

UNIVERSITA' DEGLI STUDI DI MODENA E REGGIO EMILIA

Dipartimento di Ingegneria "Enzo Ferrari"

Corso di Laurea Magistrale in Ingegneria Informatica

Assistenza Interventi Tecnici: progetto e sviluppo di un'App ibrida attraverso il framework Phonegap

Relatore:
Prof. Sonia Bergamaschi

Candidato:
Gianluca Stazione

Anno Accademico 2013-2014

Indice

1	Introduzione	7
2	L'app ibrida Assistenza Interventi Tecnici	9
2.1	User Case Diagram	9
2.1.1	Gerarchia degli attori	9
2.1.2	Primo accesso	10
2.1.3	Home page	11
2.1.4	Azienda	12
2.1.5	Clienti	13
2.1.6	Materiali	14
2.1.7	Inserisci chiamata	15
2.1.8	Gestisci chiamata	16
2.1.9	Storico	31
2.1.10	Impostazioni	32
2.2	Activity Diagram	33
2.2.1	Anagrafica Azienda	33
2.2.2	Tecnici	35
2.2.3	Prestazioni	36
2.2.4	Trasferte	37
2.2.5	Aliquota d'iva	38
2.2.6	Clienti	39
2.2.7	Materiali	40
2.2.8	Inserisci chiamata	41
2.2.9	Gestisci chiamata	42
2.2.10	Storico	44
2.2.11	Impostazioni	45
2.3	Sequence Diagram	46
2.3.1	Anagrafica Azienda	46
2.3.2	Tecnici	47
2.3.3	Prestazioni	48
2.3.4	Trasferte	49

2.3.5	Aliquota d'iva	50
2.3.6	Condizioni generali di fornitura	51
2.3.7	Clienti	52
2.3.8	Materiali	53
2.3.9	Inserisci Chiamata	54
2.3.10	Gestisci Chiamata	55
2.3.11	Storico	57
2.3.12	Impostazioni	58
2.4	Modello E-R	59
2.5	Descrizione dettagliata del funzionamento	63
3	Ambiente di sviluppo	65
3.1	Framework Phonegap	65
3.1.1	Utilizzo di Phonegap	66
3.2	Software per visual editor	68
3.2.1	Intel XDK	69
3.3	Xcode	76
3.4	Eclipse	76
3.5	Architettura di Assistenza Interventi Tecnici	77
3.6	Database SQLite	77
3.6.1	Tabelle Assistenza Interventi Tecnici	79
4	Caratteristiche avanzate dell'App	83
4.1	Fotocamera/Galleria	83
4.2	Firma	86
4.3	Gestione multilingua	87
4.4	Rapportino di lavoro	88
4.5	Invio E-mail	90
4.6	Collegamento a Dropbox	91
4.6.1	Backup su Dropbox	92
5	Conclusioni e sviluppi futuri	95
	Bibliografia	96
	Indice analitico	98

Elenco delle figure

2.1	Use Case - Primo Accesso	10
2.2	Use Case - Home Page	11
2.3	Use Case - Azienda	12
2.4	Use Case - Clienti	13
2.5	Use Case - Materiali	14
2.6	Use Case - Inserisci chiamata	15
2.7	Use Case - Gestisci chiamata	16
2.8	Use Case - Gestione della chiamata	17
2.9	Screenshot - Dettaglio	18
2.10	Use case - Dettaglio	19
2.11	Screenshot - Prestazioni	20
2.12	Use case - Prestazioni	21
2.13	Screenshot - Dettaglio trasferte	22
2.14	Use case - Dettaglio trasferte	23
2.15	Screenshot - Materiali, prima della ricerca	24
2.16	Screenshot - Materiali, dopo la ricerca	25
2.17	Use case - Materiali	26
2.18	Screenshot - Informazioni cliente	27
2.19	Use case - Informazioni cliente	28
2.20	Screenshot - Chiusura	29
2.21	Use case - Chiusura	30
2.22	Use case - Storico	31
2.23	Screenshot - Impostazioni	32
2.24	Activity Diagram - Anagrafica Azienda	34
2.25	Activity Diagram - Tecnici	35
2.26	Activity Diagram - Prestazioni	36
2.27	Activity Diagram - Trasferte	37
2.28	Activity Diagram - Aliquota d'iva	38
2.29	Activity Diagram - Clienti	39
2.30	Activity Diagram - Materiali	40
2.31	Activity Diagram - Inserisci chiamata	41

2.32	Activity Diagram - Gestisci chiamata	43
2.33	Activity Diagram - Storico	44
2.34	Activity Diagram - Impostazioni	45
2.35	Sequence Diagram - Anagrafica Azienda	46
2.36	Sequence Diagram - Tecnici	47
2.37	Sequence Diagram - Prestazioni	48
2.38	Sequence Diagram - Trasferte	49
2.39	Sequence Diagram - Aliquota d'iva	50
2.40	Sequence Diagram - Condizioni generali di fornitura	51
2.41	Sequence Diagram - Clienti	52
2.42	Sequence Diagram - Materiali	53
2.43	Sequence Diagram - Inserisci Chiamata	54
2.44	Sequence Diagram - Gestisci Chiamata	55
2.45	Sequence Diagram - Chiamata	56
2.46	Sequence Diagram - Storico	57
2.47	Sequence Diagram - Impostazioni	58
2.48	Modello E-R	60
3.1	Architettura di un'app Phonegap	66
3.2	Screenshot - Login Intel XDK	69
3.3	Screenshot - Intel XDK, Prima pagina	70
3.4	Screenshot - Scelta framework	71
3.5	Screenshot - Schede Intel XDK	72
3.6	Screenshot - Ambiente di lavoro Intel XDK	72
3.7	Screenshot - Scheda emulatore	73
3.8	Screenshot - Scheda Test	74
3.9	Screenshot - Scheda Build	75
3.10	Architettura app	77
3.11	Struttura database Client/Server MySql	78
3.12	Struttura database Client/Server SQLite	78
3.13	Tabelle Assistenza Interventi Tecnici	80
4.1	Fotocamera nativa	84
4.2	Galleria	84
4.3	Scelta tra fotocamera e galleria	85
4.4	Esempio di firma	86
4.5	Impostazioni - Cambio lingua	87
4.6	Rapportino di intervento tecnico	88
4.7	Screenshot- Apertura e-mail su iOS	90
4.8	Pagina di autenticazione di Dropbox	92

Capitolo 1

Introduzione

La presente tesi è stata svolta durante un periodo di stage della durata di 6 mesi presso l'azienda Eurosystem2000 s.r.l di Modena.

L'azienda sviluppa direttamente numerosi software destinati alle aziende; in particolare ha iniziato da pochi anni lo sviluppo di applicazioni per smartphone, tra cui lo sviluppo di Assistenza Interventi Tecnici.

Lo scopo del progetto, era quello di realizzare un'applicazione mobile che potesse lavorare su diverse piattaforme (iOS, Android, Windows Mobile), al fine di offrire agli utenti la possibilità di gestire le chiamate di assistenza in modo autonomo. L'adozione di tale sistema dovrebbe offrire numerosi vantaggi:

- *Dematerializzazione* - eliminazione totale del cartaceo (precedentemente si procedeva alla stampa e compilazione manuale dell'intervento da parte dei tecnici e al successivo);
- inserimento a posteriori da parte dell'amministrazione;
- Possibilità di usufruire di funzionalità fornite da smartphone/tablet: visualizzazione della'agenda interventi, gestione delle chiamate di assistenza, firma del rapportino di intervento tecnico;
- rilascio al cliente in tempo reale i del rapporto di intervento tecnico. La realizzazione di questo progetto si pone quindi l'obiettivo di ottimizzare il flusso operativo legato all'assistenza, migliorando sensibilmente l'efficienza quotidiana.

Il primo periodo di stage sarà impiegato per la raccolta dei requisiti, per le scelte progettuali inerenti all'applicazione, per una breve analisi sulle applicazioni già esistenti e per lo studio di Phonegap, framework cross-platform mobile che consente di sviluppare delle applicazioni native attraverso l'utilizzo di tecnologie web quali HTML5, CSS e JavaScript. Successivamente, si passerà alla fase

di progettazione dell'app: la creazione delle pagine, la risoluzione dei problemi legati all'usabilità, alle prestazioni, e alla minimizzazione del consumo della batteria. L'ultimo periodo di stage sarà utilizzato per il test e collaudo dell'app e la correzione di relativi bug. Si dovrà inoltre effettuare un ampio studio in modo da ottimizzare la parte grafica e l'usabilità dell'app.

L'app sarà sviluppata in modo parallelo in due versioni: free e a pagamento. Quella che andremo a trattare in questa tesi è quella free. Questa versione si presenta con delle limitazioni e meno funzioni rispetto a quella a pagamento.

La tesi è organizzata nel seguente modo:

- Nel capitolo 2 verrà inizialmente illustrato il progetto dell'app prodotta utilizzando lo standard UML come linguaggio di modellazione, lo schema concettuale della base di dati secondo il modello Entity-Relationship e lo schema relazionale risultante;
- Nel capitolo 3 verranno presentati gli ambienti di sviluppo, ovvero la scelta dell'uso del framework Phonegap per la realizzazione di un'app ibrida rispetto ad una nativa o web app. Inoltre, verranno presentati i software utilizzati per la creazione dell'app, come Intel XDK (un visual editor) Xcode ed Eclipse, rispettivamente utilizzati per iOS ed Android. Infine il database utilizzato (SQLite) e le tabelle presenti;
- Nel capitolo 4 si descriveranno le caratteristiche avanzate, le quali arricchiscono l'app Assistenza Interventi Tecnici rendendola più completa e aumentandone le funzionalità; in particolare, verranno spiegati nel dettaglio:
 - Fotocamera/Galleria;
 - Firma;
 - Gestione multilingua;
 - Rapportino di lavoro;
 - Invio di e-mail;
 - Collegamento a Dropbox.
- Nell'ultimo capitolo verranno espone le conclusioni sull'attività svolta e gli sviluppi futuri relativi all'app a pagamento con ulteriori funzioni.

Capitolo 2

L'app ibrida Assistenza Interventi Tecnici

Quest'app è stata progettata allo scopo di fornire un supporto mobile ai tecnici che devono effettuare assistenze fuori dalla propria azienda, offrendo la possibilità di gestirle in modo facile e intuitivo. Inoltre, le funzionalità principali non necessitano di una connessione internet, quindi un tecnico può procedere con la gestione dell'assistenza anche in assenza di segnale. Esistono, tuttavia, due funzioni che richiedono una connessione: l'invio per mail del rapporto finale della prestazione e il backup su Dropbox (che analizzeremo in modo più approfondito nel prossimo capitolo).

2.1 User Case Diagram

Di seguito presenteremo una visione di alto livello di quello che Assistenza Interventi Tecnici sarà in grado di fare attraverso i diagrammi dei casi d'uso, strumenti utili per modellare, attraverso sequenze di azioni, le principali situazioni d'uso del software.

2.1.1 Gerarchia degli attori

Il software presenta una sola categoria di attore:

- **Tecnico:** esso corrisponde al dipendente di un'azienda. Ha la possibilità di accedere a tutte le funzionalità del software, dall'inserimento della chiamata di assistenza, fino alla sua chiusura.

2.1.2 Primo accesso

La prima volta che si effettua l'accesso viene presentato un elenco di Stati tra i quali scegliere per impostare la lingua. Tramite la conoscenza della lingua del dispositivo su cui è installato il software, esso provvede in modo automatico a pre-selezionare lo stato più rappresentativo per quella lingua (ad esempio se la lingua è lo spagnolo selezionerà la Spagna). In ogni caso, sarà possibile effettuare la scelta in base alle proprie esigenze.

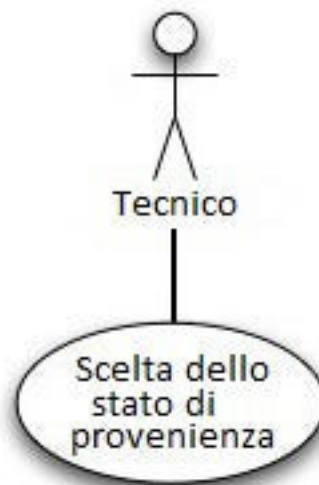


Figura 2.1: Use Case - Primo Accesso

2.1.3 Home page

La selezione di una nazione, all'interno dell'applicazione, è richiesta solo al primo accesso, e convoglierà alla visualizzazione della home page. Per i successivi accessi si presenta in automatico la schermata principale divisa in tre colonne e tre righe con icone esplicative delle azioni che il tecnico può effettuare. La scelta di una di esse produce una redirectione alla pagina optata.

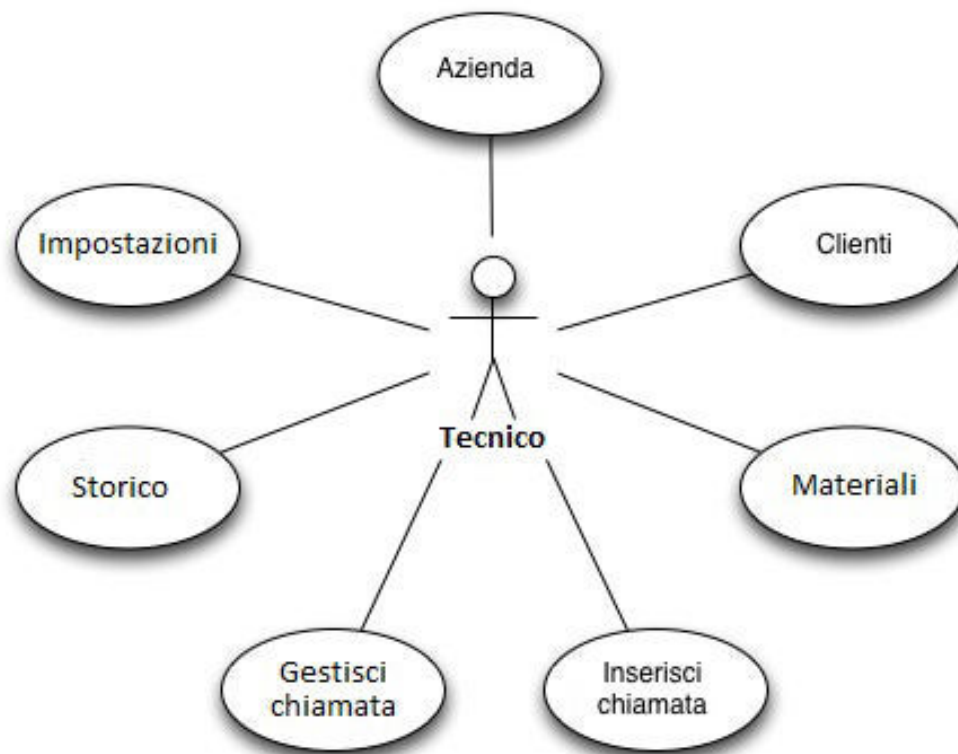


Figura 2.2: Use Case - Home Page

2.1.4 Azienda

Una volta premuto sull'icona si viene reindirizzati in una nuova pagina in cui è possibile gestire le proprietà principali dell'azienda.

In questa nuova pagina sono presenti cinque icone, ed ognuna di esse, una volta cliccato, aprirà una nuova pagina dove si potrà, a seconda dell'icona, gestire una particolare sezione della propria azienda, quali potrebbero essere l'anagrafica, i tecnici o le prestazioni che offre l'azienda.

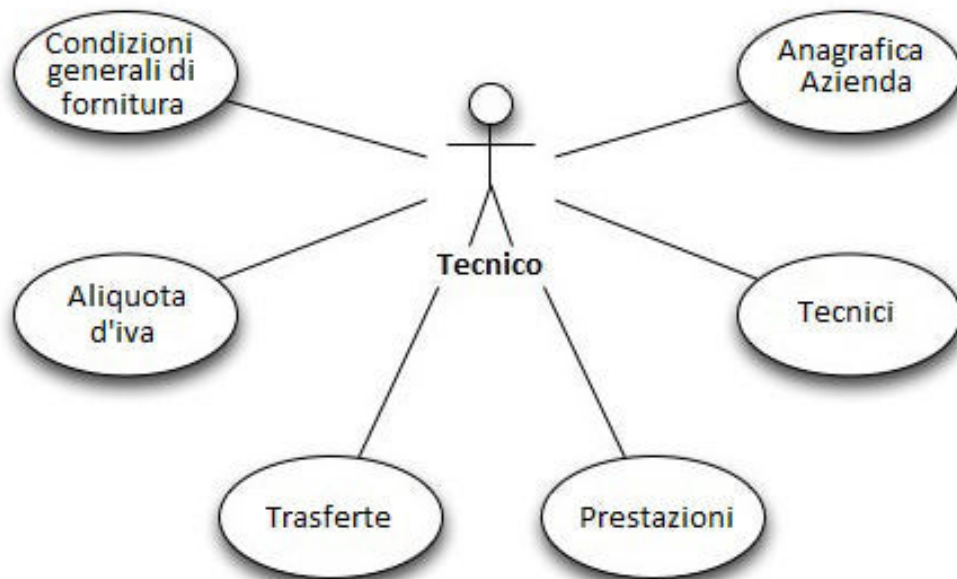


Figura 2.3: Use Case - Azienda

Andremo ora a descrivere in dettaglio le azioni delle icone:

- *Anagrafica Azienda*: Permette l'inserimento dei dati anagrafici dell'azienda (nome, indirizzo, telefono, fax ed e-mail) e l'aggiunta dei contatti dei dipendenti. Inoltre, sarà possibile inserire il logo dell'azienda stessa;
- *Tecnici*: Potranno essere inseriti tutti i tecnici dell'azienda con i rispettivi dati anagrafici e la firma;
- *Prestazioni*: qui verranno inserite le varie tipologie di prestazione che l'azienda fornisce, come per esempio manutenzione o installazione;
- *Trasferte*: in questa pagina si inseriscono i tariffari che l'azienda applica alle trasferte;

- *Aliquota d'iva*: qui si possono inserire tutte le aliquote che l'azienda applica alle varie prestazioni o ai materiali;
- *Condizioni generali di fornitura*: Permette di inserire il testo delle condizioni con cui l'azienda fornisce le sue prestazioni.

2.1.5 Clienti

Premendo sulla relativa icona verremo indirizzati alla pagina nella quale sarà possibile esaminare la lista di tutti i clienti a cui l'azienda offre assistenza, e dove avremo la possibilità di aggiungere nuovi clienti.

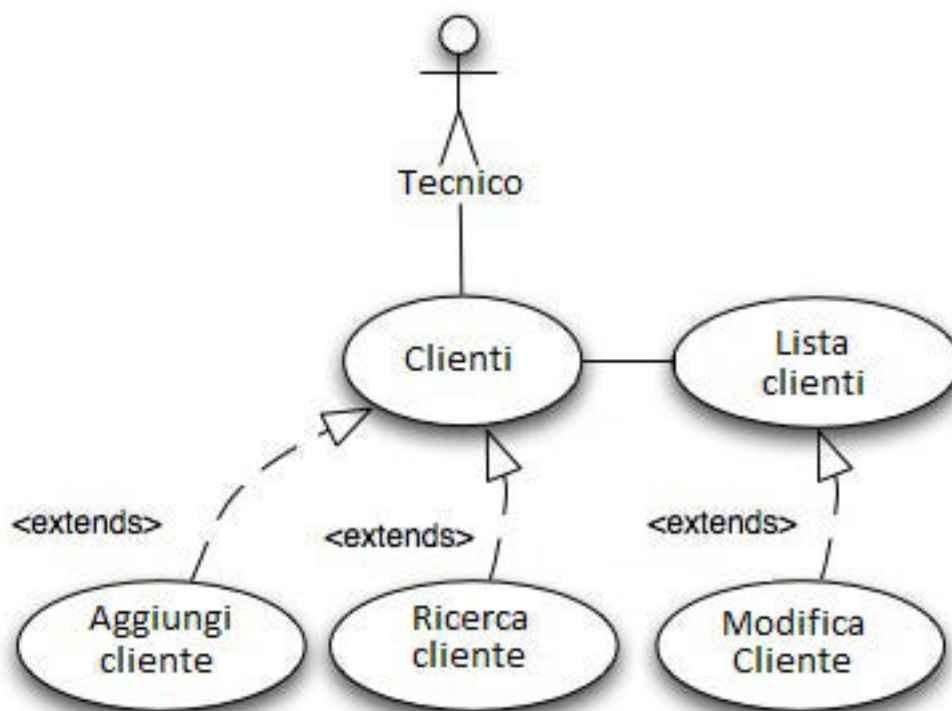


Figura 2.4: Use Case - Clienti

2.1.6 Materiali

All'apertura della corrispondente pagina visualizzeremo la lista dei materiali utilizzati dall'azienda e, come appurato in precedenza per la sezione Clienti, avremo la possibilità di aggiungere nuovi materiali o modificarne di già esistenti.

