette alla conferenza altrimenti dopo un numero preciso di tentativi viene escluso automaticamente. Il chiamante ha la possibilità di parlare, ascoltare e utilizzare i comandi di gestione della conferenza, invece gli altri partecipanti vengono aggiunti solamente come ascoltatori. Una variante può essere abilitata nel caso in cui sia selezionata la modalità Push-to-Talk(PTT) in cui i partecipanti, escluso il chiamante, possono richiedere uno per volta la parola, digitando una sequenza numerica predefinita sulla tastiera del proprio telefono. Si invia così un segnale audio all'amministratore che può accettare la richiesta oppure scartarla. Nel caso

---

<sup>1</sup>Ospiti: sono tutti i partecipanti che non hanno i diritti di amministratore sulla conferenza

<sup>2</sup>spesso utilizzato come interfono negli ospedali

di più richieste queste verranno accodate in una lista e smaltite una per volta. L'utente che ha preso la parola può essere interrotto in qualsiasi momento dall'amministratore oppure qualora avesse finito il discorso può, digitando un preciso numero sulla sua tastiera telefonica, autoescludersi e rimettersi solamente in ascolto. In questa modalità se la linea dell'amministratore si dovesse interrompere improvvisamente la conferenza verrebbe automaticamente chiusa al contrario se fosse la linea di un qualsiasi partecipante a interrompersi solo lui perderebbe la comunicazione.

3. **MeetMe:** anche questa rientra nella categoria delle conferenze attive con prenotazione, infatti tutti partecipanti tranne il chiamante vengono contattati automaticamente senza il bisogno di telefonare ad un determinato numero. Questo attiva delle chiamate in parallelo verso i numeri di telefono che sono stati inseriti in una lista, contenuta in una tabella del database, fornita al momento della configurazione del software. Qui sia il chiamante che i chiamati hanno tutti la possibilità di ascoltare e parlare contemporaneamente, di conseguenza questa modalità risulta essere un mix tra le due precedenti cioè tra la *Conference Bridge* e il *VoiceCast*. Nello specifico se cade la linea del chiamante viene chiusa la conferenza.
4. **Meet Together:** definita anche *Automatic Conference*(AConference), quest'ultima ma non meno importante funzionalità ricade nelle conferenze attive con prenotazione, in questo caso però nessuno deve chiamare il numero del servizio per attivarlo ma ci penserà automaticamente il programma a chiamare tutti i partecipanti nella data e nell'ora definite durante la prenotazione. Anche qui a tutti i partecipanti vengono assegnati canali full-duplex simultanei, così ognuno può esprimere la propria opinione e ascoltare quelle degli altri nello stesso tempo.

### 3.1.2 Sistema di prenotazione di un'Audioconferenza

Il sistema di prenotazione di una qualsiasi conferenza telefonica si basa su un database che contiene nei vari record i parametri distintivi di ciascuna conferenza come il PIN dell'amministratore, il PIN dei partecipanti, la data e l'ora in cui si terrà la conferenza, il numero complessivo dei partecipanti, il numero su cui verrà attivato il servizio(DNIS), la durata totale della conferenza e molti altri ancora. La gestione complessiva di questo sistema non è banale in quanto deve tener conto e aggiornare costantemente le risorse disponibili nel sistema. Nel caso in cui i canali voce non siano presenti in numero sufficiente perchè già riservati, comparirà sul video un mes-

saggio d'errore. Un'altro problema non di poco conto è che l'accesso al database può essere fatto simultaneamente da più processi attivi contemporaneamente sul server, questo può portare ad errori rilevanti nella gestione complessiva delle risorse e nel caso peggiore anche ad una caduta della conferenza. Questo aspetto è stato tenuto in conto e risolto grazie ai vari strumenti presenti in qualsiasi linguaggio di programmazione multi-processo; il tema viene affrontato nel capitolo 4.3. Nel capitolo 7.2 verrà mostrata e commentata la pagina di prenotazione che l'utente finale utilizza.

### 3.1.3 Attivazione di un'Audioconferenza

A seconda della tipologia di conferenza selezionata l'attivazione può avvenire in diversi modi. Nel primo caso, ossia in una *Conference Bridge*, il primo utente che chiama il numero di servizio preimpostato attiva la conferenza e viene semplicemente aggiunto nella room restando in attesa dell'arrivo degli altri partecipanti; nel caso del *VoiceCast* o del *MeetMe* il chiamante, che sarà anche l'amministratore, fa iniziare la conferenza dopo aver inserito il proprio PIN identificativo. Quando tutti gli utenti inseriti nella lista dei membri partecipanti alla conferenza hanno risposto oppure quando il sistema ha finito i tentativi di chiamata, il programma informerà il chiamante sullo stato della conferenza, cioè su un totale di M partecipanti totali quanti effettivamente sono collegati. Nel caso di una conferenza di tipo *Meet Together* l'attivazione avviene automaticamente, il programma infatti mediante un algoritmo di gestione del tempo riesce a capire qual'è il giorno e l'ora in cui attivare la conferenza e di conseguenza ad agire chiamando tutti i numeri contenuti nella lista dei partecipanti.

## 3.2 Progettazione

La terza parte più importante, dopo aver chiarito come applicare nel migliore dei modi la tecnologia a disposizione e che cosa viene richiesto, è la fase di progettazione. Questa passo fondamentale nel processo di realizzazione del sistema, se fatto in maniera precisa, può agevolare in seguito la scrittura del codice e velocizzare l'intero processo, risparmiando così tempo che potrà essere dedicato alla fase finale di testing. Come riportato in precedenza nel capitolo 2.2 è stato utilizzato il programma Microsoft Visio per il disegno dei diagrammi di flusso che rappresentano passo dopo passo quello che deve essere eseguito dal software. Grazie a questi progetti riusciamo ad avere un'idea completa sui metodi che devono essere sviluppati, oltre ad individuare possibili instabilità e parti di codice inutili o inefficienti. Le schede prodotte sono state revisionate alcune volte prima di arrivare all'approvazione definitiva. I diagrammi

più rappresentativi verranno esposti e commentati approfonditamente nel capitolo 6, gli altri invece verranno resi disponibili in appendice.

### **3.3 Multithreading e Programmazione Concorrente**

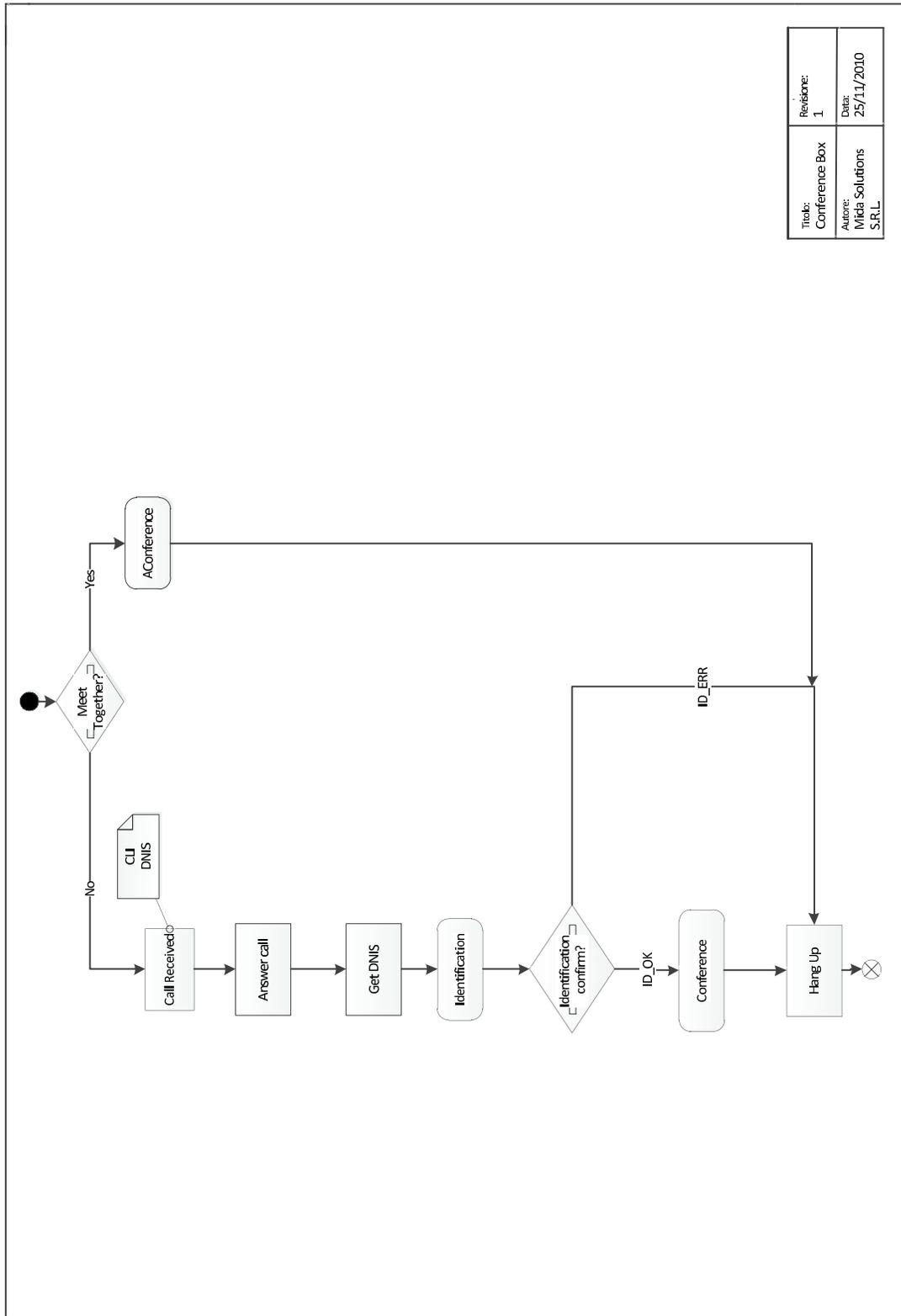
Il multithreading indica la capacità da parte di un processore di eseguire pi thread, con il termine thread s'intende la suddivisione di un processo in due o pi filoni, che vengono eseguiti concorrentemente. Questa tecnica si distingue da quella alla base dei sistemi multiprocessore per il fatto che i singoli thread condividono lo stesso spazio d'indirizzamento, la stessa cache e lo stesso translation lookaside buffer. Il multithreading migliora le prestazioni dei programmi solamente quando questi sono stati sviluppati suddividendo il carico di lavoro su più thread che possono essere eseguiti in parallelo. I sistemi multiprocessore sono dotati di più unità di calcolo indipendenti, un sistema multithread invece è dotato di una singola unità di calcolo che si cerca di utilizzare al meglio eseguendo più thread nella stessa unità di calcolo. Le tecniche sono complementari, a volte i sistemi multiprocessore implementano anche il multithreading per migliorare le prestazioni complessive del sistema. Utilizzando la tecnica del multithreading può presentarsi il problema della concorrenza, cioè l'esecuzione parallela può condurre a interazione tra processi quando viene coinvolta una risorsa condivisa. Nel mio caso la risorsa condivisa era un determinato array di oggetti, contenente i dati di una tabella del database, sul quale venivano modificate e aggiornate le situazioni delle risorse disponibili (es. canali occupati, utenti connessi, tempi di inizio e fine conferenza) delle varie audioconferenze. Il problema trova soluzione nell'applicazione di semafori che regolano l'accesso a tale risorsa condivisa, quindi se in un certo istante un processo sta utilizzando il suddetto array nessun altro thread può utilizzarlo, quando l'aggiornamento o la modifica è finita il primo processo lascia posto al secondo e così via. L'utilizzo della programmazione concorrente in questa applicazione è indispensabile perchè ogni linea telefonica e ogni audioconferenza è gestita da un singolo processo il quale ha l'assoluto controllo della risorsa fisica, quindi per avere contemporaneamente più utenti connessi e più conferenze attive simultaneamente c'è bisogno di attivare un numero equivalente di thread.

## 4 Diagrammi di flusso delle principali funzionalità implementate

In questo capitolo si trovano i principali diagrammi di flusso che sono stati realizzati in una delle primissime fasi di sviluppo del software, ognuno di essi rappresenta il percorso logico che ogni parte del programma deve eseguire per soddisfare tutti i requisiti precedentemente determinati. Tutti i flowchart sono stati revisionati dal team interno di sviluppo e infine sono stati inseriti nella documentazione globale del progetto.

### 4.1 Funzionamento audioconferenza generale

Il primo diagramma (Fig. 5.1) rappresenta il percorso più generale possibile che il programma compie nel caso di chiamata verso il servizio. Il primo passo è quello capire quale tipo di conferenza telefonica ha scelto l'utente, qui ci sono due possibili strade iniziali a seconda della scelta, se la tipologia scelta è la *MeetTogether* allora il passo seguente sarà quello di andare sul processo chiamato Aconference, finita la conference automatica si andrà al punto di fine. Invece per qualsiasi altra Audio conferenza diversa da quella precedente il software procede in altro modo, cioè accetta la chiamata entrante, risponde ad essa, salva i tasti digitati dal chiamante che conterranno il PIN, successivamente procede al processo di identificazione dell'utenete, se il PIN è corretto allora attiva il processo generale della conferenza, se il PIN è errato dopo un numero finito di tentativi oppure se la conferenza è finita il programma rilascia tutte le risorse liberando i canali voce e arriviamo così nuovamente al punto di fine attività.

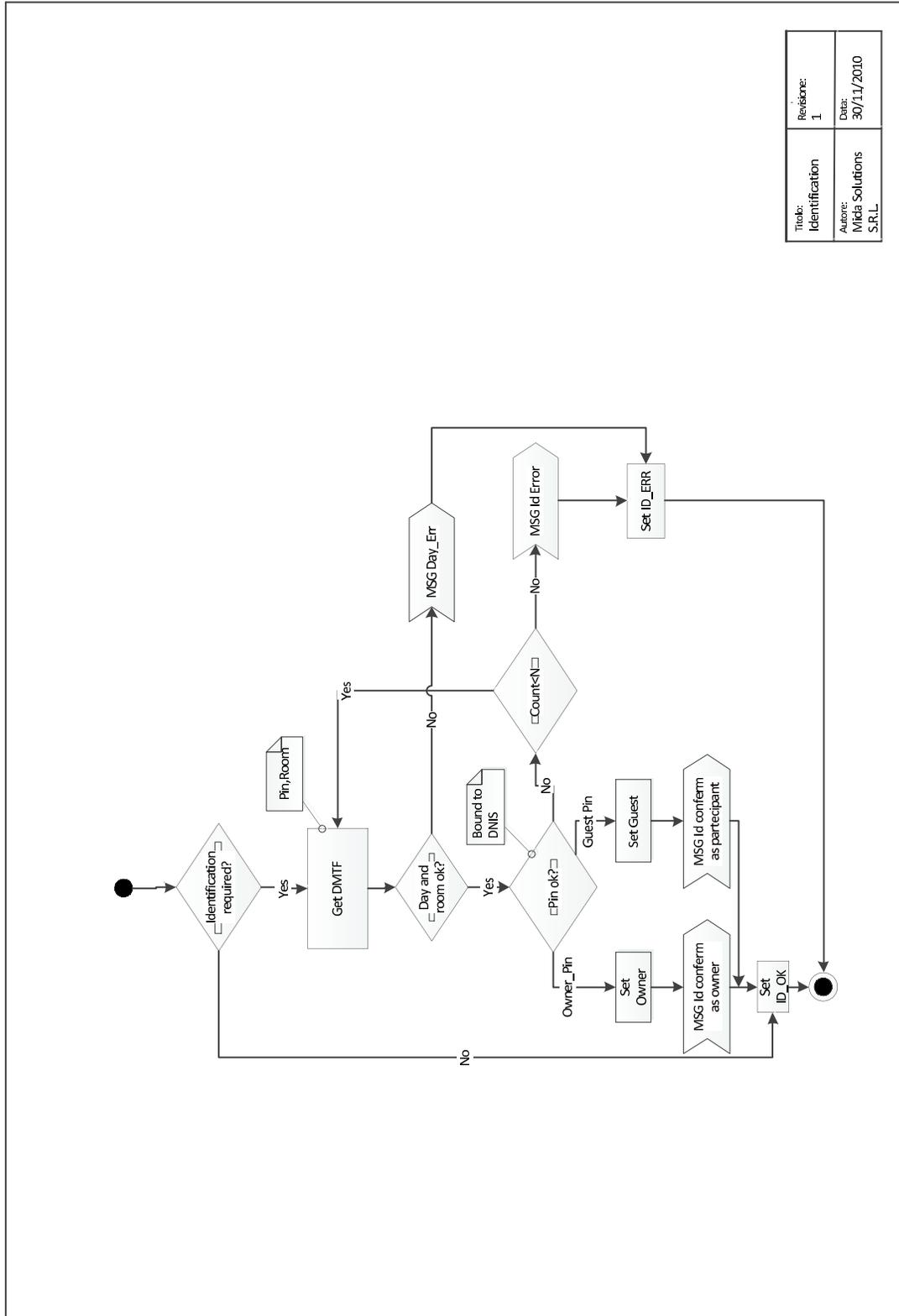


|                                      |                     |
|--------------------------------------|---------------------|
| Titolo:<br>Conference Box            | Revisione:<br>1     |
| Autore:<br>Midea Solutions<br>S.R.L. | Data:<br>25/11/2010 |

Figura 5.1: Diagramma audioconferenza generale

## 4.2 Processo di identificazione

In questa seconda attività si accede qualora nel momento della prenotazione della conferenza telefonica venga selezionato il campo che attiva l'identificazione, permettendo così di entrare ai soli possessori del PIN corretto. Nel caso si voglia permettere a chiunque di poter intervenire liberamente questa parte di programma verrà saltata. Come si vede in fig. 5.2, la prima cosa che si verifica è se l'utente ha impostato o meno l'obbligatorietà dell'identificazione, se non l'ha selezionata l'intero processo viene oltrepassato e si ritorna al passo successivo presente nella figura 5.1, viceversa l'applicazione tiene in memoria la combinazione dei tasti digitati del chiamante e controlla che la data, l'ora e il PIN ricevuto siano corretti. La risposta negativa ora può essere provocata da due diversi eventi, il caso in cui sia sbagliata la data o l'ora di attivazione della conferenza oppure il caso di PIN errato. In entrambi viene fatto sentire al chiamante un messaggio vocale di errore che lo informa sulla non corrispondenza tra l'ora attuale e l'ora della prenotazione o sull'errata composizione del PIN. Se invece la risposta risulta essere affermativa, tramite il codice identificativo digitato il chiamante viene impostato come amministratore della room con tutti i privilegi del caso oppure semplicemente come ospite. Conclusa questa attività che viene richiamata dal processo generale di fig. 5.1, si passerà al passo successivo di tale processo.