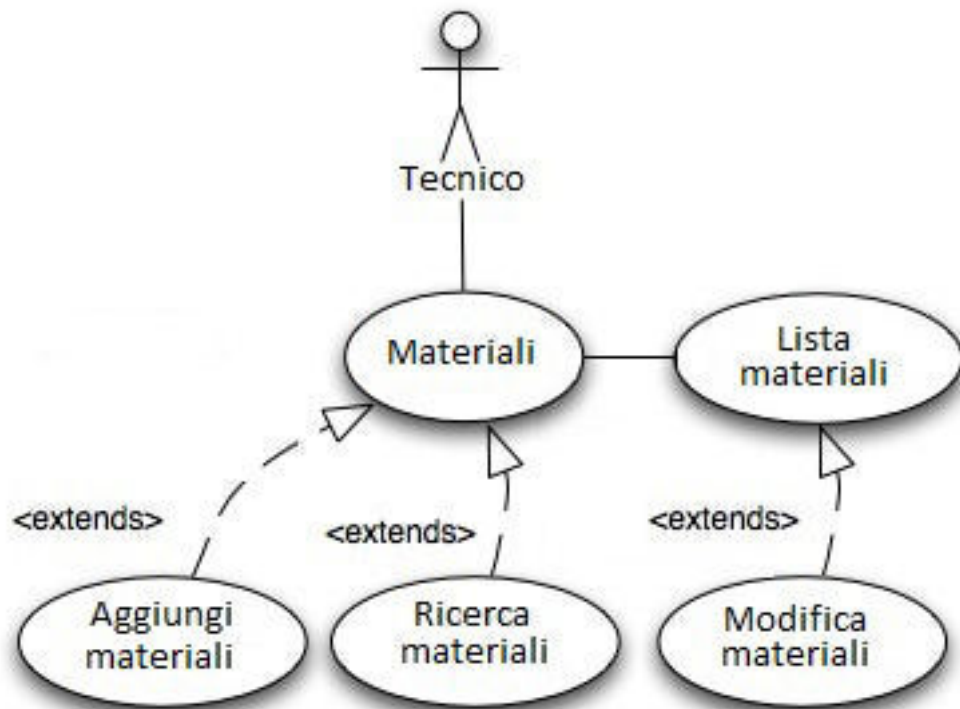


Figura 2.5: Use Case - Materiali

2.1.7 Inserisci chiamata

Da questa icona arriveremo alla pagina che consente l'inserimento della chiamata di assistenza. Qui verrà visualizzata la lista dei clienti (in modo analogo al punto 2.1.5) e selezionando il cliente desiderato saremo reindirizzati alla pagina in cui sarà possibile creare la chiamata

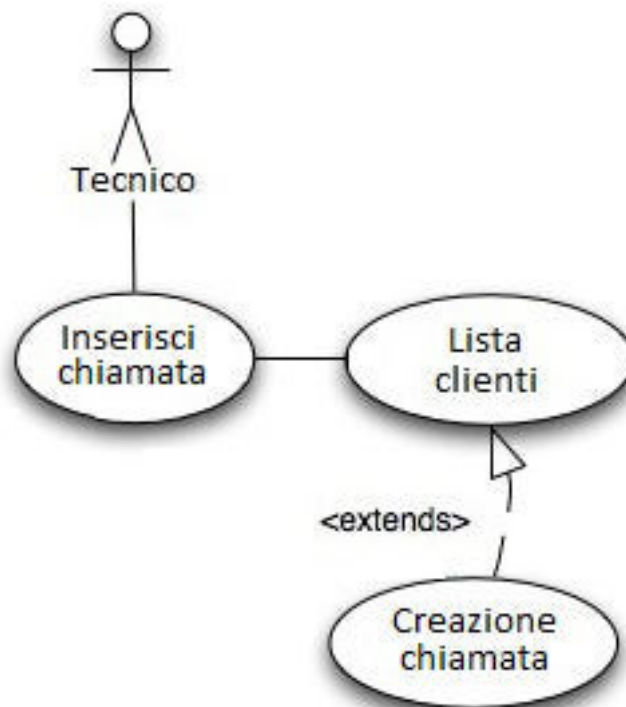


Figura 2.6: Use Case - Inserisci chiamata

2.1.8 Gestisci chiamata

Questa funzione permette di mostrare tutte le chiamate aperte raccolte per ogni cliente, e di visualizzare quelle del giorno, della settimana o del mese.

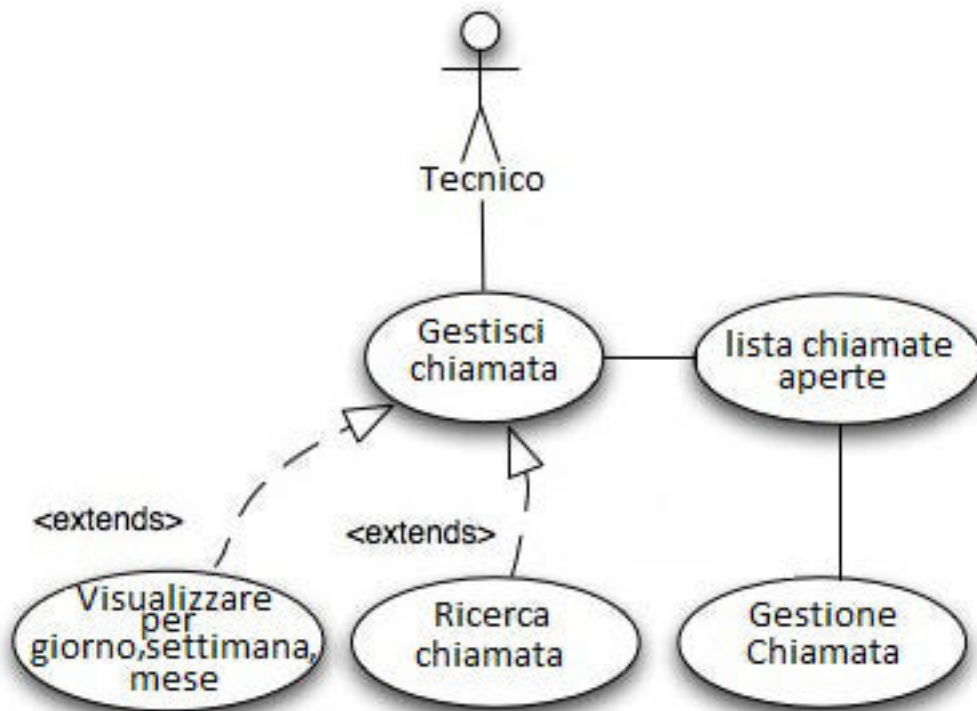


Figura 2.7: Use Case - Gestisci chiamata

Selezionando una chiamata verremo indirizzati nella sezione di gestione, la quale è suddivisa in sei pagine. Si partirà dalla prima pagina, *Dettaglio*, che presenta il riepilogo della chiamata e le informazioni del cliente ad essa associata. Si procede poi con le *Prestazioni*, ovvero dove andremo ad aggiungere le prestazioni effettuate durante la chiamata, seguite dal *Dettaglio trasferte*, dove è presente il listino delle trasferte (diritto di chiamata, costo al km, fascia del cliente) ed avremo la possibilità di selezionare ciò che l'assistenza richiede. Continuando si entra nella pagina dei *Materiali*, qui si andrà ad aggiungere i materiali utilizzati per la chiamata. Le ultime due pagine si riferiscono alle *Informazioni del cliente*, nome e firma del cliente, e *Chiusura*, creazione ed invio del rapportino finale in pdf. Non sarà comunque necessario procedere con l'ordine proposto sopra, ma si potrà navigare tra le varie pagine nel modo più gradito.

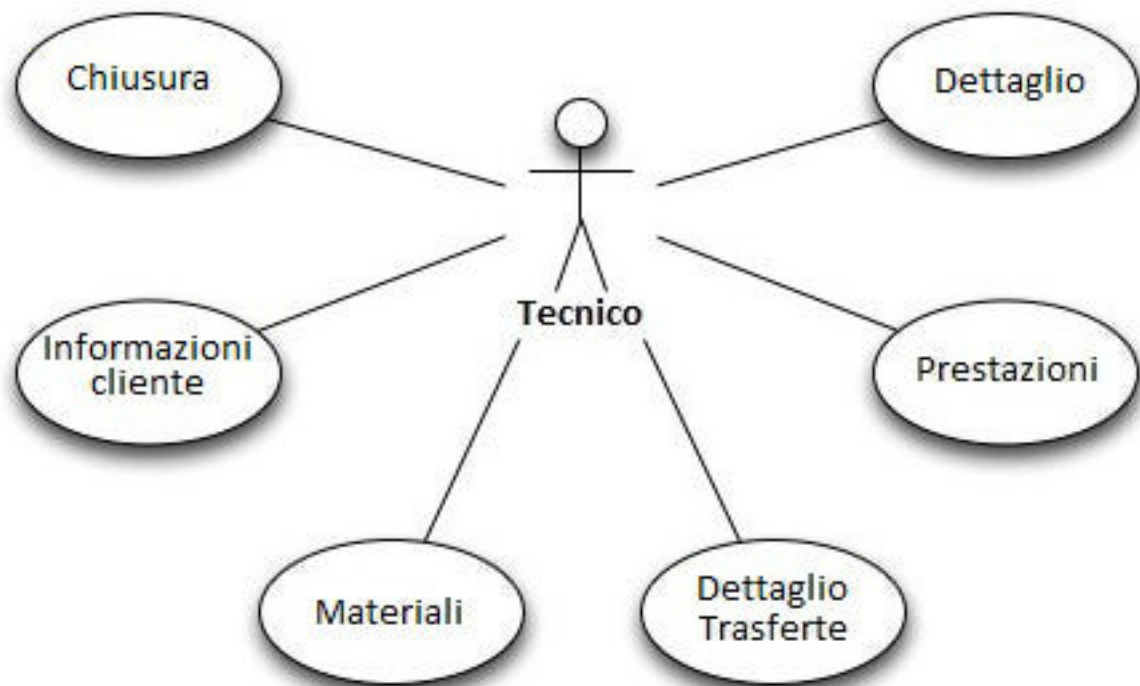


Figura 2.8: Use Case - Gestione della chiamata

Vedremo ora in dettaglio le sei pagine della gestione dell'assistenza.

Dettaglio

Mostra le informazioni generali della chiamata, offrendo anche una visione completa del cliente a cui si sta effettuando l'assistenza.



Figura 2.9: Screenshot - Dettaglio

Si può notare, dalla Figura 2.9, che sono presenti le informazioni base della chiamata (oggetto, descrizione ed eventuale nota) ed il cliente al quale si sta effettuando l'assistenza. Quest'ultimo verrà visualizzato in modo collassato per mostrare solo le informazioni essenziali che lo riguardano, ma premendo l'apposito bottone verde, tali informazioni potranno essere viste nel dettaglio con, in aggiunta, la possibilità di effettuare chiamate o inviare e-mail al referente aziendale, vedere la lista dei contatti dell'azienda, visitare il sito web o aprire la mappa per visualizzare la posizione dell'azienda. Per effettuare le chiamate sarà necessario inserire nel tag *href* elemento *tel* seguito dal numero telefonico, per le e-mail, invece, sarà necessario inserire l'elemento *mailto*, mentre per la mappa si dovrà inserire *maps* per iOS e *geo* per Android.

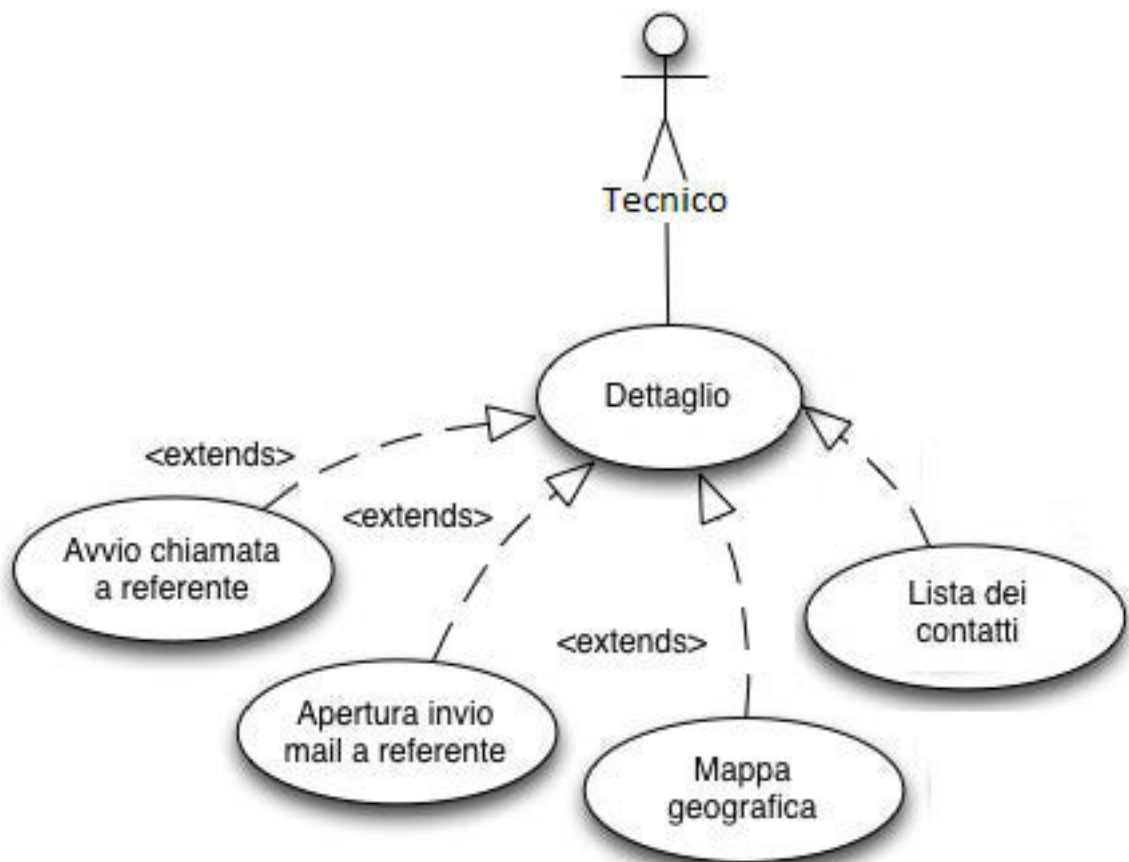


Figura 2.10: Use case - Dettaglio

Prestazioni

In questa pagina vedremo come il tecnico potrà inserire le prestazioni effettuate durante l'assistenza e salvarle.

Figura 2.11: Screenshot - Prestazioni

Questa è una pagina dinamica, ovvero le *select* con i relativi campi *option* e la quantità della prestazione vengono aggiornate in modo automatico una volta scelta la prestazione principale. A quest'ultima, difatti, sono associate più prestazioni secondarie ed ognuna di queste ha un suo metodo di calcolo della prestazione, identificabile in *tempo* o *quantità* (numero). Salvando, verrà prodotta un'area di riepilogo in fondo alla pagina ed identificato, con un numero crescente, le varie prestazioni memorizzate. Inoltre, i dati verranno registrati sul database locale in modo da essere reperiti in un secondo momento. Attraverso i bottoni numerati del riepilogo sarà possibile recuperare la prestazione salvata e caricarla nelle apposite sezioni.

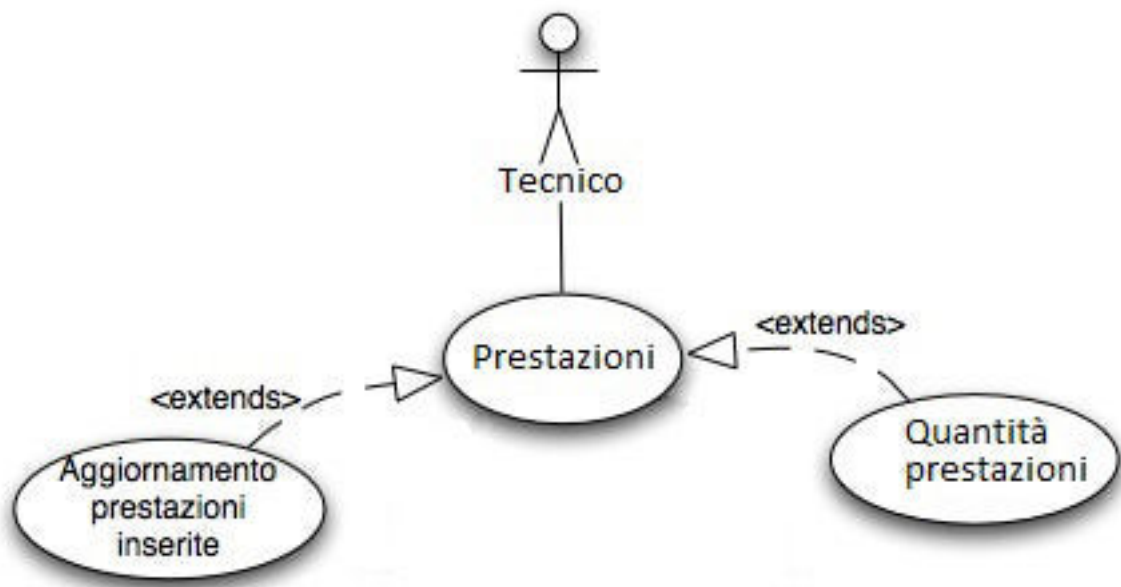


Figura 2.12: Use case - Prestazioni

Dettaglio trasferte

Questa è la pagina in cui potrà essere selezionato, se richiesto dall'assistenza, il costo della trasferta.

The screenshot shows a mobile application interface for managing travel expenses. The title bar at the top is labeled "Dettaglio trasferte" and includes a back arrow. Below the title bar, there are two main sections. The first section, titled "Data", contains a label "Data:" and a text input field displaying "16/mar/2015". The second section, titled "Trasferte", contains three toggleable options, each with a grey square checkbox: "Diritto di chiamata (50,00 €)", "Fascia trasferta", and "Km da fatturare". At the bottom of the screen, there are navigation arrows and a series of five dots, with the second dot from the left being green, indicating the current screen in a sequence.

Figura 2.13: Screenshot - Dettaglio trasferte

Qui si potrà scegliere se la trasferta è soggetta al diritto di chiamata (quota fissa), la fascia cui appartiene il cliente ed inoltre i chilometri percorsi (ai quali sarà applicato il costo al chilometro registrato nella sezione *Trasferte* dell'Azienda).