|                        |            |
|------------------------|------------|
| Tracciato:             | Revisione: |
| Identification         | 1          |
| Autore:                | Data:      |
| Mhida Solutions S.R.L. | 30/11/2010 |

Figura 5.2: Diagramma attività identificazione

### 4.3 Attività della conference

L'attività descritta in questa sezione (fig. 5.3) è una delle più importanti in quanto risulta essere il nucleo di tutto l'algoritmo di implementazione delle funzionalità utilizzate all'interno del sistema vocale interattivo da me creato. Innanzitutto una volta entrati nel sistema viene riprodotto un messaggio audio di benvenuto pre-registrato, il secondo step è quello di verificare se l'utente in questione è o meno l'amministratore della stanza, in caso affermativo esso viene aggiunto come *Talker*<sup>3</sup>. Col passo seguente si entra nell'attività di *MakeCall* la quale provvede a inoltrare parallelamente un numero di chiamate corrispondente ai membri che sono stati selezionati dall'amministratore durante la prenotazione della conferenza, in pratica tutti gli altri partecipanti. È stato previsto in caso di non risposta un numero massimo di tentativi dopo i quali il chiamato viene escluso dal meeting. Finita questa attività si arriva ad un altro blocco informativo il quale avverte l'utente che è stato aggiunto alla conferenza, anche se non è l'amministratore della room, con la variante che se ci troviamo in una tipologia *VoiceCast* si viene aggiunti solamente come Listener,<sup>4</sup> per le restanti due tipologie di audioconferenza si viene aggiunti come Talker. Una volta entrati veramente nella chiamata di gruppo, viene lanciata periodicamente l'attività di *Input Management* ( par 5.4 ) la quale controlla tutti i comandi che vengono digitati dai membri della conferenza. Ci sono due modi possibili per uscire dalla room e sono:

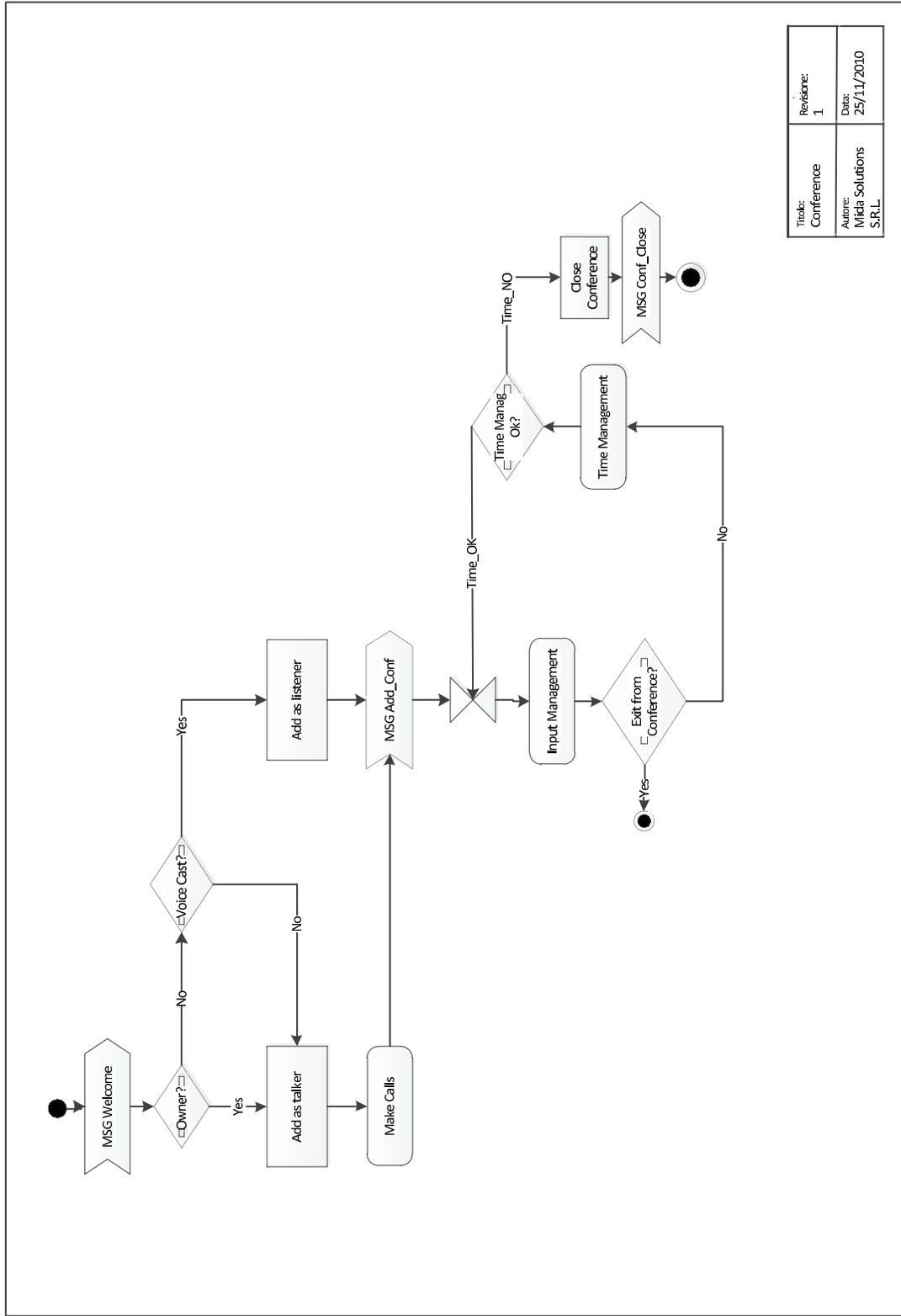
- digitare uno specifico comando;
- attendere la fine del tempo a disposizione: il controllo del tempo viene effettuato automaticamente dal programma attraverso l'attività, sempre periodica, di *Time Managment* descritta nel sottocapitolo 5.5. Quando il tempo è finito si chiude la room e si riproduce un messaggio di saluto

Usciti dalla conferenza telefonica si ritorna al processo generale di figura 5.1 che come detto prima ha il compito finale di rilasciare tutte le risorse precedentemente occupate e renderle di nuovo disponibili per altre conferenze.

---

<sup>3</sup>Talker: con questo termine si specificano le proprietà con cui viene aggiunto un utente in una conferenza, in questo caso può parlare e ascoltare simultaneamente

<sup>4</sup>Listener: con questo termine definiamo l'utente che accede alla conferenza ma che ha solamente la possibilità di ascoltare



|                                      |                     |
|--------------------------------------|---------------------|
| Titolo:<br>Conference                | Revisione:<br>1     |
| Autore:<br>Midea Solutions<br>S.R.L. | Data:<br>25/11/2010 |

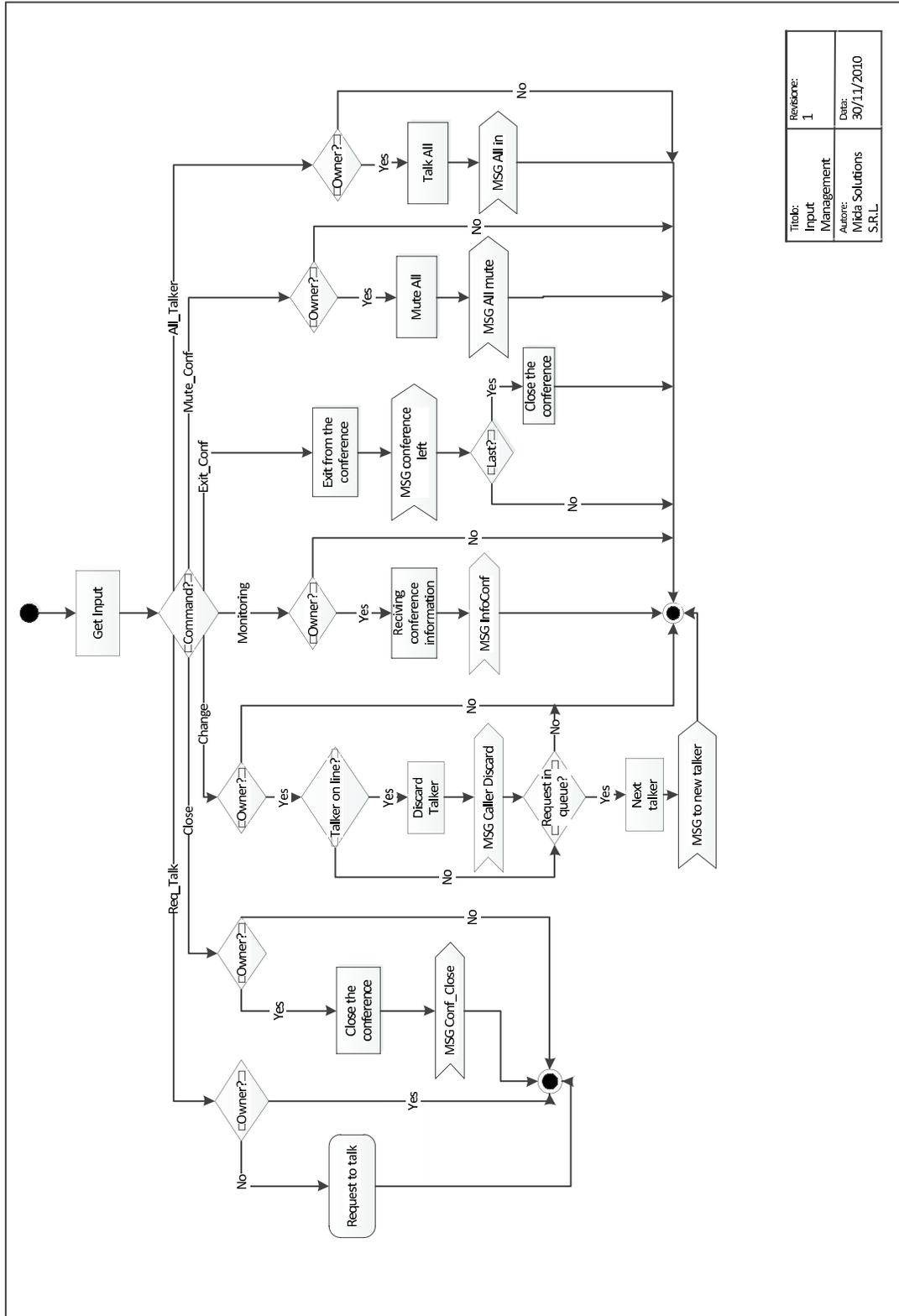
Figura 5.3: Diagramma attività della conferenza