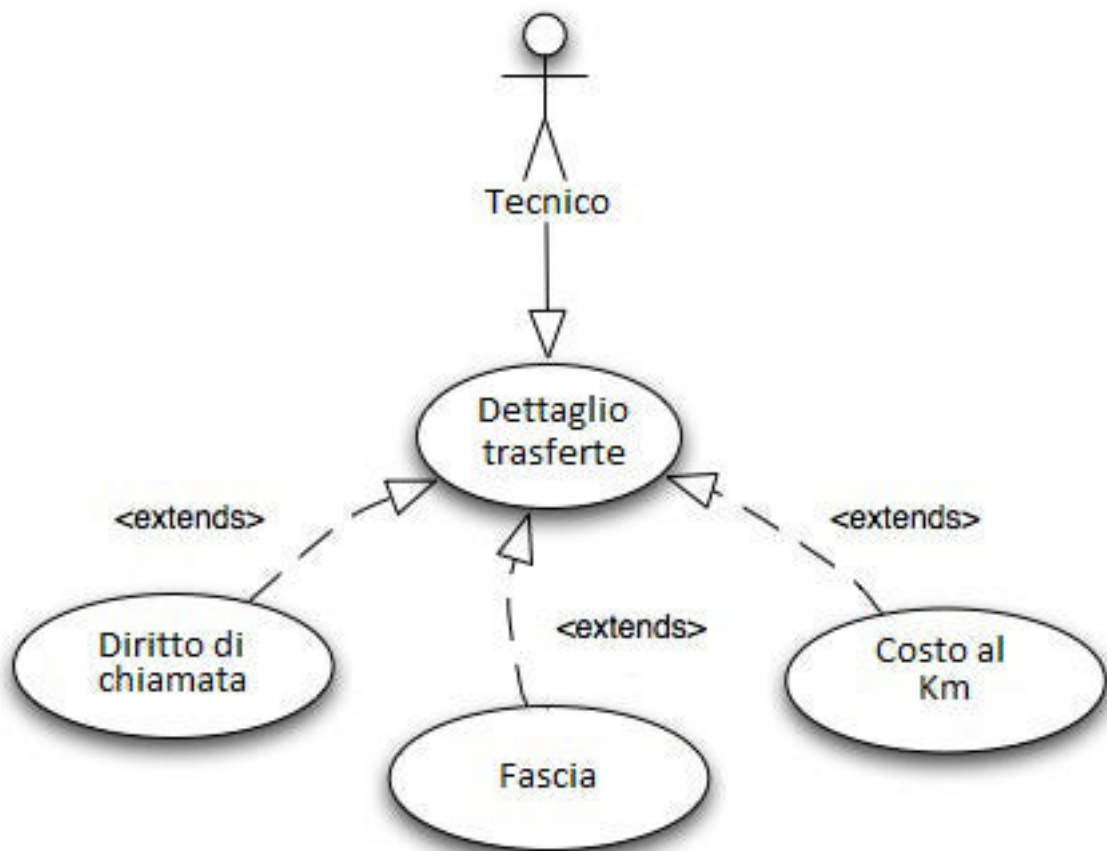


Figura 2.14: Use case - Dettaglio trasferte

Materiali

In questa pagina il tecnico inserisce tutti i materiali utilizzati per effettuare l'assistenza.



Figura 2.15: Screenshot - Materiali, prima della ricerca

Nella figura 2.15 si può vedere come la pagina si mostri prima della ricerca del materiale. Infatti anche questa, come quella delle prestazioni, è una pagina dinamica, ovvero il tecnico una volta cliccato sull'icona della ricerca, verrà indirizzato a una nuova pagina con l'elenco dei materiali corrispondenti alla ricerca effettuata (tramite codice o nome), scelto il materiale verrà reindirizzato alla pagina precedente.



Figura 2.16: Screenshot - Materiali, dopo la ricerca

La figura 2.16 mostra la pagina dopo aver selezionato il materiale, ora si potrà inserire la quantità del materiale (a seconda della sua unità di misura) e, se necessario, delle note sul materiale stesso. Infine, il tutto potrà essere registrato nell'assistenza, creando un'area di riepilogo del tutto analoga a quella prodotta alla pagina due, ovvero quella delle prestazioni.

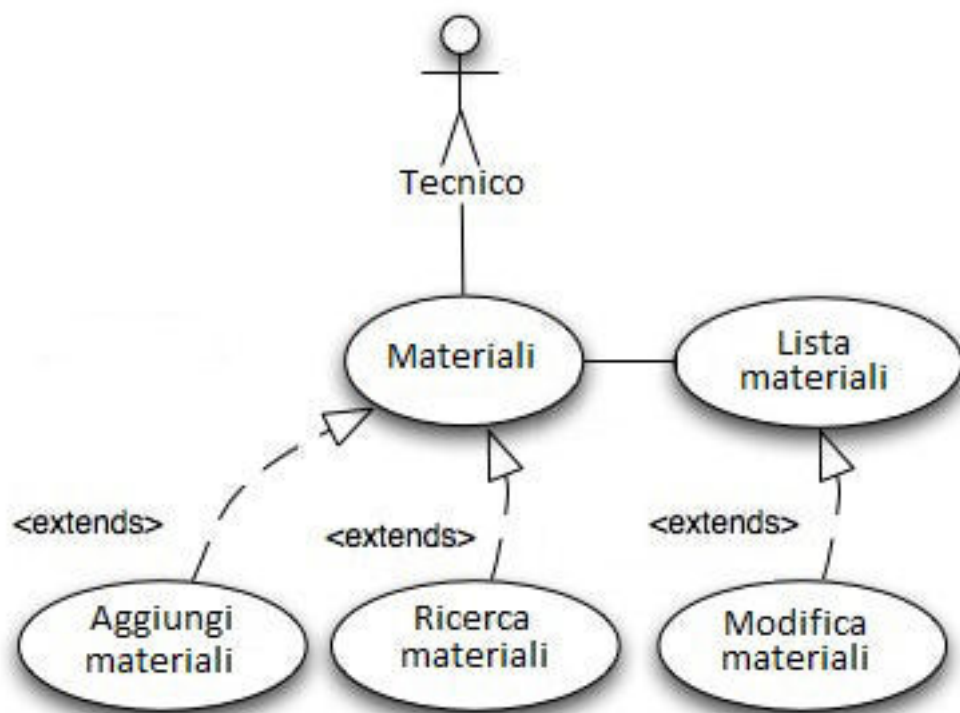


Figura 2.17: Use case - Materiali

Informazioni cliente

In questa pagina sarà possibile inserire il nome e cognome del contatto del cliente a cui si sta facendo assistenza e la firma di esso che verrà utilizzata sul rapportino finale.

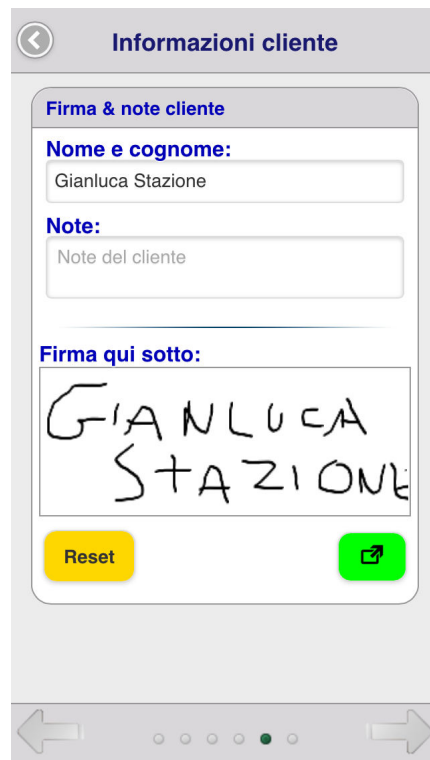


Figura 2.18: Screenshot - Informazioni cliente

La pagina è divisa in tre sezioni, dove le prime due sono dei campi *input* in cui si inserisce il nome e cognome della persona a cui si sta effettuando l'assistenza, e le relative note. L'ultima sezione, invece, è uno spazio in cui la persona può effettuare la sua firma, e questa verrà poi applicata e stampata sul rapportino finale. Essa ha lo stesso valore legale di una firma tradizionale effettuata a penna.

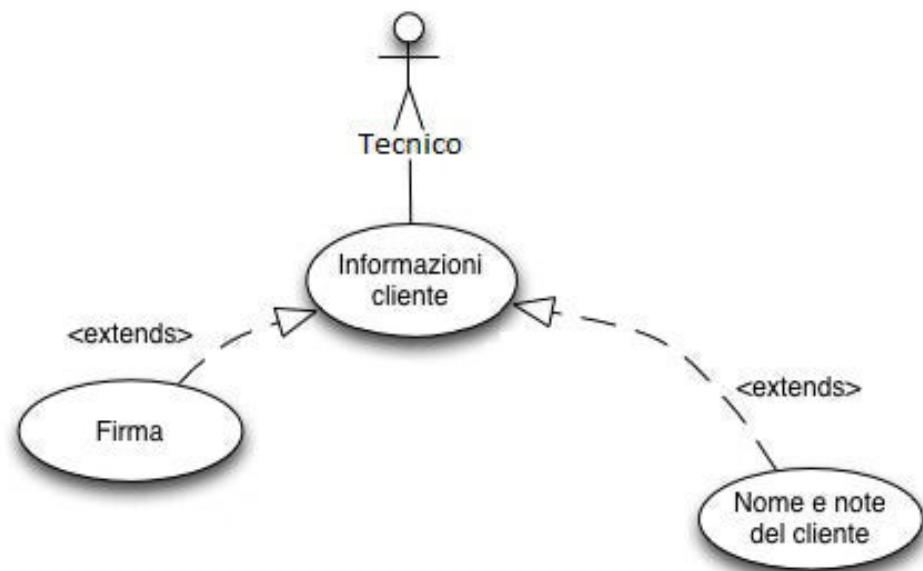


Figura 2.19: Use case - Informazioni cliente

Chiusura

Questa è l'ultima pagina di gestione della chiamata, qui si può procedere con la chiusura dell'assistenza.

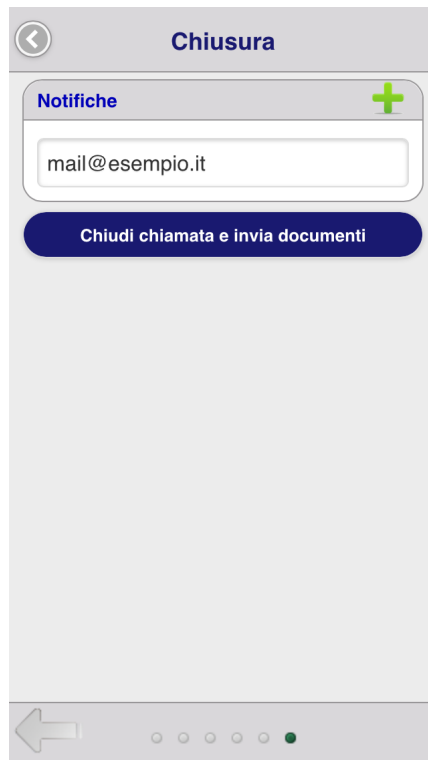


Figura 2.20: Screenshot - Chiusura

In questa pagina si possono inserire tutte le e-mail a cui verrà inviato il rapportino finale in pdf, il quale verrà creato una volta cliccato il pulsante di chiusura ed inviato come allegato agli indirizzi e-mail debitamente indicati.

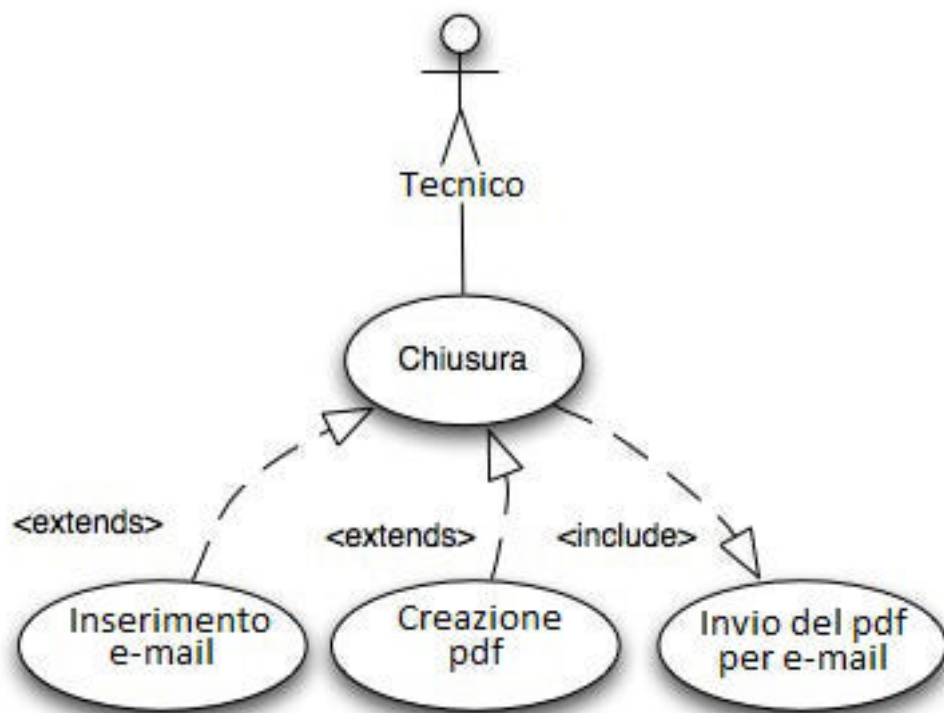


Figura 2.21: Use case - Chiusura

2.1.9 Storico

Selezionando quest'icona si potrà accedere alla sezione dello storico delle assistenze effettuate. Come nel caso della Gestione delle chiamate (Capitolo 2.1.8), anche qui sarà possibile visualizzare le chiamate chiuse raccolte per giorno, settimana o mese.

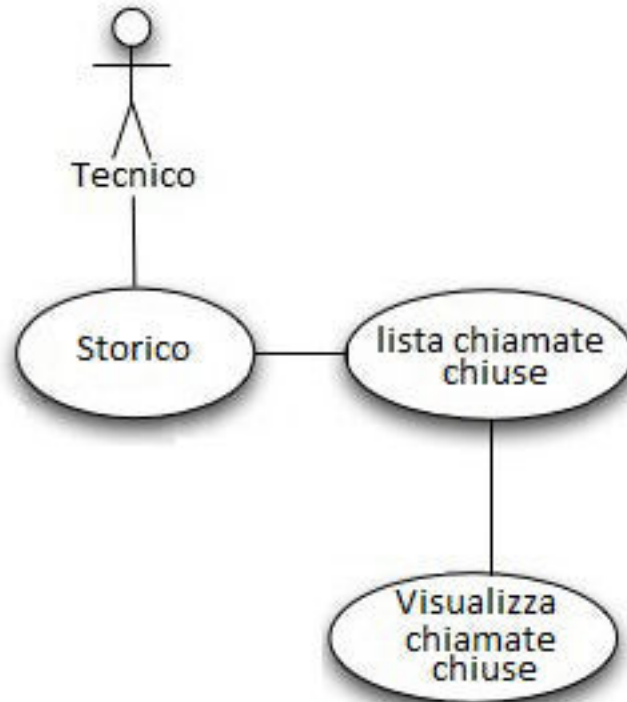


Figura 2.22: Use case - Storico

2.1.10 Impostazioni

L'ultima icona che andremo a descrivere ci permette di accedere alle impostazioni generali dell'applicazione.

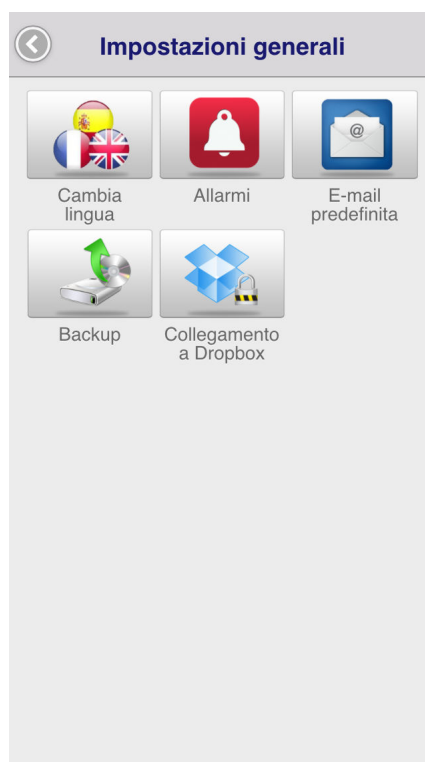


Figura 2.23: Screenshot - Impostazioni

Come si può facilmente osservare dalla Figura 2.23, questa sezione è formata da cinque pagine distinte, che andremo ora a definire. La prima icona *Cambia lingua* permette, una volta entrati, di modificare la lingua che si era impostata alla prima installazione tra le varie presenti: Inglese, Italiano, Spagnolo, Tedesco, Portoghese, Francese. La seconda, denominata *Allarme*, permette di impostare il numero di giorni per cui sarà necessario evidenziare le assistenze in scadenza, ovvero nella sezione “gestisci chiamata” le assistenze in scadenza saranno bordate di rosso. *E-mail predefinita*, invece, consente di inserire un'e-mail predefinita (e l'oggetto della stessa) per inviare il rapportino in pdf della chiusura dell'assistenza. Le ultime due, ovvero *Backup* e *Collegamento a Dropbox*, sono strettamente legate tra di loro, infatti, per effettuare il backup di tutti i database su Dropbox è necessario prima collegare l'applicazione al proprio account di Dropbox.

2.2 Activity Diagram

L'activity Diagram è un diagramma definito all'interno dello Unified Modeling Language (UML) che definisce le attività da svolgere per realizzare una data funzionalità. Più in dettaglio, un activity diagram definisce una serie di attività o flusso, anche in termini di relazioni tra le attività, i responsabili per le singole attività e i punti di decisione. Risultano utili anche per approfondire i casi d'uso presentati precedentemente.

2.2.1 Anagrafica Azienda

Una volta aperta la pagina potremo trovarci in due situazioni differenti: se questo risultasse essere il nostro primo accesso nessun dato aziendale sarebbe stato ancora registrato, pertanto si dovrebbe procedere al relativo inserimento. Invece, nel caso in cui le informazioni dell'azienda fossero già state inserite (in quanto non fosse la prima volta che visitiamo la pagina) verrebbero visualizzate correttamente. Tramite l'utilizzo di alcuni pulsanti potranno essere inseriti il logo aziendale e i contatti dell'azienda stessa e sarà possibile modificare dati inseriti precedentemente.

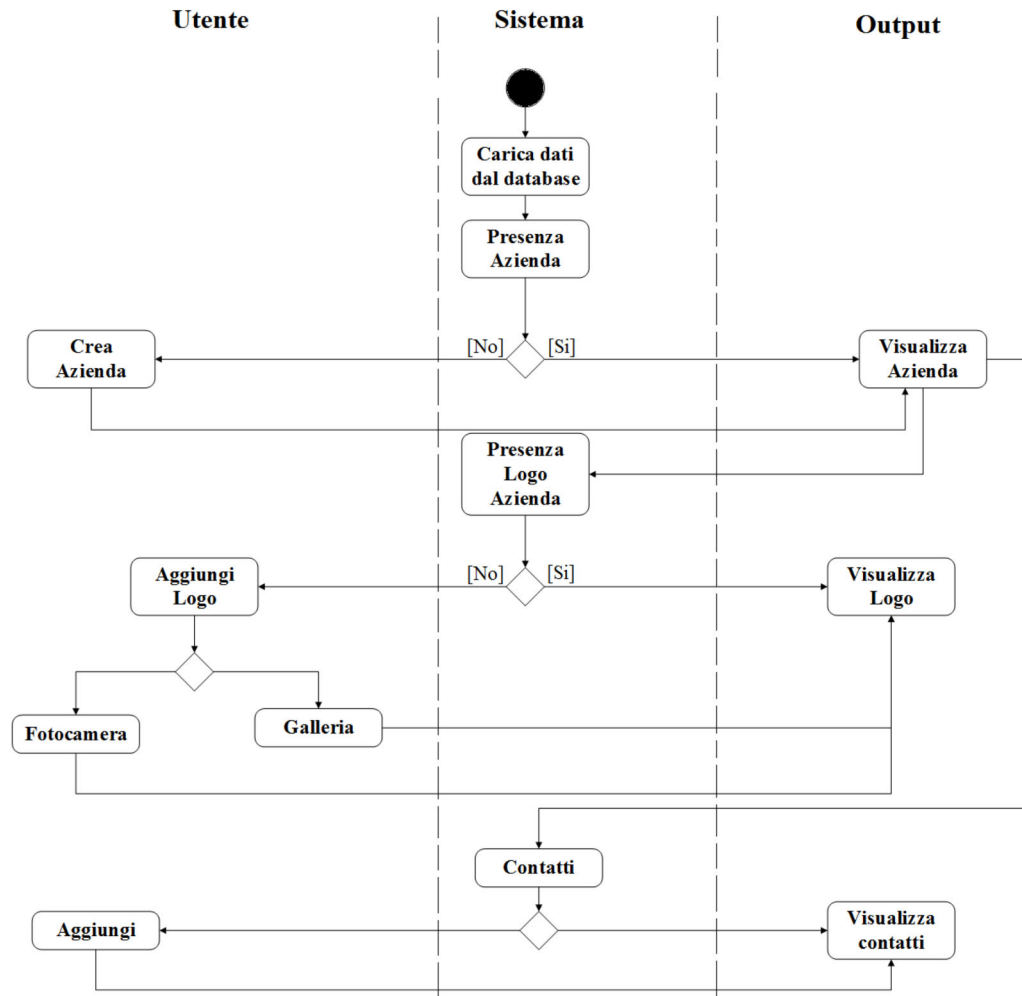


Figura 2.24: Activity Diagram - Anagrafica Azienda

2.2.2 Tecnici

All'apertura di questa pagina il sistema interrogherà il database per ricercare i tecnici registrati. In caso di assenza, il database non ritornerà nulla. Una volta caricati e visualizzati sarà possibile aggiungere un nuovo tecnico, oppure modificare o eliminare uno già esistente.

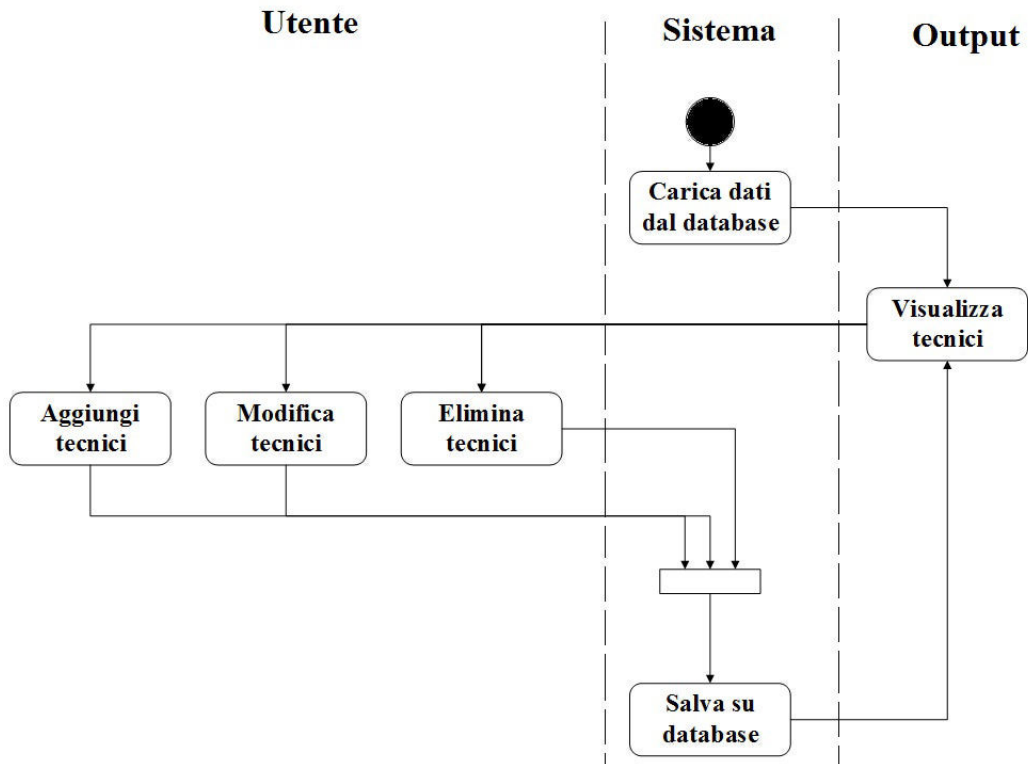


Figura 2.25: Activity Diagram - Tecnici

2.2.3 Prestazioni

Il sistema carica dal database le prestazioni salvate in precedenza. Esse serviranno a delineare le prestazioni comuni che offre l'azienda e verranno richiamate durante l'assistenza.

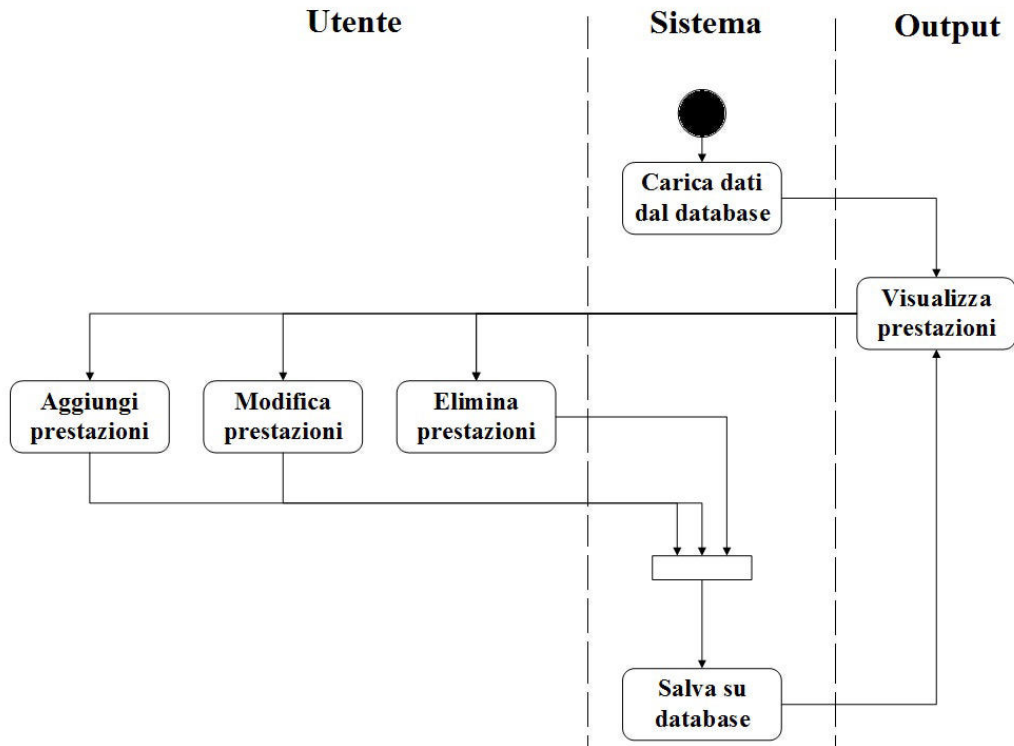


Figura 2.26: Activity Diagram - Prestazioni

2.2.4 Trasferte

Pagina in cui si caricheranno i costi che l'azienda predispone per le trasferte, interrogando il database.

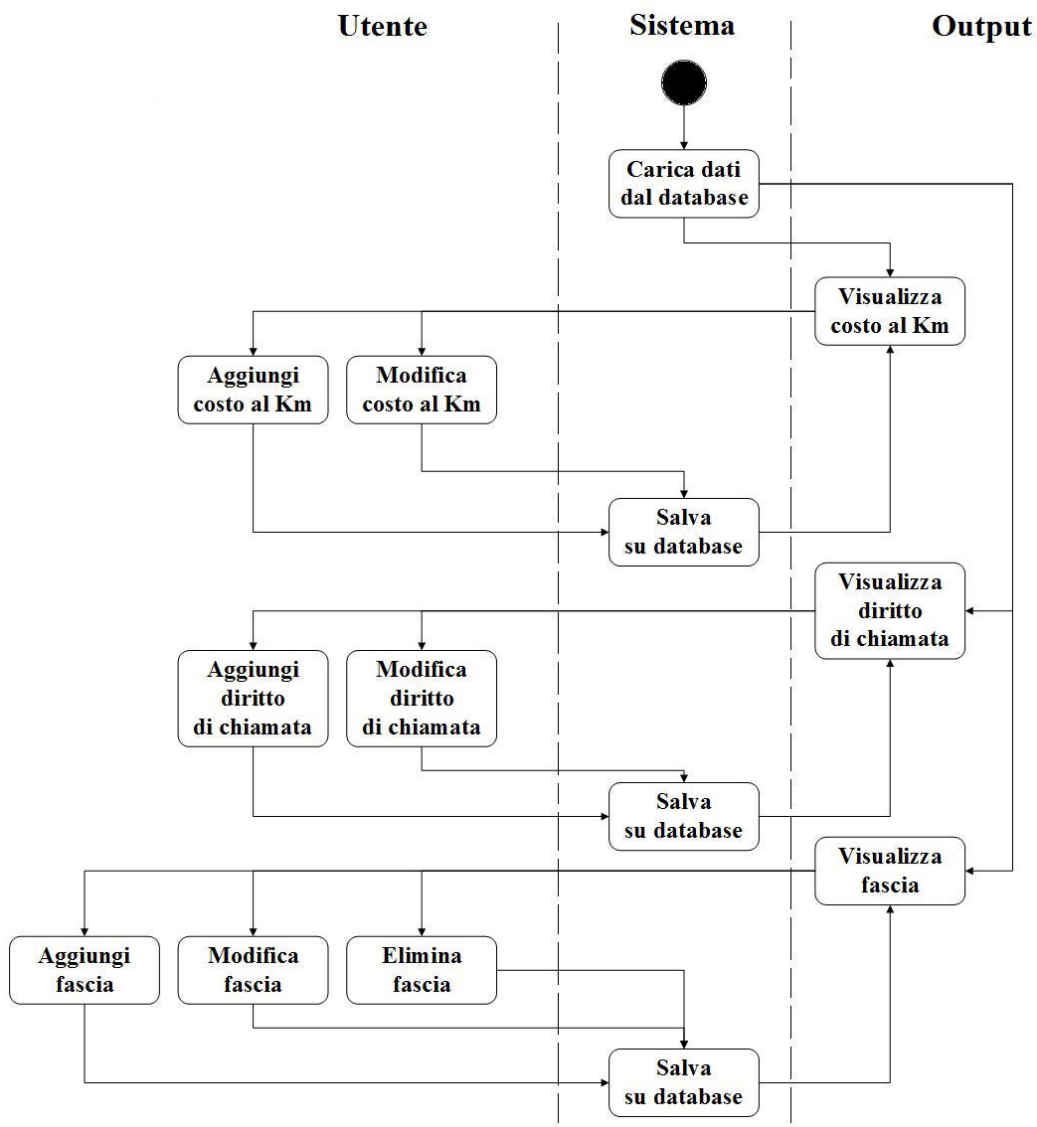


Figura 2.27: Activity Diagram - Trasferte

2.2.5 Aliquota d'iva

Questa è semplicemente la pagina in cui si potranno salvare diverse aliquote d'iva che l'azienda applica rispetto a prestazioni, materiali o altro.

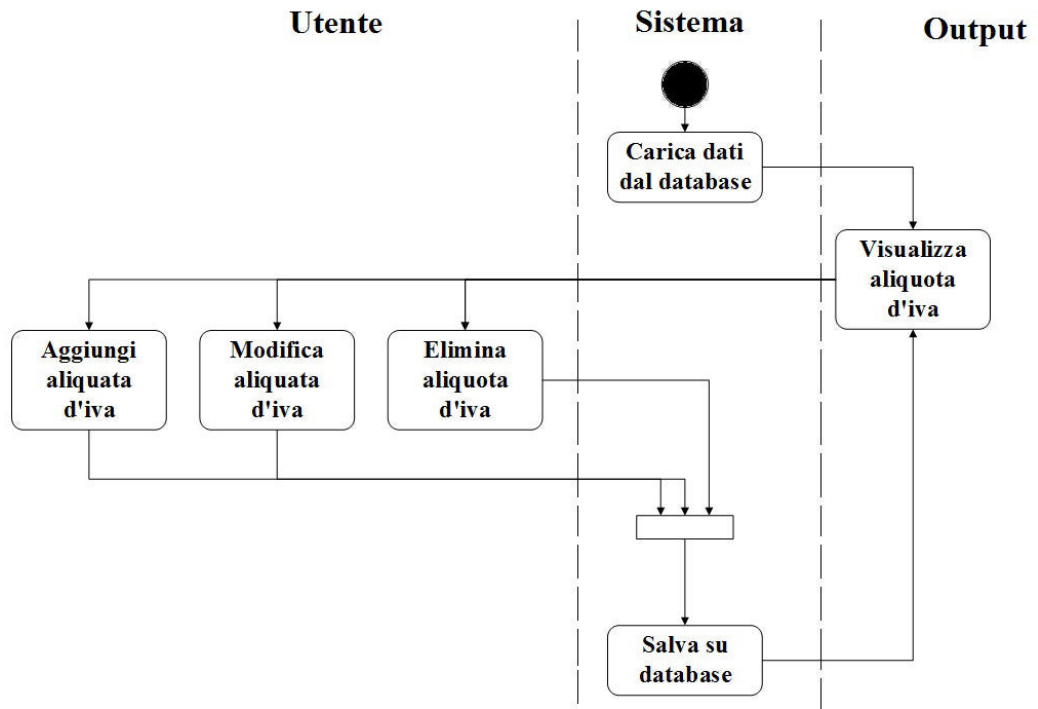


Figura 2.28: Activity Diagram - Aliquota d'iva

2.2.6 Clienti

All'apertura della pagina, il sistema interroga il database e carica i clienti salvati. Dopodiché potrà essere inserito un nuovo cliente. Selezionando uno di questi si verrà indirizzati alla corrispondente pagina riepilogativa dove sarà possibile modificare i dati del cliente in questione, aggiungere i suoi contatti oppure eliminarlo.

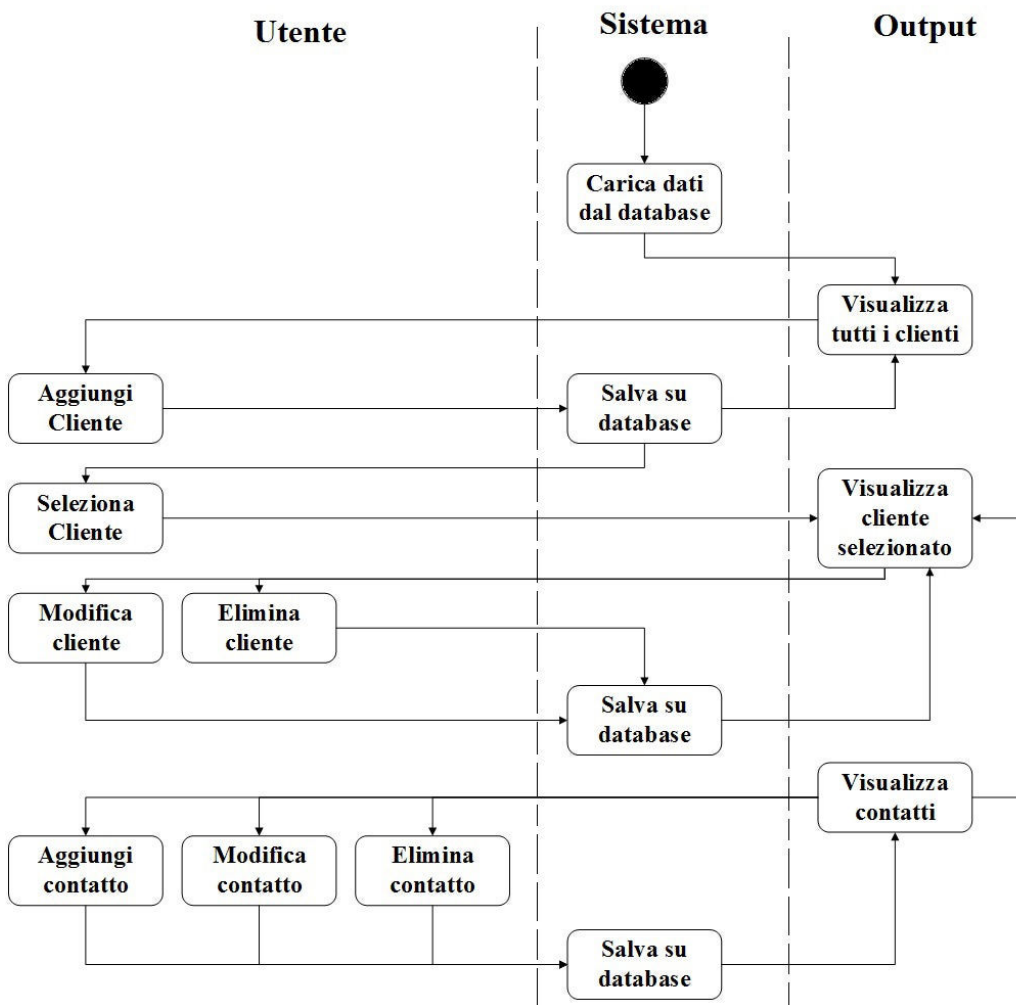


Figura 2.29: Activity Diagram - Clienti

2.2.7 Materiali

Questa pagina ha un funzionamento simile alla pagina dei Tecnici (Capitolo 2.2.2). Il sistema interroga il database per caricare i materiali, offrendo la possibilità di aggiungerne di nuovi e di modificare o eliminare materiali già esistenti.

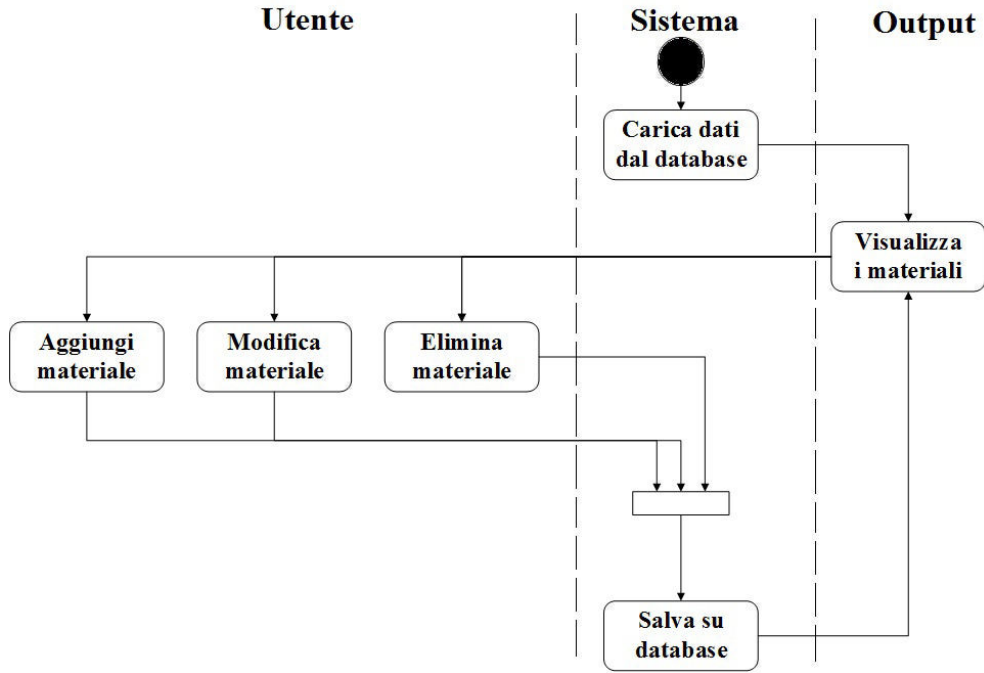


Figura 2.30: Activity Diagram - Materiali

2.2.8 Inserisci chiamata

All'apertura di questa pagina il sistema carica i clienti e li visualizza in ordine alfabetico. Selezionando il cliente raggiungeremo la pagina in cui verrà creata la chiamata inserendo i primi dati della chiamata di assistenza. Una volta immessi, si potrà scegliere se gestire subito la chiamata o, in caso contrario, si verrà indirizzati alla home page.