## 4.4 Attività di gestione degli input

Nella seguente attività ( fig. 5.4 ) vengono esplicitati tutti i comandi che l'amministratore o gli altri utenti possono impartire alla conferenza nella quale si trovano. Il primo passo comune a tutti gli input è quello di verificare chi ha fatto la richiesta per un determinato ordine, infatti alcuni comandi sono riservati al solo amministratore. I comandi vengo comunicati tramite la digitazione di predefinite sequenze di numeri presenti sulla tastiera telefonica che vengono successivamente convertiti in segnali: questo sistema di codifica è chiamato *Dual-Tone Multi-Frequency (DTMF)*, i segnali vengono riconosciuti dal server e riconvertiti in comandi.

1. **RequestToTalk:** questo comando può essere abilitato nella modalità *VoiceCast* se viene selezionato il campo PTT, e invia all'amministratore della room un tono con il quale lo si avvisa che un utente chiede il permesso di intervenire;
2. **Close:** serve per chiudere la conferenza e invia su ogni canale voce un messaggio in cui si informano i partecipanti che la conferenza è stata terminata;
3. **Change:** comando attivo solo in modalità *VoiceCast*, quando viene composta la sequenza di questo comando esso controlla se c'è già qualcuno che sta parlando in quel caso lo scarta e passa la parola al successivo partecipante sulla lista dei partecipanti che richiedono la parola; nel caso in cui la lista sia vuota non fa nulla.
4. **Monitoring:** utilizzando questo comando si ottengono le informazioni principali sullo stato della conferenza come ad esempio partecipanti in linea, tempo trascorso, tempo rimanente etc.
5. **Exit:** serve per uscire singolarmente dalla conference, se questo comando è impartito dall'amministratore si chiude la conference tranne nel caso di *Conference Bridge*;
6. **Mute Conference:** serve a zittire tutti i partecipanti, i quali potranno solo ascoltare, tranne l'amministratore;
7. **Talk All:** passa tutti i partecipanti da soli Listener a Talker quindi tutti hanno la possibilità di parlare;

I comandi individuati dai punti 2, 3, 4 e 6 possono essere utilizzati solamente dall'amministratore dell'audioconferenza, il comando del punto 1 può essere utilizzato esclusivamente da un partecipante qualsiasi e infine il comando del punto 5 può essere usato da tutti i partecipanti senza alcuna distinzione.



|                       |            |
|-----------------------|------------|
| Titolo:               | Revisione: |
| Input Management      | 1          |
| Autore:               | Data:      |
| Mhda Solutions S.R.L. | 30/11/2010 |

Figura 5.4: Diagramma attività di gestione input

#### 4.4.1 Cenni sul sistema DTMF

Il Dual-tone multi-frequency in sigla DTMF, chiamato in italiano anche multifrequenza<sup>5</sup>, è un sistema di codifica usato in telefonia per codificare codici numerici sotto forma di segnali sonori in banda audio. Il sistema è utilizzato per trasmettere alla centrale telefonica i numeri digitati sulla tastiera del telefono, ma anche per telecontrollare servizi di telefonia, sistemi di integrazione computer/telefono, segreterie telefoniche e per comporre codici di carte di pagamento. La tastiera DTMF è costituita da una matrice 4X4, in cui ogni riga rappresenta una frequenza bassa, e ogni colonna rappresenta una frequenza alta; premendo per esempio il tasto 1 vengono inviate due onde sinusoidali alle frequenze di 697 e 1209 Hz. Le frequenze sono state scelte in modo che le armoniche e le intermodulazioni non generino segnali rilevanti. Nessuna frequenza è un multiplo intero di un'altra e la differenza e la somma tra due frequenze non corrisponde ad alcun tono. Le frequenze sono in rapporto 21/19, leggermente inferiore ad un tono del sistema temperato, ed il valore deve rimanere entro una tolleranza di 1,5% per essere riconosciuta dalla centrale. L'intensità del tono alto deve essere uguale o maggiore rispetto al tono basso, con una differenza massima di 3 dB, chiamata "twist".

### 4.5 Attività di gestione del tempo della conferenza

Il primo passo quando si entra in questa attività è quella di verificare se il tempo per cui sono disponibili le risorse della room è scaduto, se non è così la conferenza continua normalmente altrimenti entriamo in un ulteriore blocco decisionale nel quale il programma controlla se la conferenza può essere estesa. In caso negativo si esce da questa parte di routine e la conferenza viene chiusa avvisando tutti i partecipanti, altrimenti si chiede all'amministratore di quanto vuole prolungare il meeting ed inserito il nuovo riferimento temporale la conferenza riprende normalmente. Nella figura 5.5 viene riportato lo svolgimento a livello logico l'algoritmo in questione.

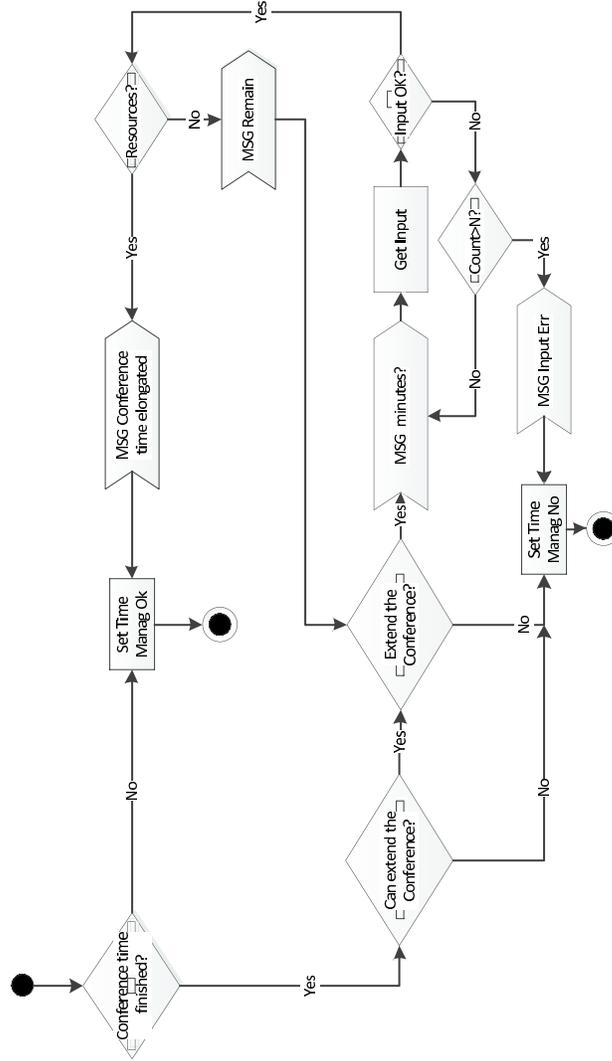
### 4.6 Attività di conferenza automatica

Questa viene attivata solo nel caso in cui la conferenza sia di tipo *MeetTogether*. L'amministratore, selezionando questo tipo di audioconferenza attiva un orologio interno al sistema che arrivato all'ora e alla data stabilita chiama tutti i numeri inseriti nella lista dei membri di una determinata room. Questa funzionalità (fig. 5.6) risulta essere molto utile nel caso in cui si vogliono pianificare riunioni periodiche

---

<sup>5</sup>Il termine multifrequenza deriva da un uso contemporaneo di due toni.

automatiche, ovviamente è sempre possibile richiedere di identificarsi per aumentare la sicurezza.



|                                   |                     |
|-----------------------------------|---------------------|
| Titolo:<br>Time Management        | Revisione:<br>1     |
| Autore:<br>Mhida Solutions S.R.L. | Data:<br>30/11/2010 |

Figura 5.5: Diagramma attività di gestione del tempo di conferenza

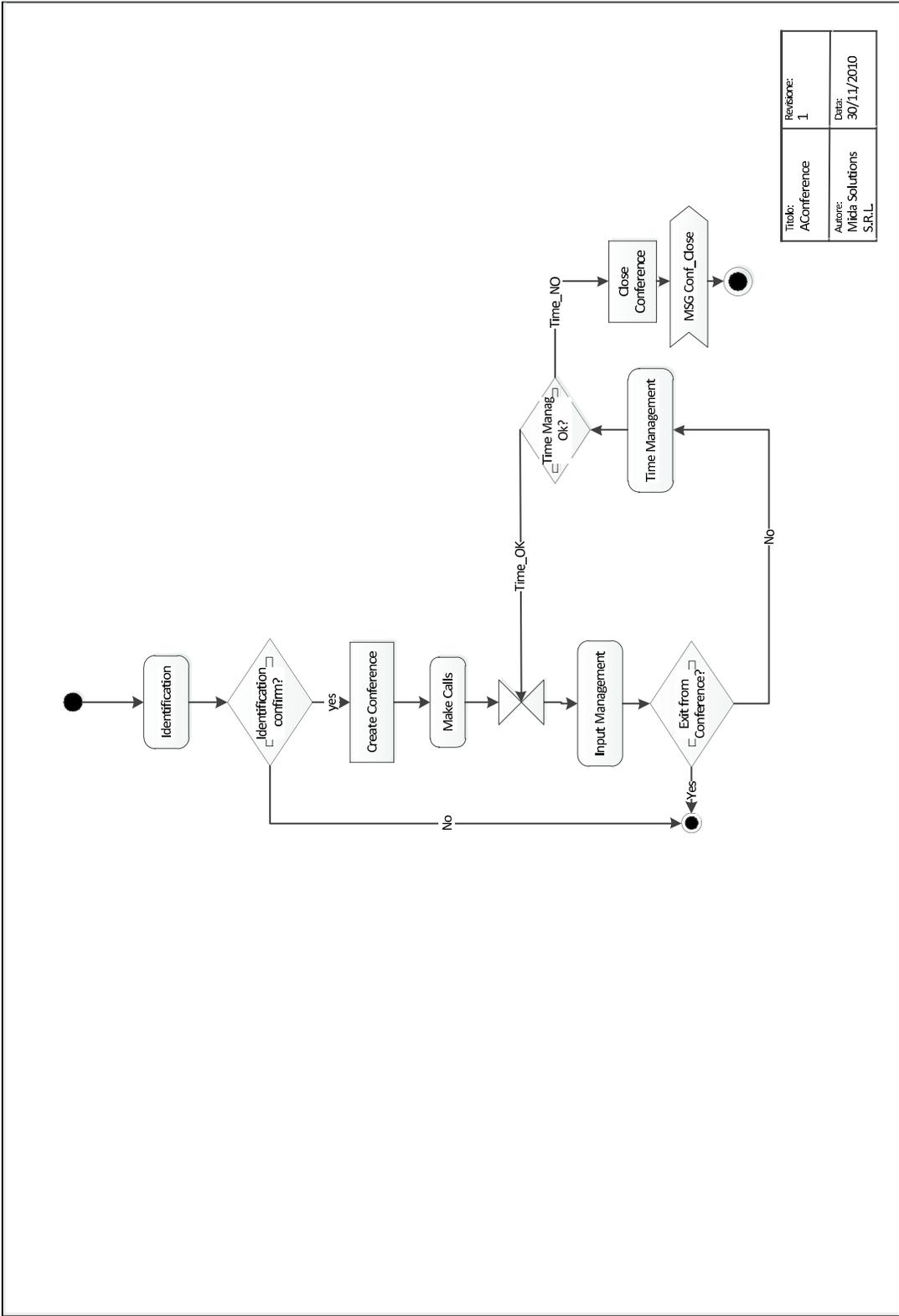


Figura 5.6: Diagramma attività della conferenza automatica

## 5 Progetto UML del database

In questo capitolo verranno illustrate attraverso una breve descrizione tutte le entità ed i rispettivi attributi, che costituiscono il database in cui vengono salvate le prenotazioni per le conferenze telefoniche e i dati utili a loro connesse, che vanno dal semplice numero telefonico per accedere al servizio alla lista completa dei partecipanti ad una specifica audioconferenza. Questi verranno passati come parametri d'ingresso nell'applicazione. Come si vede dalla Fig. 6.1 il database è composto da

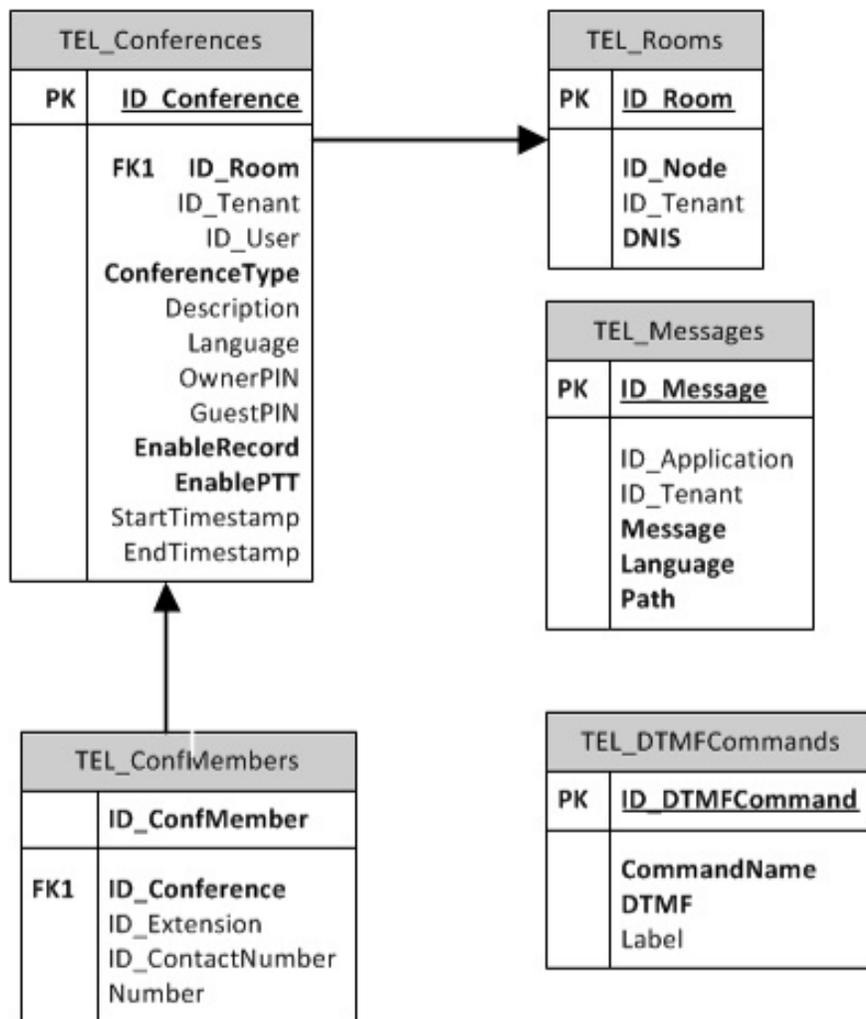


Figura 6.1: Tabelle Database

cinque tabelle:

1. **TELConferences**: rappresenta e contiene tutti i record specifici che distinguono ogni singola conferenza, come per esempio l'IDConferences numero unico e identificativo o il tipo di audioconferenza. La tabella in questione presenta un legame uno a uno con altre due tabelle TELConfMembers e TELRooms;
  - **ID\_Conference**: chiave primaria di questa tabella, questo campo è un numero sequenziale univoco che identifica la conferenza;
  - **ID\_Room**: chiave esterna utilizzata per collegare la tabella in questione con la tabella *TELRooms*;
  - **ID\_Tenant**: numero sequenziale unico per identificare il tenant<sup>6</sup> a cui è abbinato il messaggio;
  - **ID\_User**: numero identificativo dell'utente associato alla conferenza;
  - **Conference Type**: tipo di conferenza scelto al momento della prenotazione;
  - **Description**: eventuali commenti o specificazioni sull'audioconferenza;
  - **language**: lingua in cui è configurata la conferenza;
  - **OwnerPIN**: numero assegnato all'amministratore per accedere alla conferenza;
  - **GuestPIN**: numero assegnato dall'amministratore per permettere agli altri partecipanti di accedere alla conferenza;
  - **EnableRecord**: campo di tipo boolean che se impostato a True permette la registrazione dell'intera conferenza;
  - **EnablePTT**: campo di tipo boolean che permette l'attivazione del push to talk se la tipologia della conferenza è un *VoiceCast*;
  - **StartTimestamp**: contiene l'ora di inizio dell'audioconferenza;
  - **EndTimestamp**: contiene l'ora di conclusione della conferenza;
2. **TELConfMembers**: contiene i record inerenti a ciascun membro che pu ò accedere ad una determinata conferenza o che verrà chiamato automaticamente se inserito in questa lista all'atto della prenotazione;