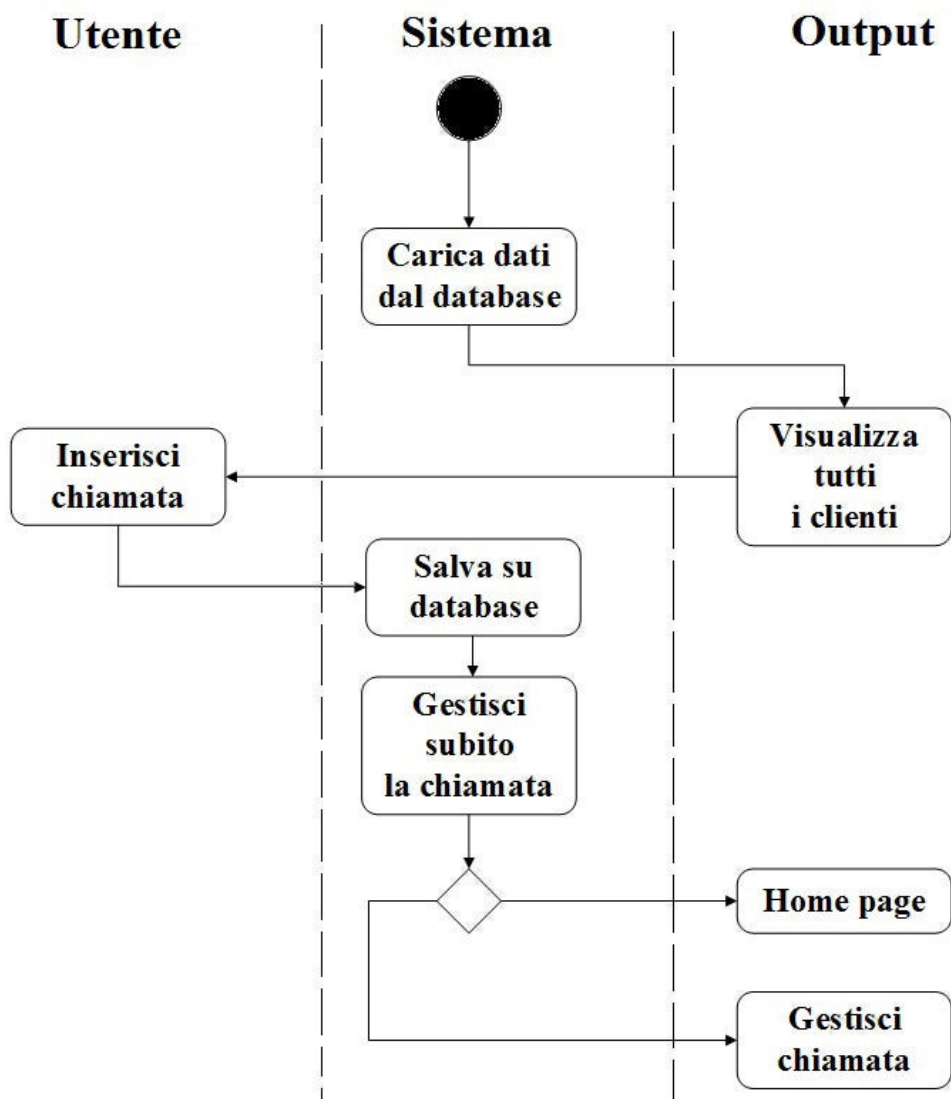


Figura 2.31: Activity Diagram - Inserisci chiamata

2.2.9 Gestisci chiamata

Appena si accede a questa pagina, il sistema caricherà le chiamate del giorno corrente tramite i filtri giorno, settimana e mese. Pertanto, il sistema interrogherà il database a seconda del filtro di visualizzazione scelto. Dopodiché, per ogni chiamata potranno essere modificati i dati iniziali oppure si potrà scegliere di gestire la chiamata stessa.

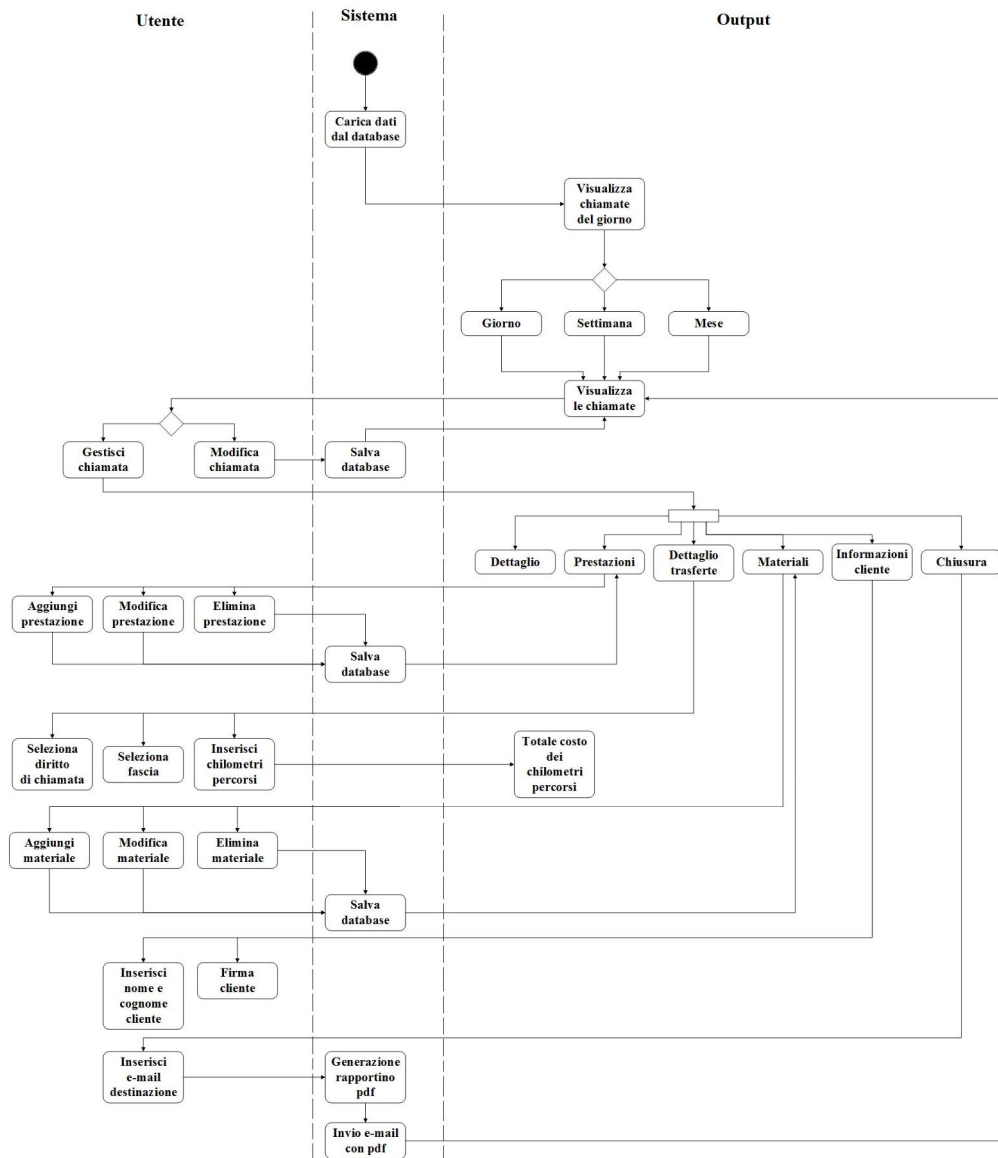


Figura 2.32: Activity Diagram - Gestisci chiamata

2.2.10 Storico

Questa pagina, utilizza gli stessi filtri visti nel capitolo precedente, ma a differenza di prima qui verranno raccolte tutte le chiamate chiuse.

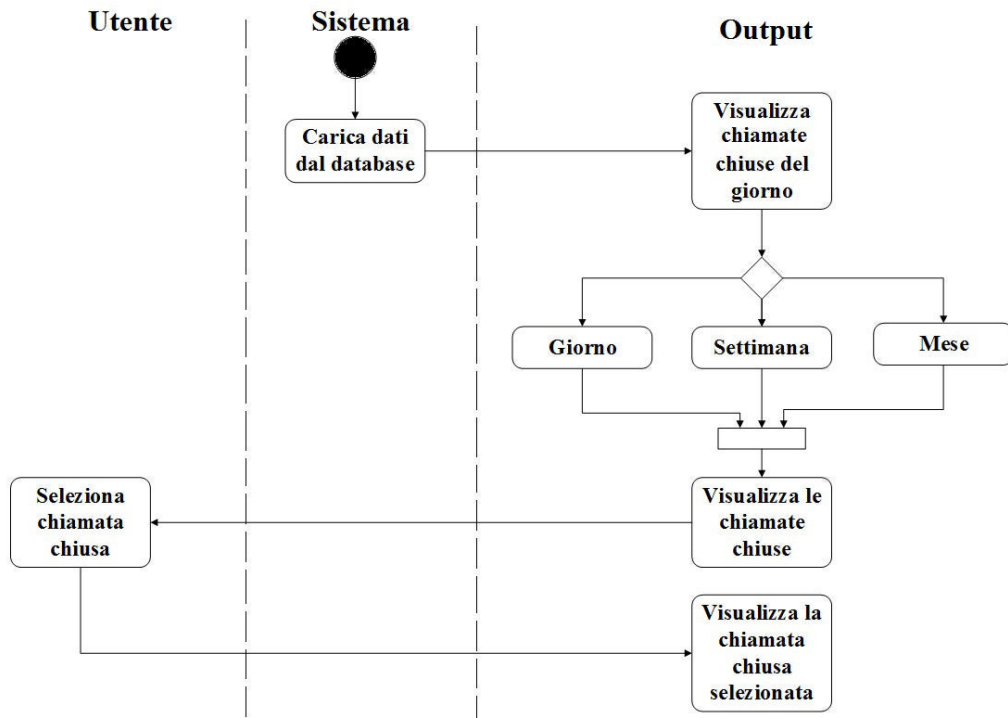


Figura 2.33: Activity Diagram - Storico

2.2.11 Impostazioni

All'apertura della pagina, il sistema interroga il database e carica i clienti salvati. Dopodiché potrà essere inserito un nuovo cliente. Selezionando uno di questi si verrà indirizzati alla corrispettiva pagina riepilogativa dove sarà possibile modificare i dati del cliente in questione, aggiungere i suoi contatti oppure eliminarlo.

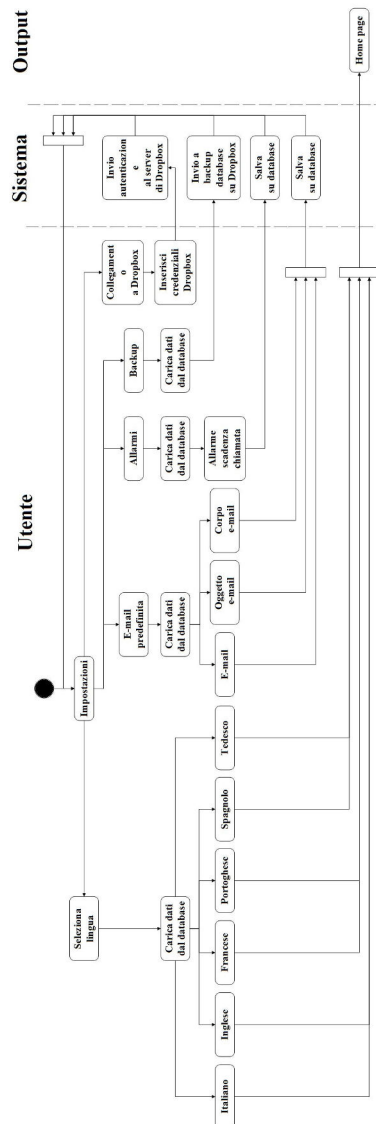


Figura 2.34: Activity Diagram - Impostazioni

2.3 Sequence Diagram

Un sequence diagram è un diagramma utilizzato per descrivere uno scenario. Uno scenario è una determinata sequenza di azioni in cui tutte le scelte sono state già effettuate; in pratica nel diagramma non compaiono scelte, né flussi alternativi. Descrive le relazioni che intercorrono, in termini di messaggi, tra attori, oggetti o entità del sistema che si sta rappresentando.

2.3.1 Anagrafica Azienda

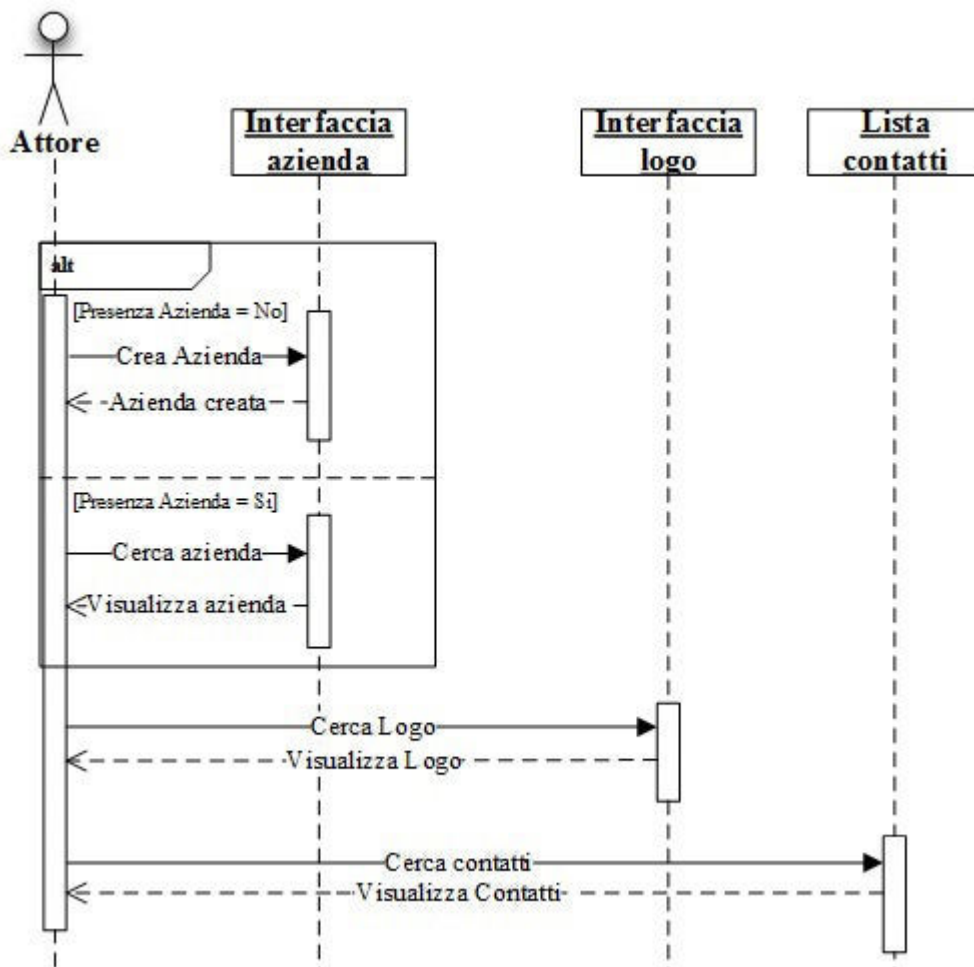


Figura 2.35: Sequence Diagram - Anagrafica Azienda

2.3.2 Tecnici

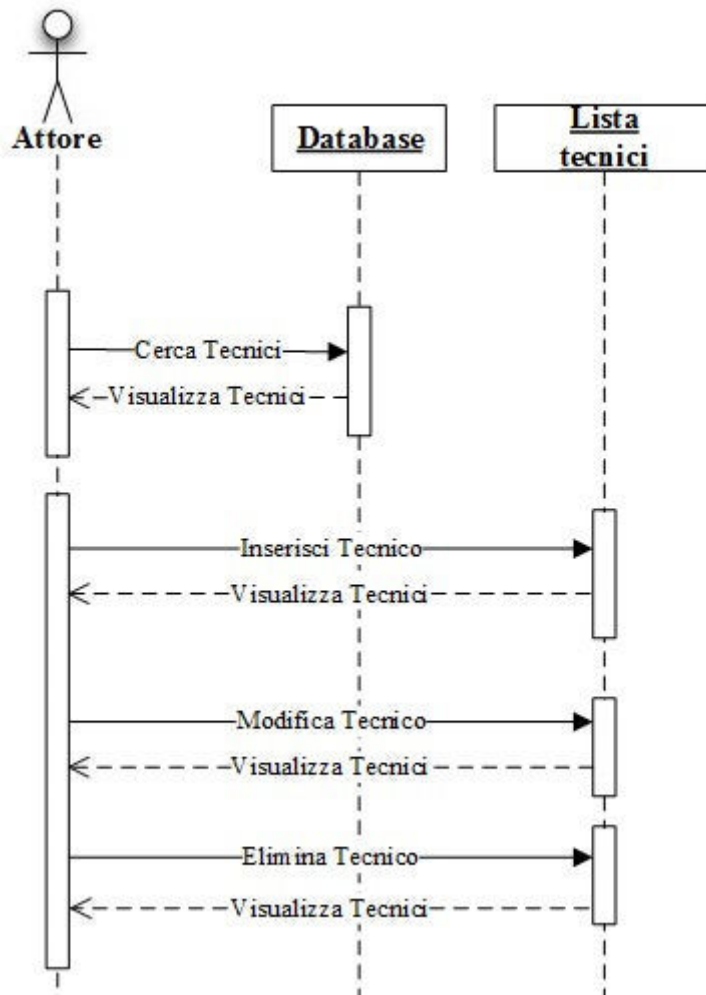


Figura 2.36: Sequence Diagram - Tecnici

2.3.3 Prestazioni

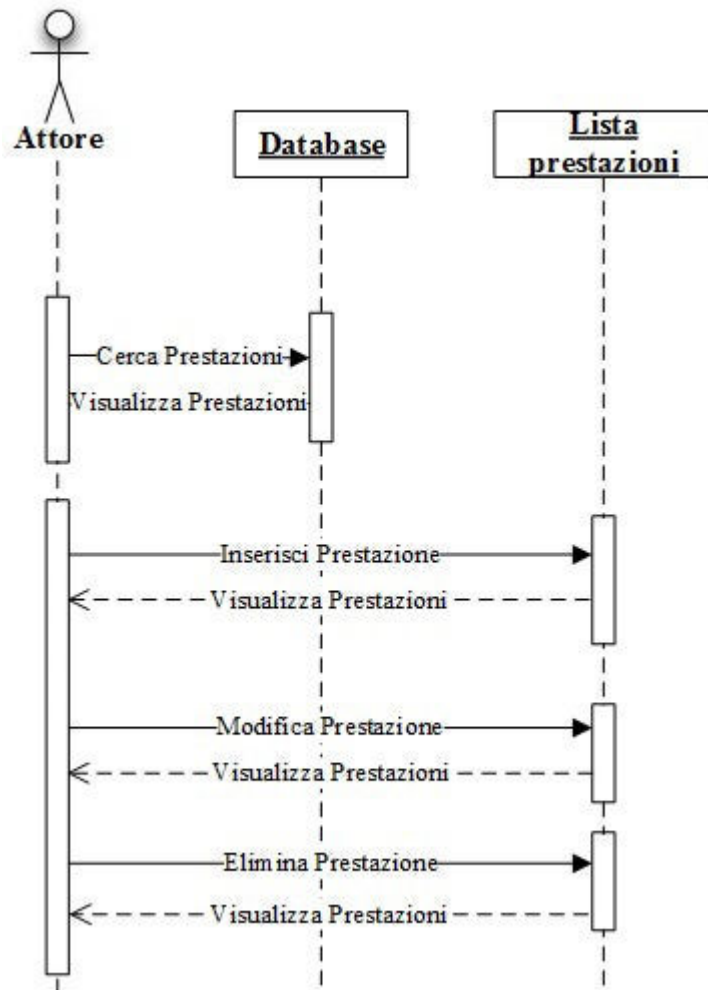


Figura 2.37: Sequence Diagram - Prestazioni

2.3.4 Trasferte

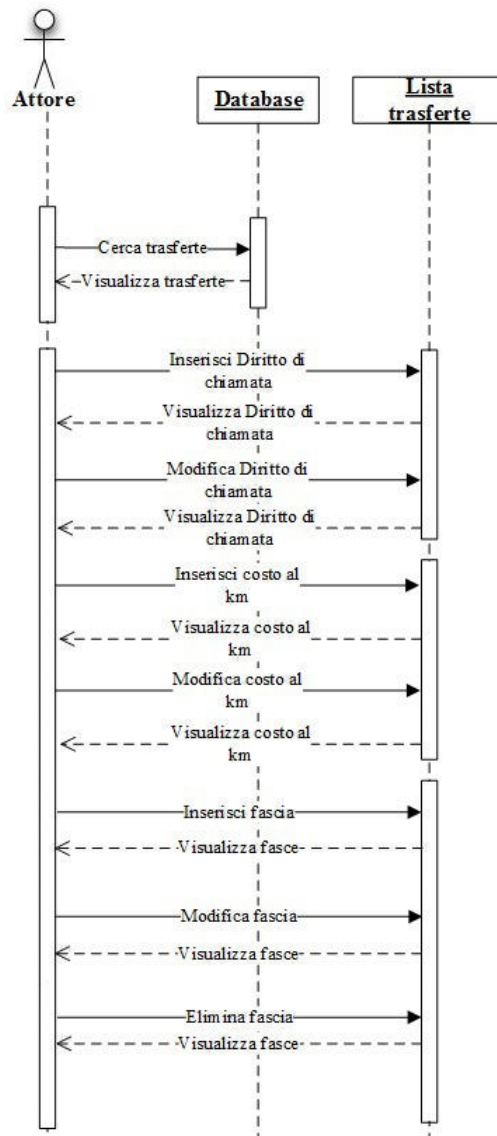


Figura 2.38: Sequence Diagram - Trasferte

2.3.5 Aliquota d'iva

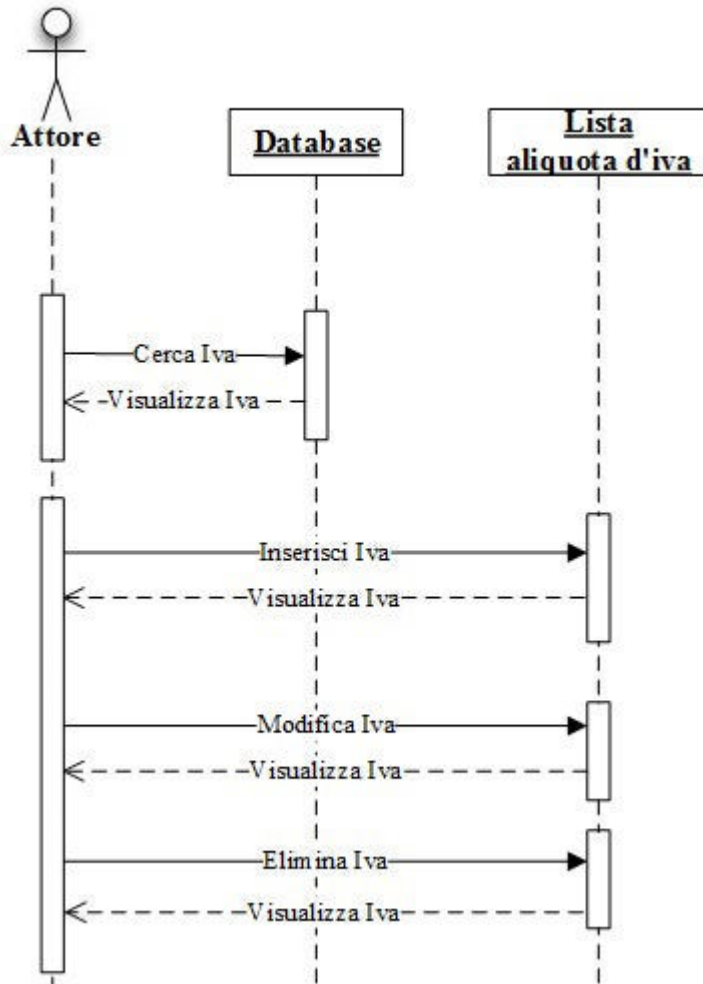


Figura 2.39: Sequence Diagram - Aliquota d'iva

2.3.6 Condizioni generali di fornitura

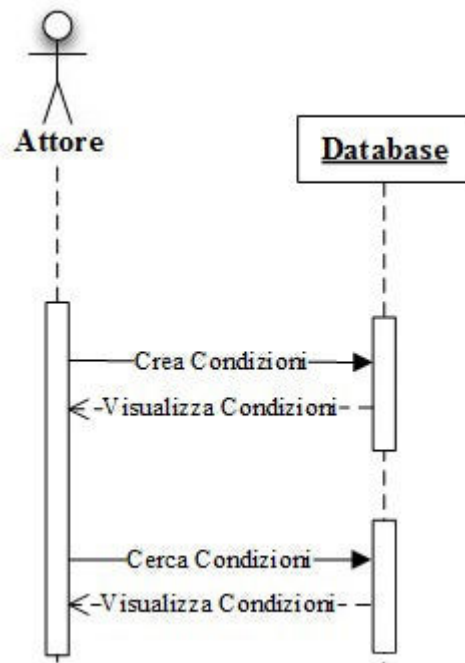


Figura 2.40: Sequence Diagram - Condizioni generali di fornitura

2.3.7 Clienti

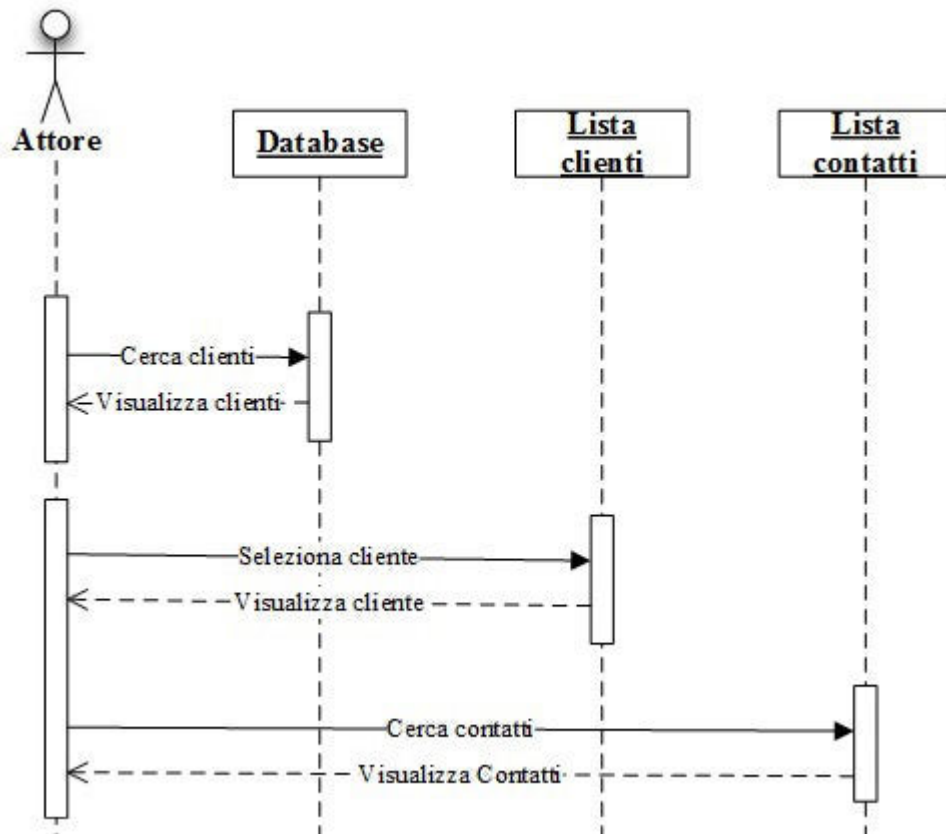


Figura 2.41: Sequence Diagram - Clienti

2.3.8 Materiali

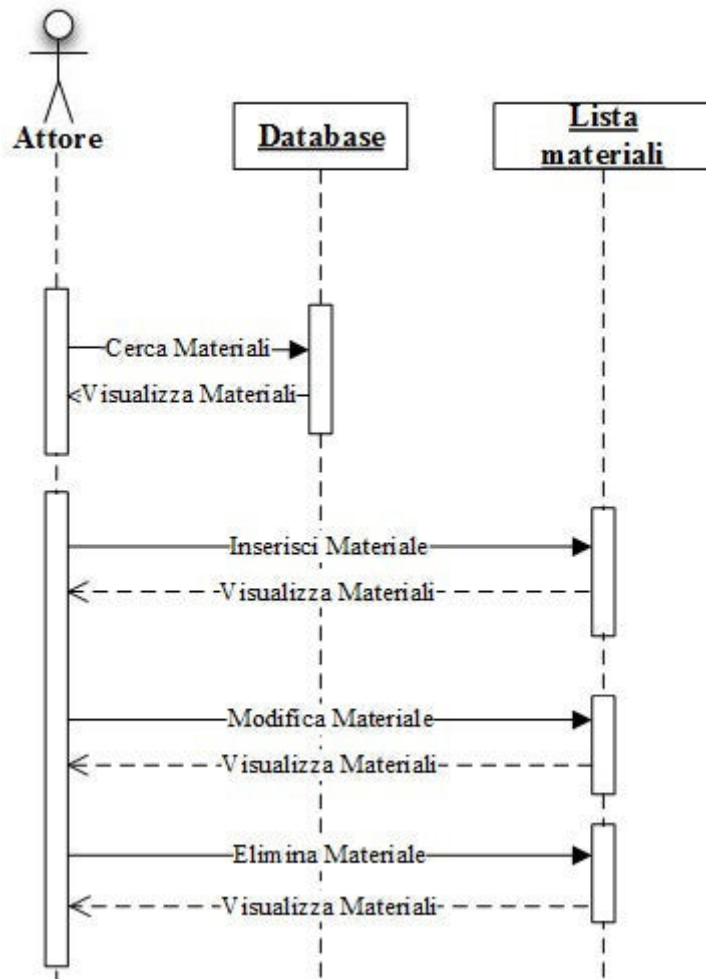


Figura 2.42: Sequence Diagram - Materiali

2.3.9 **Inserisci Chiamata**

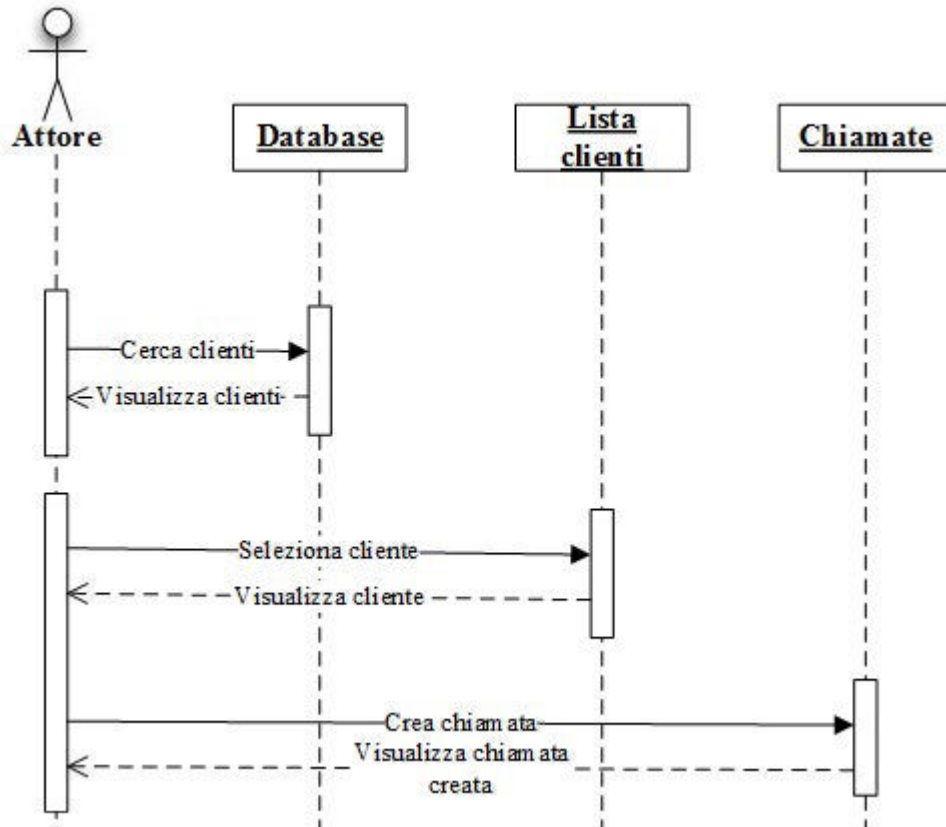


Figura 2.43: Sequence Diagram - Inserisci Chiamata

2.3.10 Gestisci Chiamata

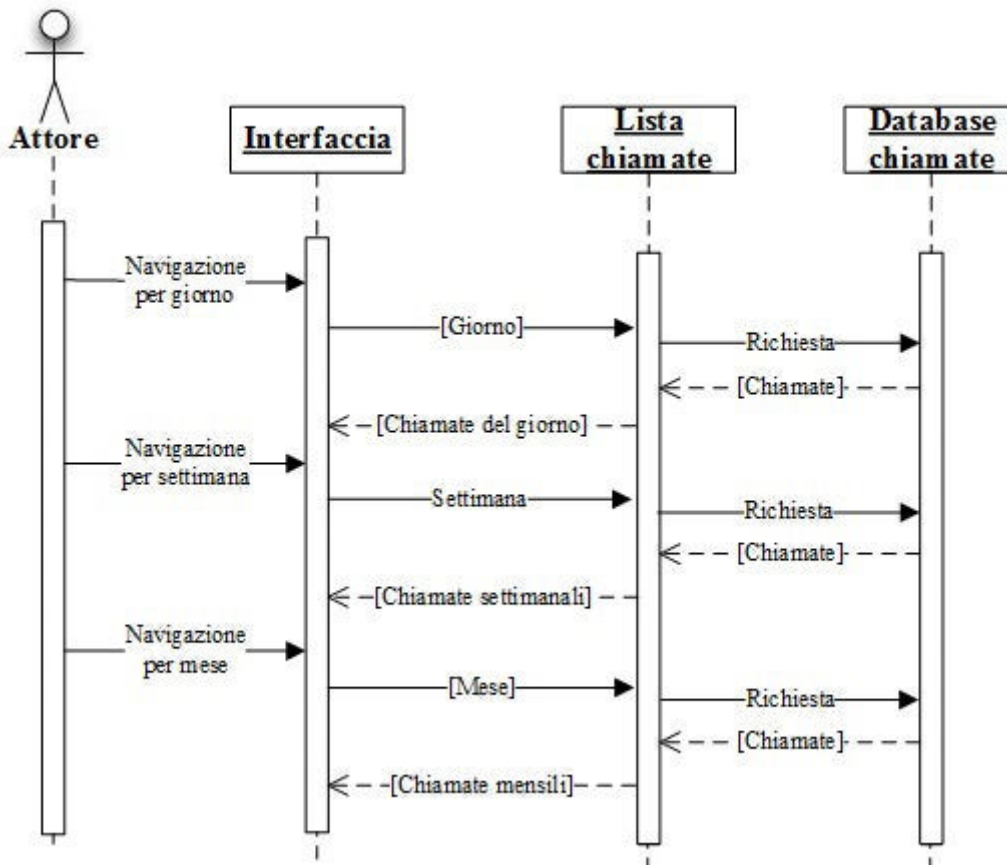


Figura 2.44: Sequence Diagram - Gestisci Chiamata

Vedremo ora in modo più approfondito il sequence diagram di una singola chiamata.