---

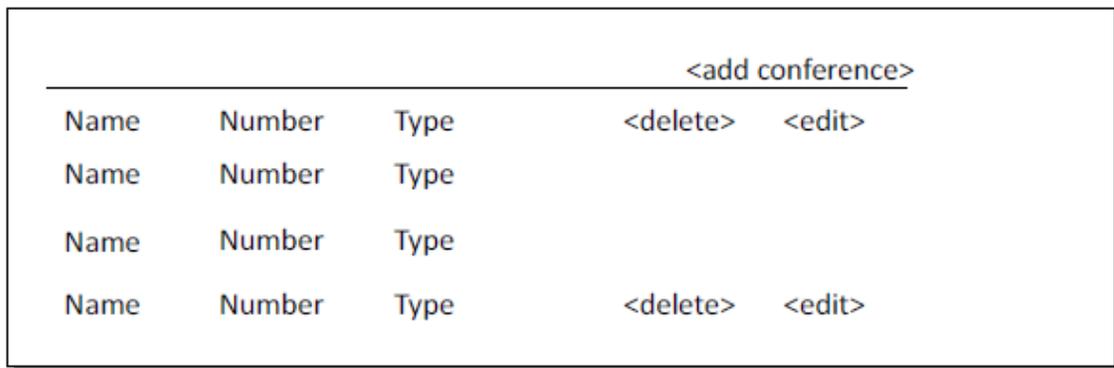
<sup>6</sup>Tenant è un partizionamento logico del sistema che permette ad un membro di un tenant di vedere tutto e solo quello che riguarda il proprio tenant

- ***ID\_ConfMemeber***: utilizzata come chiave primaria in questa tabella, è numero sequenziale identificativo unico per ogni lista dei membri di una conferenza;
  - ***ID\_Conference***: chiave esterna che collega la tabella *TELConfMembres* alla tabella *TELConferences*;
  - ***ID\_Extension***: numero sequenziale di identificazione dell'estensione collegata alla lista dei partecipanti;
  - ***ID\_ContactNumber***: numero identificativo unico per ogni contatto presente nella lista dei partecipanti alla conferenza;
  - ***Number***: numero telefonico di ciascun potenziale partecipante alla conferenza presente nella lista dei membri;
3. **TELRooms**: nel record di questa tabella è custodito il numero telefonico che l'utente deve chiamare per accedere ad una conferenza;
- ***ID\_Room***: utilizzata come chiave primaria, è un numero sequenziale identificativo unico per ogni room;
  - ***DNIS***: numero di telefono a cui è associata ogni conferenza o servizio;
  - ***ID\_Node***: numero sequenziale identificativo unico identificativo del nodo a cui è associato il messaggio;
  - ***ID\_Tenant***: numero sequenziale unico per identificare il tenant a cui è abbinato il messaggio;
4. **TELMessages**: nel suo interno sono salvati i percorsi relativi alla posizione in cui si trovano i messaggi vocali preregistrati che informano sulle operazioni che l'utente può eseguire;
- ***ID\_Message***: è la chiave primaria della tabella ed è un numero sequenziale identificativo unico per ogni messaggio;
  - ***ID\_Application***: numero sequenziale identificativo unico dell'applicazione a cui è associato il messaggio;
  - ***ID\_Tenant***: numero sequenziale unico per identificare il tenant a cui è abbinato il messaggio;
  - ***Message***: nome assegnato a ciascun messaggio;
  - ***Language***: lingua con la quale verrà riprodotto il messaggio

- ***Path***: percorso specifico per localizzare dov'è situato il file audio del messaggio;
5. **TELDTMFCommands**: in ogni record è inserita la combinazione di tasti che l'utente deve premere per far eseguire all'applicazione determinati comandi prestabiliti;
- ***ID\_ConfMember***: numero sequenziale identificativo unico per ogni lista di membri della conferenza utilizzato come chiave primaria della tabella;
  - ***ID\_DTMFCommand***: numero sequenziale identificativo unico per ogni comando, questo campo è la chiave primaria della tabella;
  - ***Command Name***: nome del comando;
  - ***DTMF***: sequenza di tasti a cui verrà associato ciascun comando e che verranno convertiti in DTMF;
  - ***Label***: etichetta per descrivere il comando a cui è associata;

## 6 Pagine Web di configurazione e di prenotazione per l'utente finale

In quest'ultima sezione trovano spazio le descrizioni delle varie interfacce grafiche per gli utenti finali e di configurazione per installare il sistema, verranno quindi descritti e illustrati tutti i campi e i bottoni presenti sulle pagine web. La progettazione delle pagine rientra nella parte conclusiva di tutto il lavoro svolto durante il periodo di tirocinio. La figura 7.1 rappresenta la pagina accessibile dal menu, utilizzata dall'utente per visualizzare tutte le audioconferenze già prenotate e presenti sul DataBase, che viene aggiornata periodicamente per permettere a tutti di vedere se ci sono a disposizione risorse per usufruire del servizio. I campi ed i tasti presenti nell'interfaccia



| <u>&lt;add conference&gt;</u> |        |      |          |        |
|-------------------------------|--------|------|----------|--------|
| Name                          | Number | Type | <delete> | <edit> |
| Name                          | Number | Type |          |        |
| Name                          | Number | Type |          |        |
| Name                          | Number | Type | <delete> | <edit> |

Figura 7.1: Pagina riassuntiva delle conferenze già prenotate con i rispettivi dettagli

riassuntiva di tutte le conferenze sono:

- **Name:** nome assegnato ad ogni singola conferenza nel momento della prenotazione;
- **Number:** numero della stanza(room) assegnato ad ogni conferenza;
- **Type:** indica il tipo di conferenza tra i quattro disponibili;
- **AddConference:** bottone che viene utilizzato per aggiungere una nuova conferenza, se non ci sono room disponibili il tasto in questione viene sostituito da un messaggio di warning;
- **Delete:** utilizzato per rimuovere la rispettiva conferenza;
- **Edit:** il tasto permette la configurazione della conferenza ;

Un'altra pagina fondamentale è la pagina di configurazione dell'audioconferenza (fig. 7.2); questa pagina viene visualizzata dall'utente che sta effettuando la prenotazione per poter utilizzare il sistema e indire un meeting telefonico multiplo, quando viene creata una nuova conferenza. Nel momento del salvataggio il sistema assegna una room presa dalla lista creata dall'amministratore in base a quelle disponibili nel giorno e nell'orario richiesti dall'utente. I campi presenti nella pagina e modificabili

The image shows a web form for configuring a conference. It includes the following elements:

- Name:** A text input field.
- Conference type:** A dropdown menu.
- Language:** A dropdown menu.
- Identification:** A checked checkbox.
- Pin owner:** A text input field.
- Pin guest:** A text input field.
- Day:** A text input field.
- Time start:** A text input field.
- Time end:** A text input field.
- PTT:** A checked checkbox.
- Member list box:** A large empty rectangular box with the text "<Add member>" in the top right corner and "Member list box" centered below it.

Figura 7.2: Pagina di configurazione di una singola conferenza

dall'utente sono:

- **Name:** nome che l'utente vuole dare alla propria conference box;
- **Type Conference:** lista drop down con i vari tipi di conference. Se l'utente è di tipo Amministratore, il tipo di conferenze selezionabili sono : *Conference Bridge*, *MeetMe*, *MeetTogether* e *VoiceCast* altrimenti potrà scegliere solo tra: *Conference Bridge* o *MeetTogether*;

- **Language:** l'utente sceglie tra la lista di lingue del portale. Questo configura la lingua dei messaggi che verranno emessi dalla conference box;
- **Identification:** Flag che stabilisce se è necessario un PIN per accedere alla conference;
- **Pin Owner/Room:** vengono inseriti dall'utente il pin per l'Owner della conference (Guest) e il pin per accedere ad una determinata conference, questi campi vengono attivati solamente se il flag identification è selezionato;
- **Day:** sarà inserita la data per prenotare le risorse richieste dalla conference, visibile solo se la conference è di tipo *Conference Bridge*, *MeetTogether* oppure *MeetMe*;
- **Time Start e Time end:** sarà selezionata l'ora d'inizio della conference, ad esempio scegliendole da una rubrica che contenga le conference prenotate in quel giorno;
- **PTT:** checkbox che compare solamente se la conference è un voice cast, si decide se attivare o meno la modalità push to talk;

### Gestione membri della conference

Visibile all'utente solamente se la conference non è di tipo *Conference Bridge*

- **Member List box:** lista paginata di tutti i membri che si vogliono far partecipare alla conference con possibilità di eliminarli. I membri sono numeri telefonici che vengono inseriti dalla rubrica o direttamente a mano;
- **Bottone Add Member:** apre un pop up di possibili contatti i membri che voglio far partecipare alla conference;

Nella figura 7.3 viene riprodotta la pagina di configurazione dei comandi presenti in una conferenza, ossia ad ogni comando viene associata una sequenza di numeri i quali verranno catturati mediante il sistema DTMF dall'applicazione.