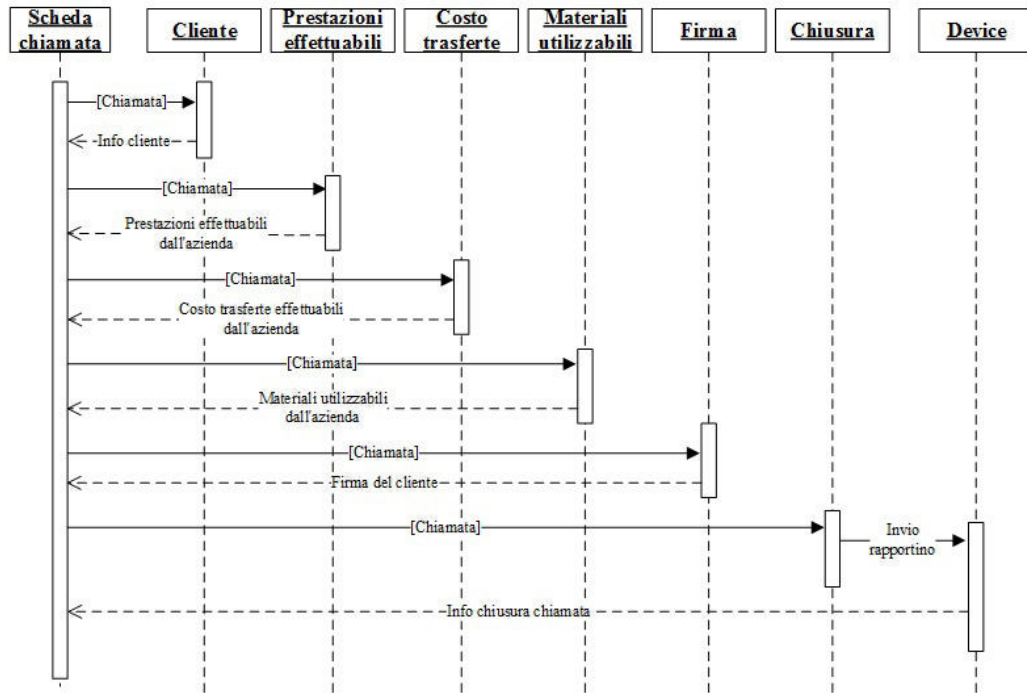


Figura 2.45: Sequence Diagram - Chiamata

2.3.11 Storico

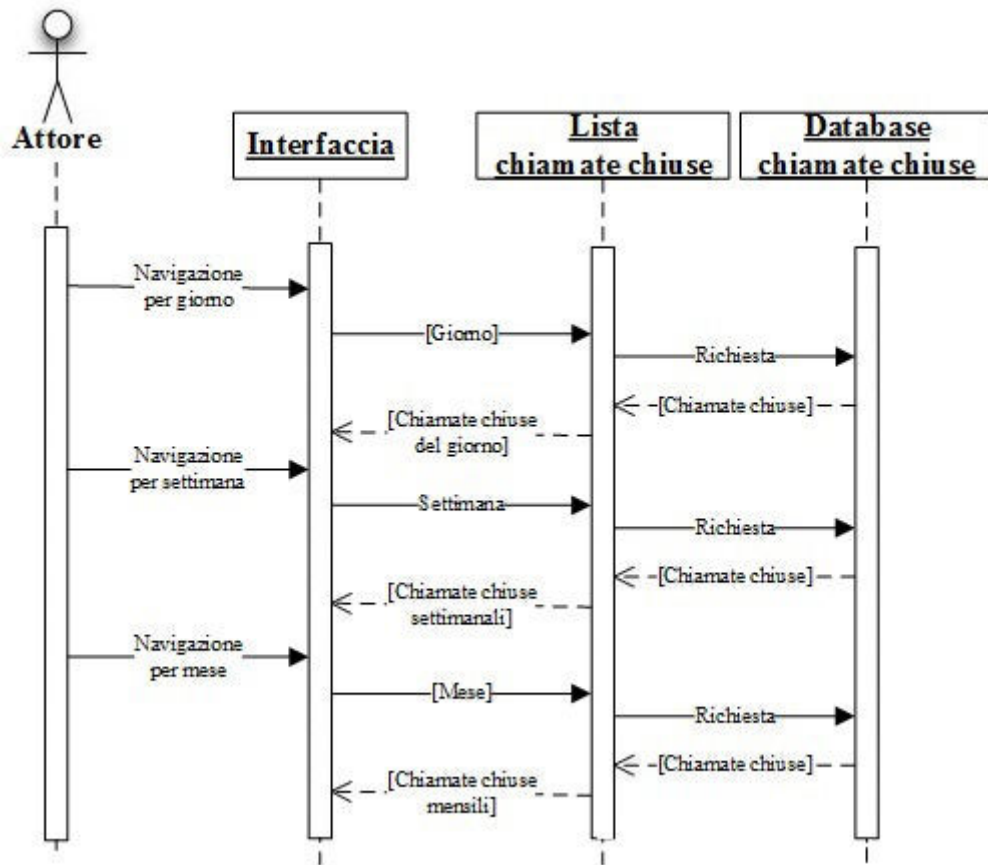


Figura 2.46: Sequence Diagram - Storico

2.3.12 Impostazioni

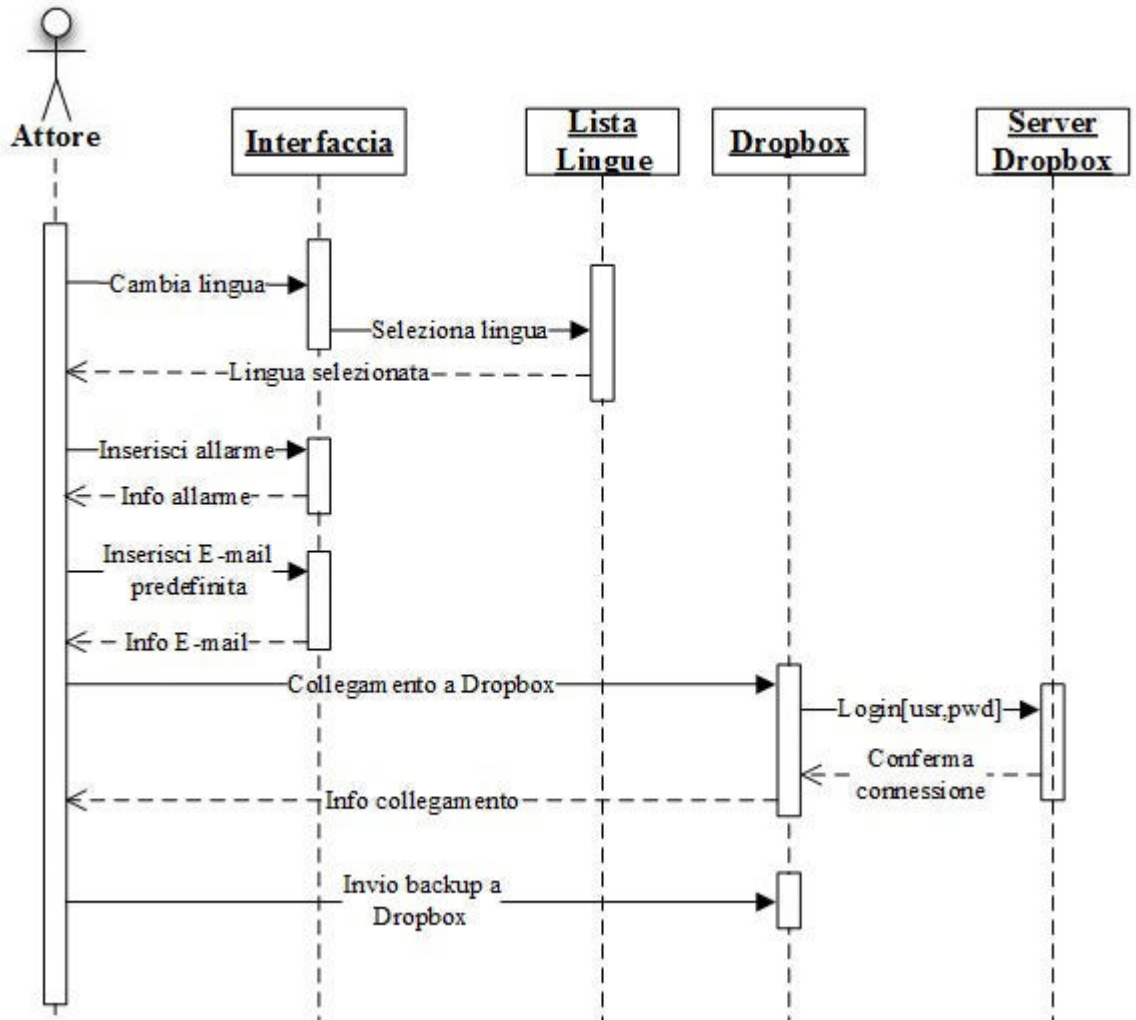


Figura 2.47: Sequence Diagram - Impostazioni

2.4 Modello E-R

Il modello entity-relationship (modello E-R) è un modello per la rappresentazione concettuale dei dati ad un alto livello di astrazione, formalizzato dal prof. Peter Chen nel 1976. Viene spesso utilizzato nella prima fase della progettazione di una *base di dati* in cui è necessario tradurre le informazioni risultanti dall'analisi di un determinato dominio in uno *schema concettuale*. Tramite il modello E-R è possibile evidenziare gli aspetti fondamentali del modello concettuale con i dati che lo caratterizzano e le relative associazioni. Gli elementi del modello E-R sono:

- **Entità:** rappresentano classi di oggetti (fatti, cose, persone, ...) che hanno proprietà comuni ed esistenza autonoma ai fini dell'applicazione di interesse. Un'occorrenza di un'entità è un oggetto o istanza della classe che l'entità rappresenta. Non si parla qui del valore che identifica l'oggetto ma dell'oggetto stesso. Un'interessante conseguenza di questo fatto è che un'occorrenza di entità ha un'esistenza indipendente dalle proprietà ad essa associate. In questo, il modello E-R presenta una marcata differenza rispetto al modello relazionale nel quale non possiamo rappresentare un oggetto senza conoscere alcune sue proprietà. In uno schema, ogni entità ha un nome che la identifica univocamente, e viene rappresentata graficamente tramite un rettangolo con il nome dell'entità al suo interno;
- **Associazione:** rappresentano un legame tra due o più entità. Il numero di entità legate è indicato dal grado dell'associazione: un buono schema E-R è caratterizzato da una prevalenza di associazioni con grado due. È possibile legare un'entità con se stessa (attraverso un'associazione ad anello), nonché legare le stesse entità con più associazioni. Di norma viene rappresentata graficamente da un rombo contenente il nome dell'associazione. Il nome può essere un verbo in modo da fornire una direzione di lettura, oppure può essere un sostantivo in modo da non dare una direzione di lettura. L'orientamento accademico e professionale più recente propende per l'utilizzo del sostantivo proprio per evitare di dare un verso all'associazione;
- **Attributi:** ciò che descrive le proprietà delle entità e delle associazioni. La scelta degli attributi riflette il livello di dettaglio con il quale vogliamo rappresentare le informazioni sulle entità e sulle associazioni. Per ciascuna classe entità o associazione si definisce una chiave. La chiave è un insieme minimale di attributi che identifica univocamente un'istanza di entità o associazione.

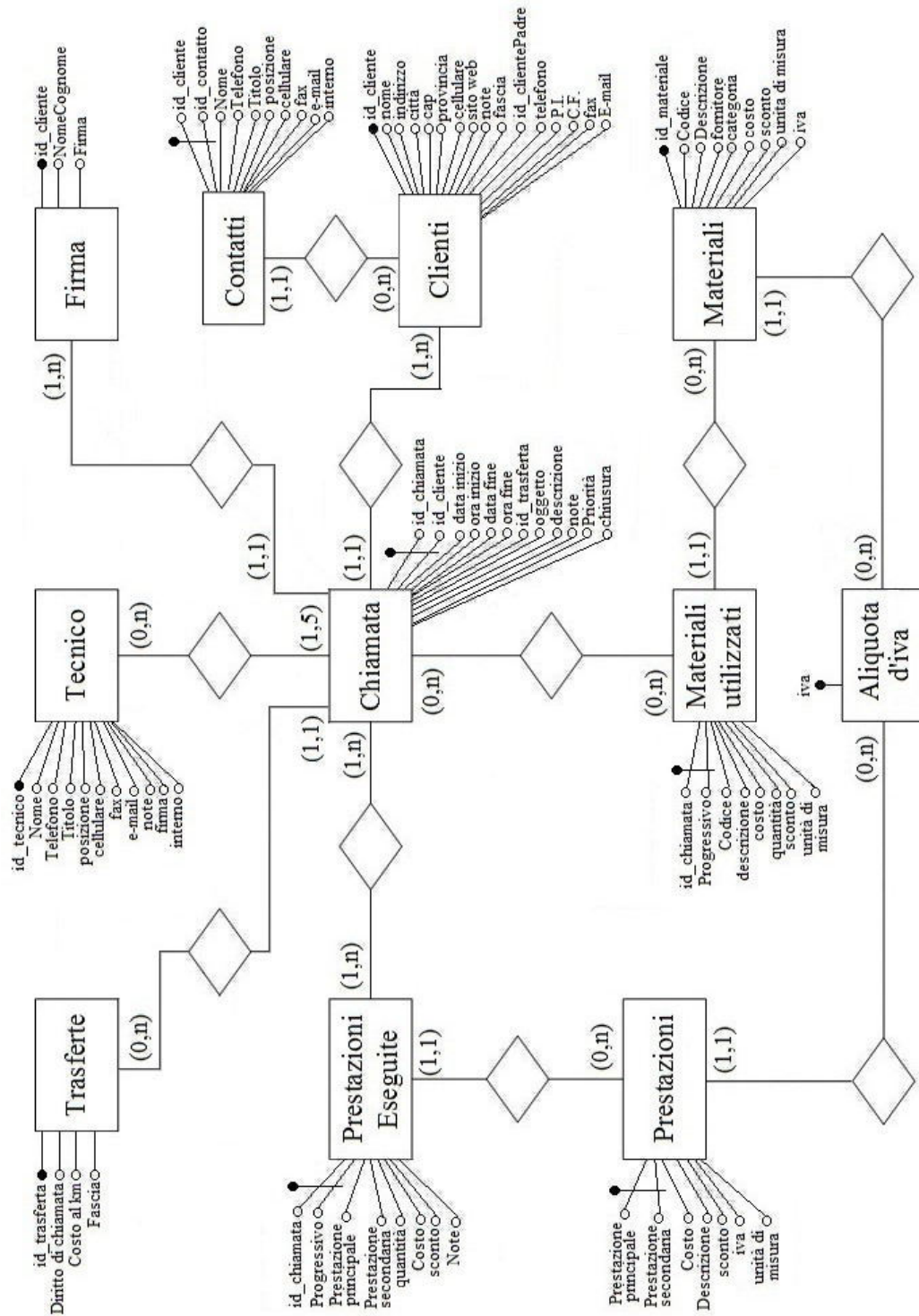


Figura 2.48: Modello E-R

Entità	Descrizione	Attributi	Identificatore
Chiamata	Chiamata di assistenza	id_chiamata, id_cliente, data inizio, ora inizio, data fine, ora fine, oggetto, descrizione, note	id_chiamata, id_cliente
Clienti	Cliente al quale si effettua l'intervento	id_cliente, nome, indirizzo, città, cap, provincia, cellulare, sito web, note, fascia, id_clientePadre, telefono, P.I., C.F., fax, e-mail	id_cliente
Contatti	Contatti dei clienti	id_cliente, id_contatto, nome, telefono, titolo, posizione, cellulare, fax, e-mail, interno	id_cliente, id_contatto
Firma	Firma del cliente	id_cliente, nomeCognome, firma	id_cliente
Tecnico	Tecnico che svolge l'assistenza	id_tecnico, nome, telefono, titolo, posizione, cellulare, fax, e-mail, note, firma, interno	id_tecnico
Trasferte	Costo della trasferta dell'assistenza	id_trasferta, diritto di chiamata, costo al km, fascia	id_trasferta
Prestazioni eseguite	Prestazioni effettuate dal tecnico durante l'assistenza	id_chiamata, progressivo, prestazione principale, prestazione secondaria, quantità, costo, sconto, note	id_chiamata, progressivo
Prestazioni	Tutte le prestazioni che può effettuare il tecnico	prestazione principale, prestazione secondaria, costo, iva, descrizione, unità di misura, sconto	prestazione principale, prestazione secondaria

Materiali utilizzati	Materiali utilizzati dal tecnico durante l'assistenza	id_chiamata, progressivo, codice, descrizione, costo, quantità, sconto, unità di misura	id_chiamata, progressivo
Materiali	Tutti i materiali che può utilizzare il tecnico	id_materiale, codice, descrizione, categoria, fornitore, unità di misura, costo, iva, sconto	id_materiale
Aliquota d'iva	Le aliquote d'iva che applica l'azienda	iva	iva

Tabella 2.1: Dizionario dei dati entità[1]

Lo schema precedentemente illustrato (Figura 2.48), riguarda il modello E-R utilizzato per la creazione del database. È stato scelto l'utilizzo di un unico database per raccogliere insieme tutti i dati dell'azienda e dei clienti, così in caso di aggiornamenti si dovrà modificare un solo database. Inoltre, sono stati creati due database aggiuntivi riguardanti le prestazioni e i materiali utilizzati durante l'assistenza, in modo da mantenere separate le possibili prestazioni che può effettuare l'azienda da quelle effettuate durante l'assistenza, e i materiali che fornisce, da quelli utilizzati durante l'assistenza.

2.5 Descrizione dettagliata del funzionamento

L'app ibrida Assistenza Interventi Tecnici fa uso del framework JQuery Mobile per la grafica e la gestione degli eventi. Essa, infatti, fa leva sulla struttura semantica delle pagine HTML5 e sugli attributi *data-* per definire le varie parti dell'interfaccia. Una volta caricata la pagina, JQuery Mobile utilizzerà questa struttura per arricchirla con altri tag e *agganciare* gli eventi e le interazioni ai componenti dell'applicazione.

L'app è stata ideata in modo da lavorare *stand-alone*, cioè tutti i database e le interazioni avvengono sul dispositivo senza la necessità di utilizzare la connessione internet. Esistono però due funzioni in cui è necessario utilizzarla:

- Chiusura chiamata di assistenza: una volta conclusa una chiamata verrà creato in pdf il rapportino finale dell'assistenza ed inviato per e-mail;
- Dropbox: per collegarsi all'account di dropbox ed effettuare i backup.

L'interfaccia grafica di Assistenza Interventi Tecnici deve essere analoga su tutti i dispositivi nei quali verrà installata. Per realizzare ciò, si utilizza il CSS in quanto capace di adoperare dimensioni percentuali e assolute; in particolare:

- Margini, padding, dimensioni verticali e label hanno dimensioni in pixel;
- Le larghezze dei div hanno dimensioni percentuali.

I tecnici, i contatti dell'azienda e dei clienti, i materiali, le prestazioni, le aliquote d'iva e le fasce delle trasferte sono tutti visualizzati sotto forma di elenchi organizzati secondo l'ordine di inserimento. Invece, i clienti sono ordinati in modo alfabetico sia nella relativa sezione (*Clienti*), sia in *Inserisci chiamata*. In *Gestisci le chiamate*, queste ultime vengono ordinate per data/ora e per clienti. Saranno così raggruppate tutte le chiamate per ogni cliente ad esso assegnate e ordinate per data/ora. Per rendere la visualizzazione più semplice e intuitiva è stato aggiunto un filtro temporale:

1. Visualizzazione giornaliera;
2. Visualizzazione settimanale;
3. Visualizzazione mensile.

Alla pressione su di una chiamata si aprirà una schermata suddivisa in sei pagine per la sua gestione, inserendo così i dati riguardanti l'assistenza svolta. È possibile navigare attraverso le varie sezioni tramite swipe JQuery o grazie ad un particolare tag di JQuery. Si ha avuto necessità di introdurre due metodi per la navigazione poiché la nostra prima opzione swipe JQuery (l'evento creato dal movimento

orizzontale del dito) in alcune versioni di Android non si attivava. Pertanto è stato deciso di permettere all'utente la scelta del metodo a lui più opportuno per navigare tra le sezioni. Lo stesso concetto appena descritto vale anche per lo *storico*, ma in questo caso alla pressione su di una chiamata si aprirà semplicemente una pagina in cui verrà visualizzato il resoconto della chiamata chiusa (del tutto simile al rapportino in pdf creato e spedito via e-mail).

Capitolo 3

Ambiente di sviluppo

Stabiliti i requisiti che Assistenza Interventi Tecnici dovrà soddisfare (e dopo aver fornito un progetto UML), si è passati alla fase di sviluppo del software.

3.1 Framework Phonegap

Lo sviluppo di un'app per il mondo mobile può essere suddiviso in tre categorie a seconda delle tecnologie utilizzate:

- **App native:** sono app scritte e compilate per una specifica piattaforma utilizzando uno dei linguaggi di programmazione supportati dal particolare sistema operativo (per esempio per iOS il linguaggio nativo è Objective-C);
- **Web app:** sono pagine web ottimizzate per dispositivi mobili sfruttando le tecnologie Web, in particolare HTML5, javascript e CSS;
- **App Ibride:** sono app che cercano di sfruttare il meglio delle due categorie precedenti in quanto scritte con tecnologie web (HTML5, javascript e CSS citati prima) ed eseguite localmente all'interno di un'applicazione nativa;

Ognuna di queste tipologie ha aspetti positivi e negativi, per cui la scelta tra esse deve essere fatta valutando attentamente le condizioni e il contesto in cui si dovrà operare. Le *applicazioni native*, per esempio, hanno una maggiore velocità di esecuzione e una maggiore integrazione con l'interfaccia grafica e le interazioni, ma non possono essere direttamente portabili su altre piattaforme. Una *web app*, invece, risulta indipendente dalla piattaforma ma richiede una connessione internet sempre attiva. Non è in grado di accedere al file system o ad altre risorse hardware. L'*app ibrida*[2], infine, unisce all'indipendenza dalle piattaforme delle web app (senza la necessità di una connessione internet) la capacità di accedere alle risorse locali del dispositivo delle app native. Ovviamente questa tipologia

soffre una minore efficienza del rendering grafico e spesso l'aspetto dell'interfaccia grafica potrebbe risultare non omogeneo. Per la creazione di quest'app si è scelto di utilizzare l'approccio ibrido in modo da poterla rendere operante su più piattaforme e, per fare ciò, ci siamo avvalsi del framework Phonegap.

3.1.1 Utilizzo di Phonegap

Phonegap[3] è un framework open-source, sviluppato dalla Nitobi Software, che permette di realizzare applicazioni ibride. L'idea fondamentale di questo progetto può essere riassunta dallo slogan "Write once, port everywhere". Phonegap, infatti, si propone di fornire un supporto che consenta allo sviluppatore, anche sprovvisto di conoscenze specifiche riguardanti le diverse piattaforme, di realizzare un'applicazione cross-platform e di concentrarsi sulla realizzazione della sola parte web.

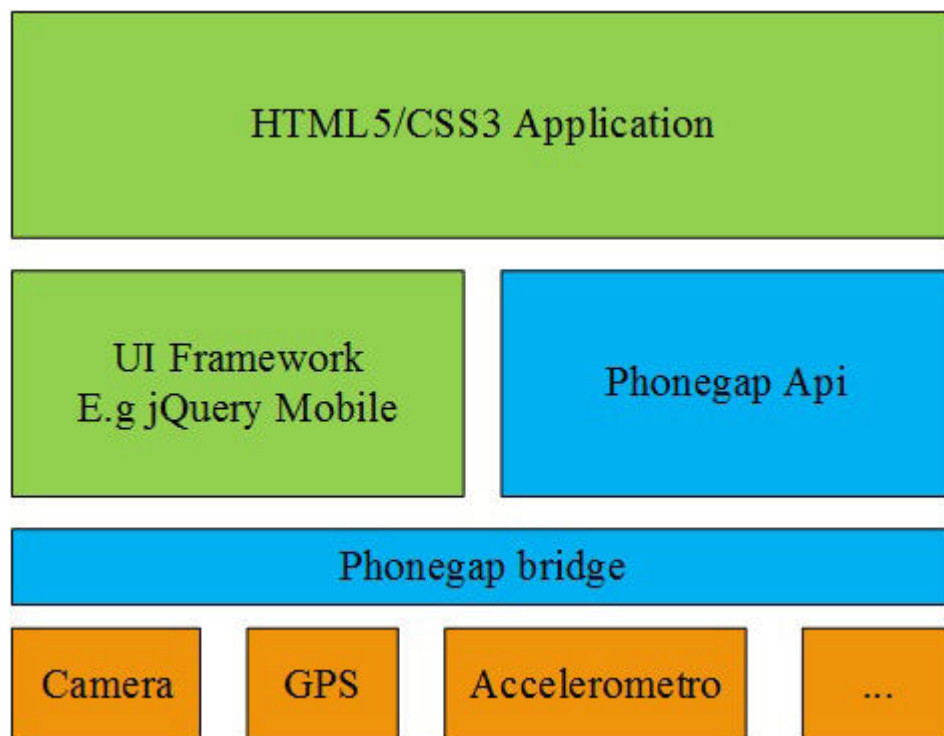


Figura 3.1: Architettura di un'app Phonegap

PhoneGap rappresenta un valido strumento che permette di costruire applicazioni in HTML, Javascript e CSS, capaci di accedere alle funzionalità di base, normalmente non accessibili da una semplice applicazione web, in piattaforme come iPhone, Android, Windows Phone, Blackberry e Tizen. PhoneGap realizza la funzione di bridge, precedentemente anticipata, tra l'applicazione e le funzionalità hardware del dispositivo mobile fornendo una modalità "standard", realizzata tramite l'utilizzo di semplici invocazioni JavaScript, per invocare le API native in maniera indipendente dal tipo di piattaforma sottostante. Il porting verso le varie piattaforme viene conseguentemente realizzato, installando gli ambienti di sviluppo relativi e compilando l'applicazione ibrida realizzata.