1. Tutti i campi contengono la combinazione di tasti per far eseguire il comando in questione;
2. I campi Close conference, Status conference, Exit conference e Mute all conference sono visibili per tutti i tipi di conference;
3. I campi Add as talker, Change talker, Request to talk e VoiceCast talk access control(flag) appaiono soltanto nella modalit VoiceCast;

|                                   |                      |               |                      |
|-----------------------------------|----------------------|---------------|----------------------|
| Close conf.                       | <input type="text"/> | Status conf.  | <input type="text"/> |
| Exit conf.                        | <input type="text"/> | Mute conf.    | <input type="text"/> |
| Change talker                     | <input type="text"/> | Add as talker | <input type="text"/> |
| Request to talk                   | <input type="text"/> | Rem. mute     | <input type="text"/> |
| VoiceCast talk<br>accesss control | <input type="text"/> |               |                      |

Figura 7.3: Pagina di configurazione dei Comandi presenti nelle conferenze

4. Il campo Remove mute non appare quando la conference è di tipo VoiceCast e viene utilizzato solo dai guest users;

La figura 7.4 di pagina, caratterizzata da tre soli campi, serve a far reagire il programma nel caso di mancata risposta da parte del numero contattato. Questo permette di iniziare la conferenza anche se non sono presenti tutti i partecipanti e di conoscere immediatamente quanti utenti sono presenti nella room dell'audioconferenza.

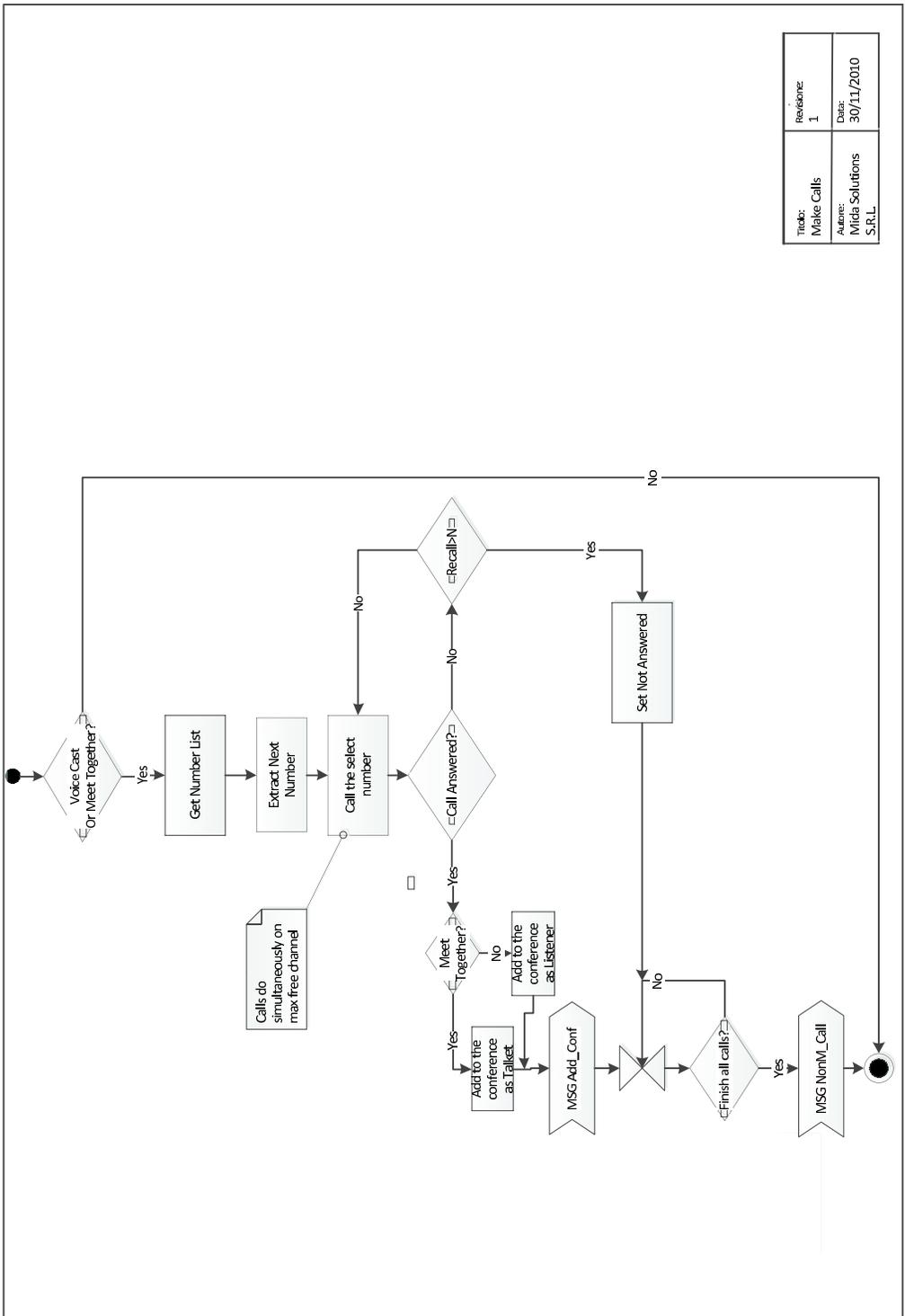
|  |                      |                   |                      |
|--|----------------------|-------------------|----------------------|
| Max number of<br>attempts of the<br>conference | <input type="text"/> | No answer timeout | <input type="text"/> |
|  |                      | Retry interval    | <input type="text"/> |

Figura 7.4: Pagina delle opzioni di chiamata

- **Max Number of:** l'utente deve poter selezionare il numero di tentativi massimi che ha a disposizione per contattare il numero;
- **No Answer Timeout:** l'utente deve poter impostare, in secondi, il tempo massimo d'attesa di non risposta sul numero chiamato;
- **Retry interval:** saranno settati il numero massimo di tentativi di chiamata, da effettuare su tutti i numeri della Member List;

Sarà presente anche una pagina che permette di gestire i messaggi audio riprodotti all'interno del sistema, verrà visualizzata la lista dei messaggi con possibilità di cancellazione, download e se possibile l'ascolto.

## A Appendice



|         |                        |            |            |
|---------|------------------------|------------|------------|
| Titolo: | Make Calls             | Revisione: | 1          |
| Autore: | Milda Solutions S.R.L. | Data:      | 30/11/2010 |

Figura 1: Diagramma della funzione che effettua le chiamate

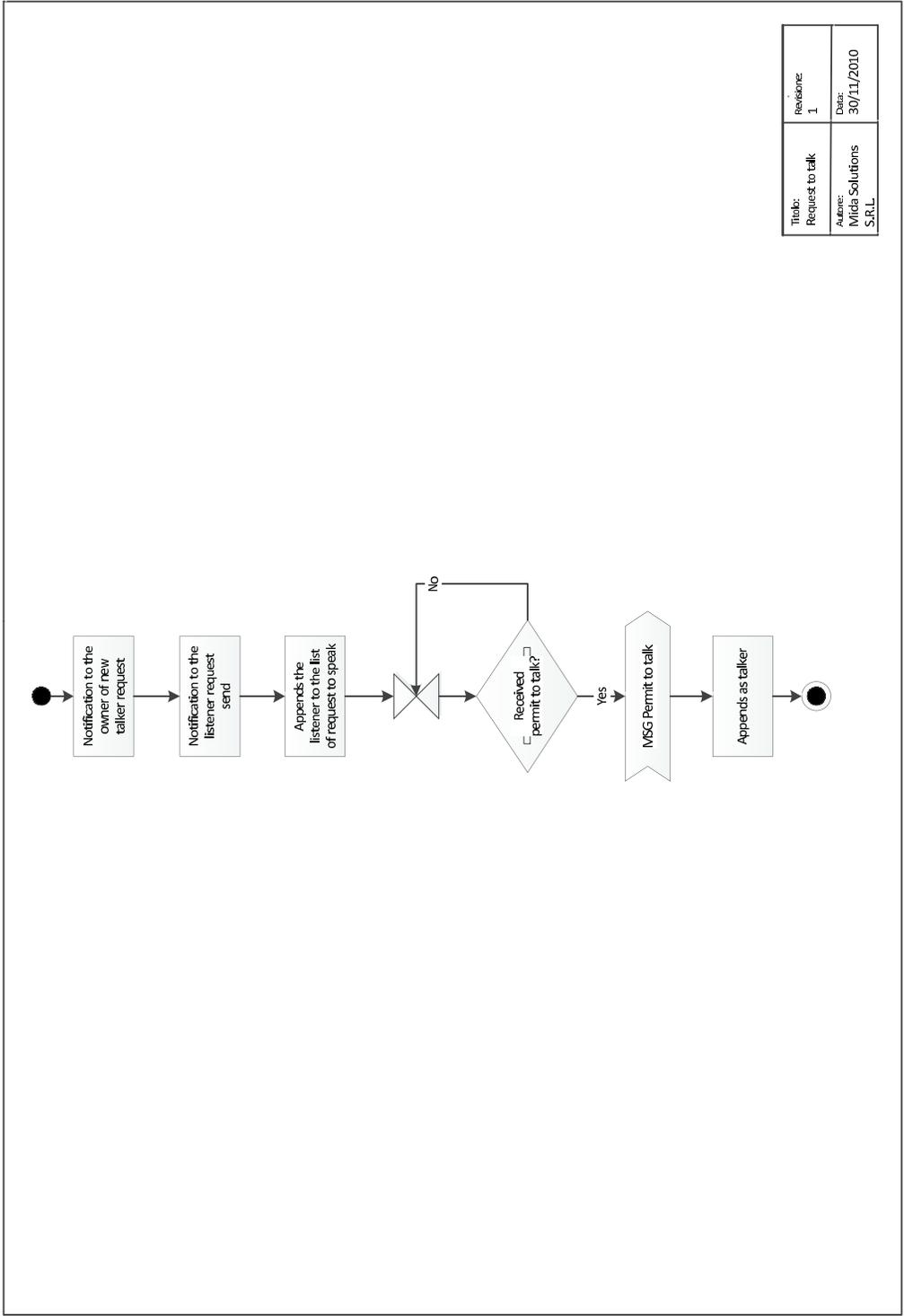


Figura 2: Diagramma della funzione che inoltra la richiesta di parlare all'amministratore

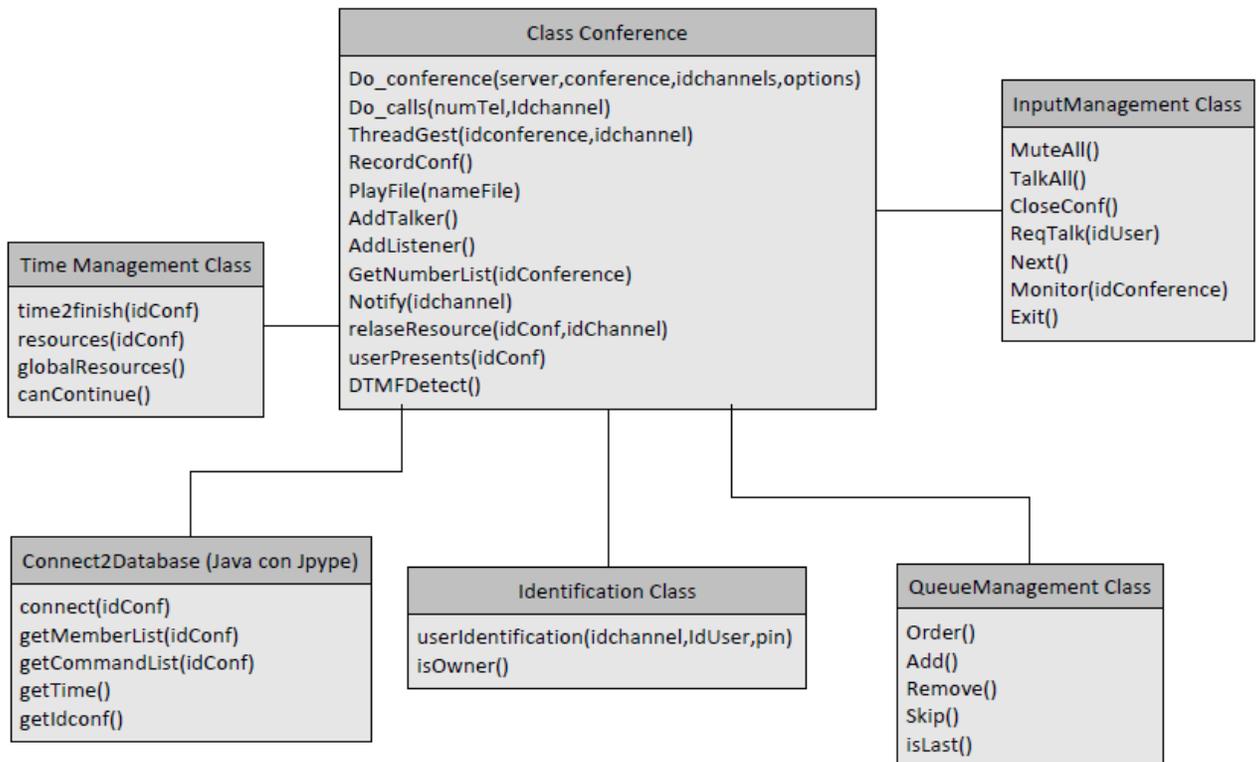


Figura 3: Diagramma delle classi e dei rispettivi metodi

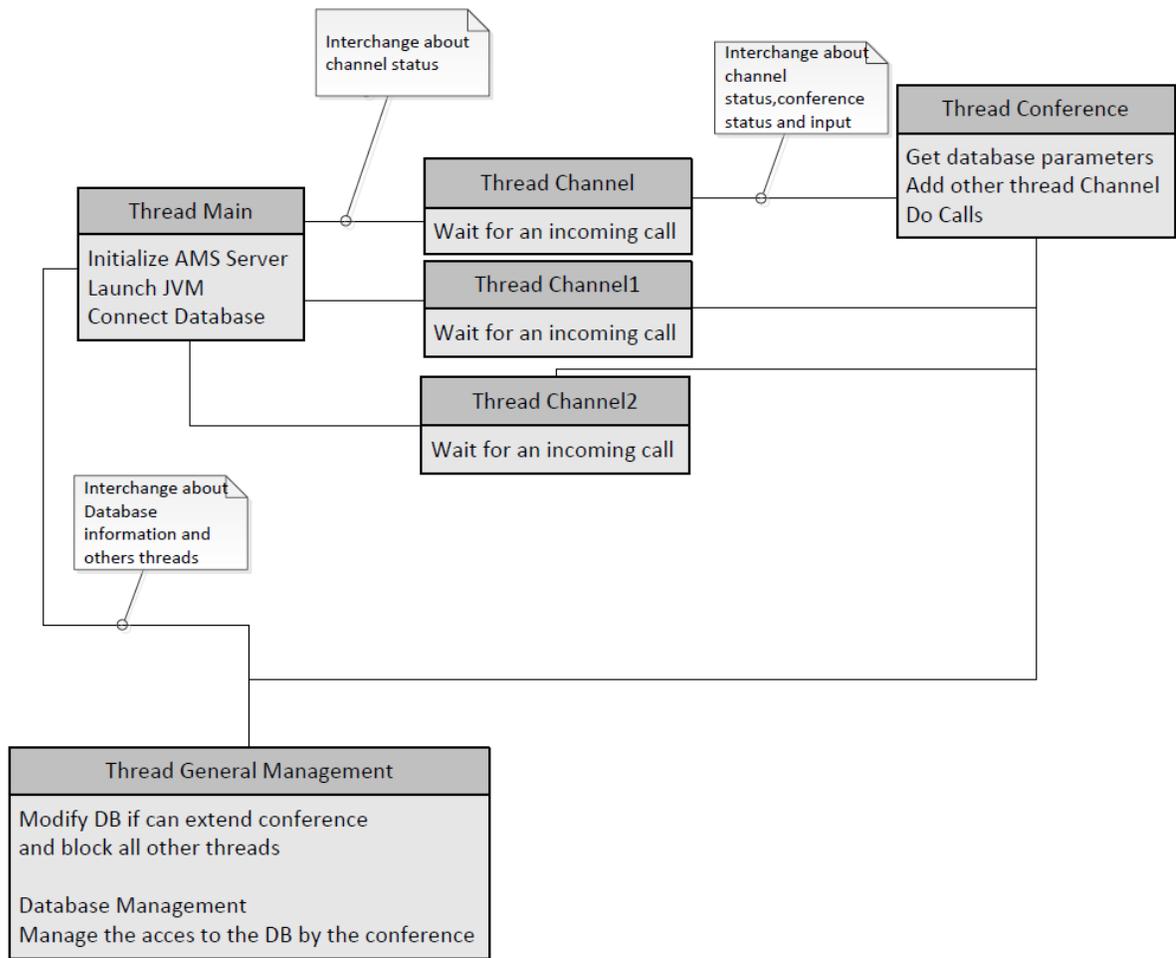


Figura 4: Diagramma dei thread attivi nel sistema e loro interazione

## Indice analitico

- Architettura e Componenti, 10
- Attivazione audioconferenza, 15
- Beta Testing, 9
- Comandi e Gestione Input, 23
- Comunicazione con DataBase e JPytype, 9
- Conference Bridge, 13
- Diagramma attività di identificazione utenti, 19
- Diagramma dei thread attivi nel sistema e loro interazione, 42
- Diagramma del processo interno di una conferenza, 21
- Diagramma della funzione che effettua le chiamate, 39
- Diagramma della funzione che inoltra la richiesta di parlare all'amministratore, 40
- Diagramma delle classi e dei rispettivi metodi, 41
- Diagramma Generale di un'Audioconferenza, 17
- Diagramma UML del Database, 29
- DTMF, 25
- Figura 7.1: Pagina riassuntiva delle conferenze già prenotate con i rispettivi dettagli, 33
- Figura 7.3: Pagina di configurazione dei Comandi presenti nelle conferenze, 36
- Figura 7.4: Pagina delle opzioni di chiamata, 36
- Full Duplex e Half Duplex, 13
- Ingegnerizzazione del software, 6
- Linguaggio Python, 7
- Meet Together, 14
- MeetMe, 14
- Multithreading e Programmazione Concorrente, 16
- Pagina di configurazione di una singola conferenza, 34
- Pagine web per l'interfacciamento con l'utente, 33
- Prenotazione audioconferenza, 14
- Schemi UML, 6
- Semafori e Condivisione di risorse, 16
- Specifiche e Requisiti, 12
- Stanze per conferenze(Rooms), 12
- Tabelle e Record del Database, 30
- Talker e Listener, 21
- Time Management, 25
- Voice over IP, 9
- VoiceCast, 13