Il vantaggio fondamentale della realizzazione di applicazioni ibride con Phonegap consiste nel realizzare un'applicazione che si "adatta" a qualunque piattaforma tra quelle supportate. Phonegap deve, quindi, interfacciarsi sia con l'applicazione che con ciascun sistema operativo della piattaforma sottostante. Tutto ciò viene realizzato usando il linguaggio nativo della piattaforma che permette l'accesso alle risorse hardware e software, mettendo a disposizione queste funzionalità allo sviluppatore attraverso il linguaggio Javascript, così da renderle facilmente utilizzabili dall'applicazione come fossero tradizionali metodi di libreria. In base alla tipologia di piattaforma con la quale dovrà interfacciarsi, l'implementazione di collegamento, fornita dallo stesso framework, sarà di conseguenza sviluppata in Objective-C per iPhone, in Java per Android, in C++/C# per Windows Phone e così via.

Per creare un progetto tramite phonegap dovremo, innanzitutto, scaricare il framework dal sito ufficiale (seguendo le linee guida in esso descritte) e, una volta installato tutto il necessario, potremo realizzare la cartella del progetto in cui saranno creati tutti i file necessari allo sviluppo dell'app ibrida. Il tutto tramite la seguente linea di comando:

```
phonegap create MiaApp
```

Tuttavia, ciò realizzerà solo la base di sviluppo della nostra app; per direzionarci sul sistema voluto (Android o iOS o Windows Phone) dovremo scrivere all'interno della cartella MiaApp la seguente linea:

```
phonegap run Android/iOS/wp8
```

Questo comando permetterà di specificare la nuova cartella disponente le librerie necessarie per poter sviluppare la nostra app ibrida sul sistema operativo scelto. All'interno della sopracitata cartella ne verrà creata una denominata www in cui andremo ad inserire le varie pagine html, i fogli di stile e i file javascript. Un file sul quale dovremo porre particolare attenzione è il *config.xml*, il quale consente di controllare molti aspetti del comportamento di un'applicazione.

Per avere possibilità di interazione con le risorse del dispositivo (e sempre utilizzando linee di comando) sarà fondamentale aggiungere le librerie necessarie al progetto che si sta realizzando (ad esempio se volessimo accedere alla fotocamera dovremmo installare la libreria apposita).

Dal momento che Phonegap mette a disposizione solo le librerie, bisognerà utilizzare i software nativi dei sistemi operativi per poter creare, simulare e pubblicare l'app. Quindi, per iOS sarà necessario utilizzare Xcode (nel quale è già presente l'SDK di iOS), per Android si utilizzerà il più comune Eclipse (ma bisognerà scaricare l'SDK di Android separatamente) e infine per windows phone si adopererà Visual studio (con relativo SDK).

3.2 Software per visual editor

Uno degli svantaggi riscontrati nell'utilizzo di phonegap consiste nel fatto che le pagine HTML5 si devono scrivere a "mano". Vale a dire che ogni tag di ciascuna pagina deve essere inserito direttamente dallo sviluppatore. Il risultato è un processo molto laborioso che potrebbe richiedere anche un notevole dispendio di tempo nel caso in cui fosse necessaria la scrittura di molte pagine. Per ovviare a questo problema si è ricercato un software che integri le funzionalità di phonegap e permetta di creare le pagine in modo automatico tramite il visual editor drag&drop. Le opzioni trovate consistono in tre applicazioni:

- *Application craft*;
- *Appery*;
- *Intel XDK*.

La scelta finale è ricaduta su Intel XDK. Di seguito spiegheremo perché gli altri sono stati scartati:

- *Application craft*: è un software online, quindi non è necessario scaricare alcun programma sul computer. Questo permette di creare l'app da qualsiasi pc dotato di browser. Inoltre, una volta ultimata la sua realizzazione, è possibile scaricarla con le librerie phonegap. Tuttavia, il software richiede un abbonamento mensile e, cosa ancora più importante, non viene aggiornato da oltre un anno (è risultato infatti un progetto abbandonato). A causa di queste ragioni, è stato scartato.
- *Appery*: è un software online compatibile con phonegap. Si è deciso di scartarlo perché la libreria di oggetti utilizzabili è molto scarna e perché anch'esso, come *Application craft*, richiede un abbonamento mensile.

3.2.1 Intel XDK

Questo è un software completamente gratuito creato dalla Intel[4] per dare un supporto a chiunque voglia approcciarsi al mondo delle app. Infatti, si tratta di un programma completo che, oltre ad offrire allo sviluppatore una serie di strumenti quali visual editor drag&drop ed emulatori per vari tipi di smartphone, integra le librerie Phonegap e jQuery Mobile e permette la creazione del pacchetto di installazione dell'applicazione, pronta ad essere caricata sugli store.

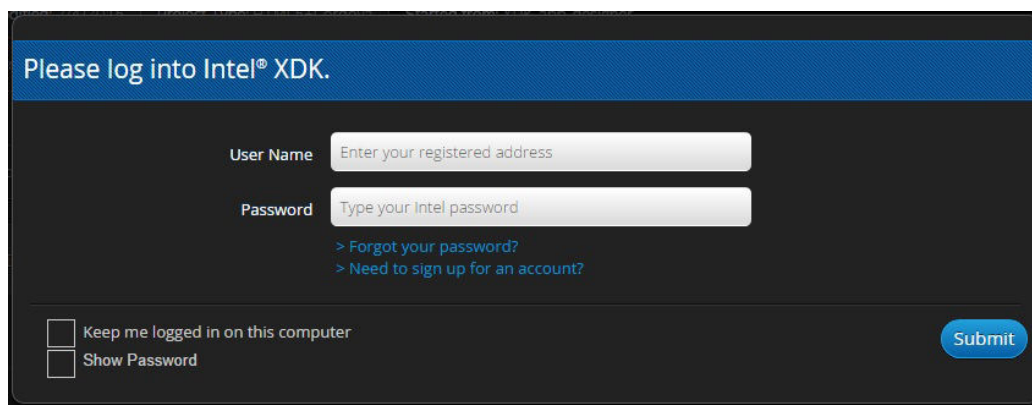


Figura 3.2: Screenshot - Login Intel XDK

Una volta scaricato dal sito dell'Intel (per il sistema operativo desiderato) ed installato, si potrà avviare il programma che, come messaggio preliminare, (vedi figura 3.2) richiede l'username e la password con cui ci siamo registrati sul sito, offrendo anche la possibilità di creare un account (nel caso in cui non ci fossimo ancora registrati). Per creare e testare sugli emulatori la nostra app non sarà necessario essere autenticati; tuttavia, se decidessimo di usufruire di tutte le funzionalità del programma (test, debug, build) occorrerà registrarsi.

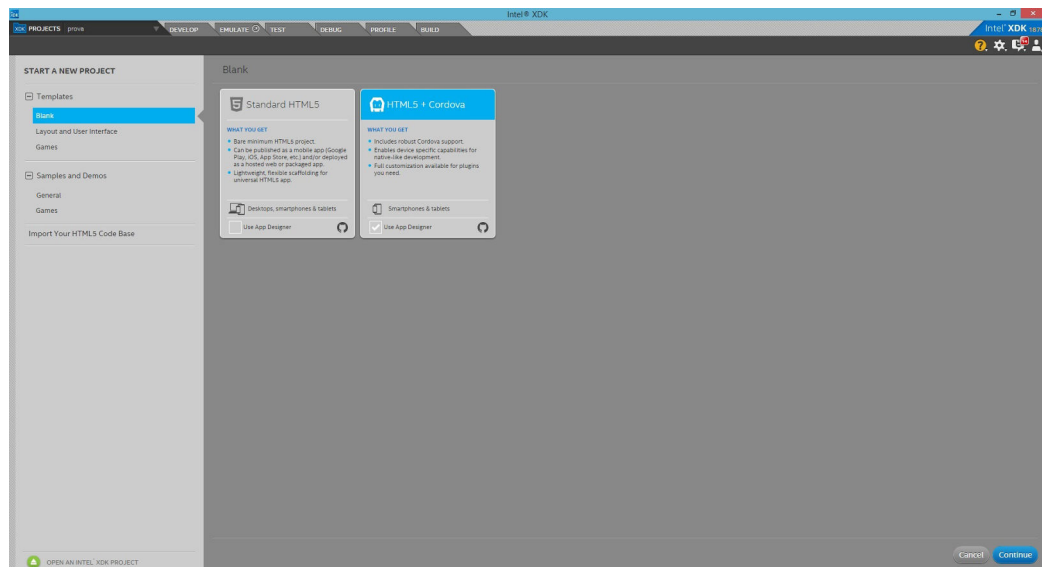


Figura 3.3: Screenshot - Intel XDK, Prima pagina

In Intel XDK il lavoro di sviluppo si suddivide in tre macro blocchi: *Templates*, *Samples and Demos* e *Import your HTML5 Code Base*. Il primo blocco annovera tre sezioni: *Blank*, nella quale sarà possibile iniziare un nuovo lavoro; *Layout and user interface*, in cui si potrà partire da un layout predefinito (per esempio con la pagina di login o con una lista di elementi); *Games*, dove potrà essere scelto un tool grafico (Cocos2d, Phaser, Pixi, EaselJS e Basic Canvas Game). In *Samples and Demos*, come suggerisce il nome, sono presenti esempi di applicazioni già precostruite: sia semplici app (come una calcolatrice o un servizio di prenotazione voli) sia giochi.

Nell'ultimo blocco invece potremo importare un codice HTML5.

Per svolgere il lavoro al meglio, Intel XDK mette a disposizione i seguenti strumenti:

- **Editor Brackets:** questa funzione offre specifiche caratteristiche per la scrittura del codice HTML5, javascript e CSS. È un vero e proprio editor di testo, attraverso il quale si possono implementare manualmente le funzioni dell'applicazione o modificarne delle parti;
- **AppDesigner:** è un editor grafico che permette di scegliere quale framework per la realizzazione delle interfacce si vuole utilizzare.

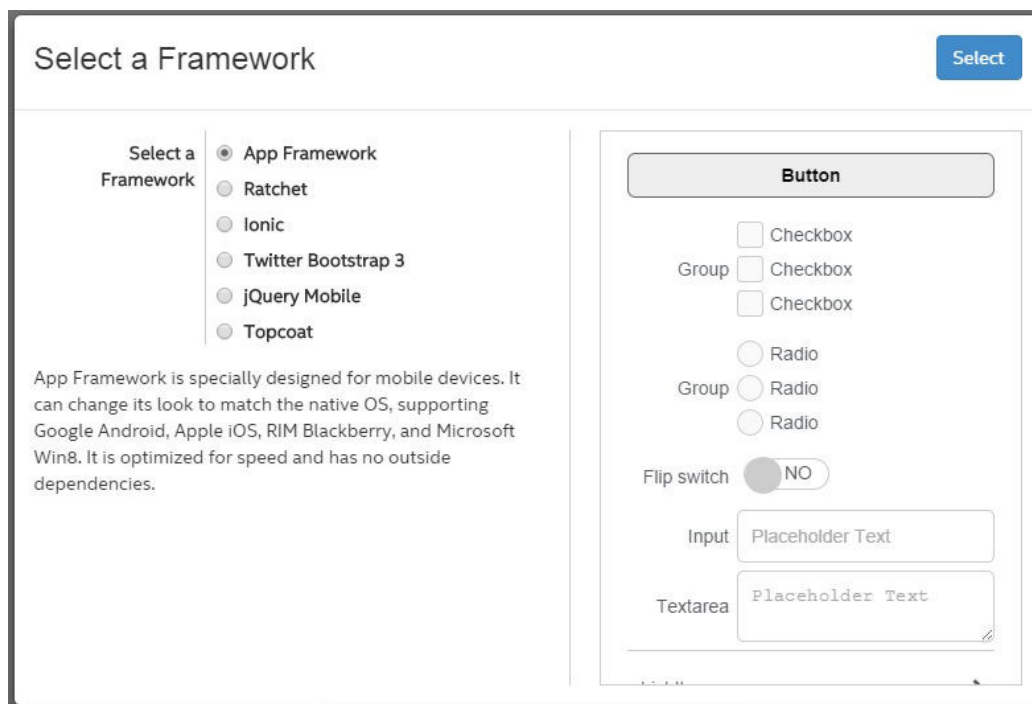


Figura 3.4: Screenshot - Scelta framework

Al momento di andare a definire una nuova pagina html su cui lavorare si apre una finestra dal titolo “Select a framework” (figura 3.4) che offre la possibilità di scegliere il framework preferito. Le scelte possibili rappresentano in buona parte piattaforme familiari agli sviluppatori del web, quali: App Framework, Ratchet, Ionic, Twitter Bootstrap 3, jQuery Mobile e Topcoat;

- **App Starter:** è un ambiente, anch’esso visuale (come AppDesigner), che nasce per sfruttare al massimo la libreria App Framework. In particolare, offre un notevole contributo nella realizzazione di layout multi-view.

Per lo sviluppo dell’app Assistenza Interventi Tecnici siamo entrati nella sezione *Templates/Blanks/HTML5+Cordova* scegliendo di utilizzare AppDesigner e selezionando il Framework jQuery Mobile.



Figura 3.5: Screenshot - Schede Intel XDK

Nella parte alta della schermata, figura 3.5, si trovano le seguenti schede: *Develop*, *Emulate*, *Test*, *Debug*, *Profile* e *Build* ed ognuna di esse rappresenta una fase del ciclo di produzione di un'app Web. *Develop* è la scheda dove i tre strumenti di sviluppo indicati precedentemente (ovvero l'editor Brackets, AppDesigner e AppStarter) possono essere utilizzati. Quindi grazie a questa scheda si potrà passare dallo sviluppo grafico a quello basato sul codice, direttamente con un click.

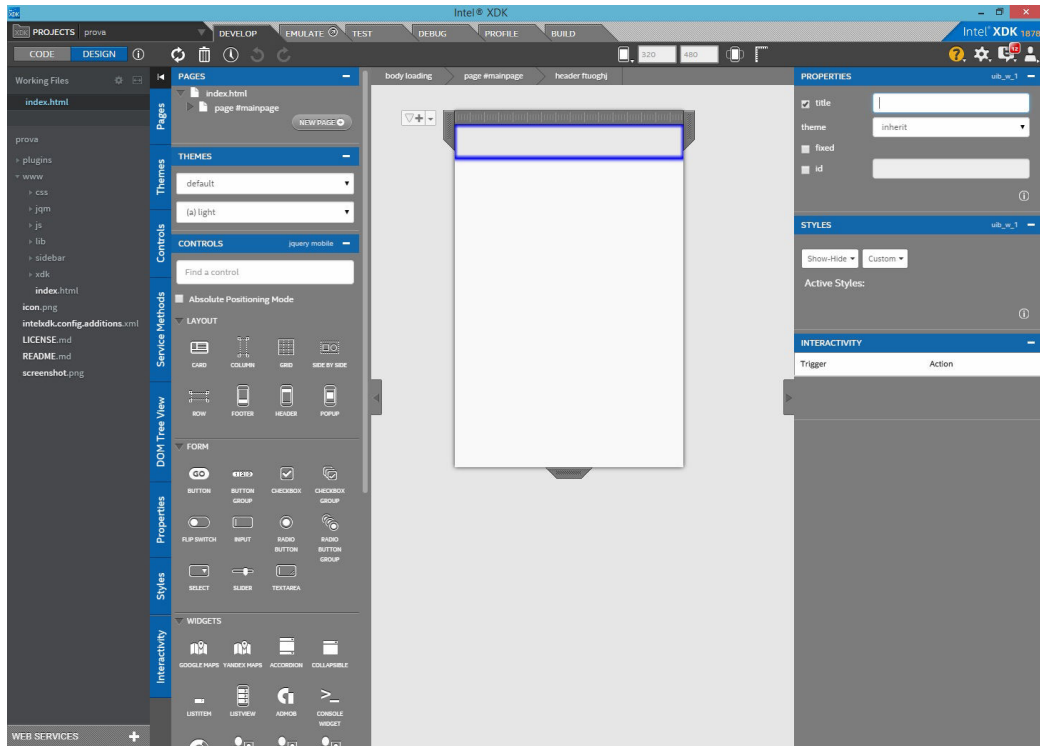


Figura 3.6: Screenshot - Ambiente di lavoro Intel XDK

La scheda *Develop* (Figura 3.6) mostra sulla colonna di sinistra l'architettura del progetto con la relativa suddivisione in cartelle e sottocartelle; subito dopo si trova la colonna all'interno della quale potranno essere inserite nuove pagine o anche trascinati gli elementi necessari all'app. Nella parte centrale della finestra è possibile visualizzare il "corpo" dell'applicazione e la parte dove potranno essere "incollati" gli oggetti anteriormente presi dalla sezione sinistra della scheda. Sull'ultima colonna di destra ci sono altre funzioni che permetteranno di modificare

le proprietà, lo stile e l'interazione degli elementi selezionati.

Emulate offre la possibilità di vedere in tempo reale la propria app al lavoro. Questa funzione è un emulatore che mostra il comportamento dell'app su vari dispositivi e sistemi operativi.

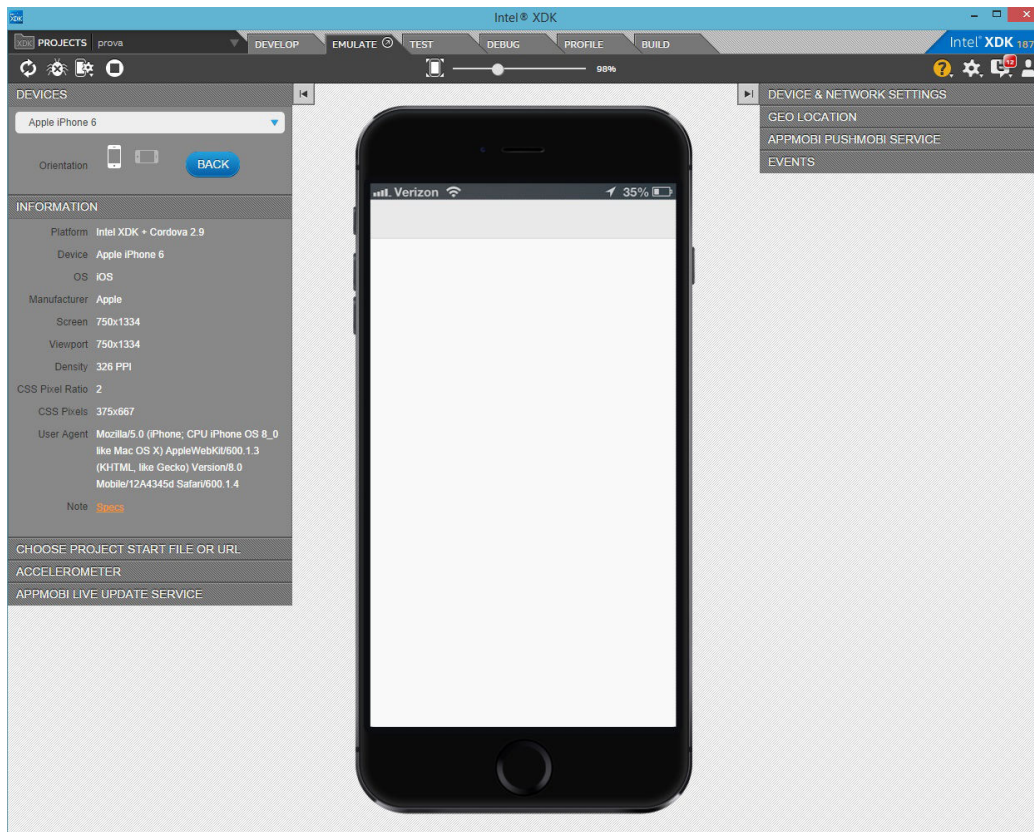


Figura 3.7: Screenshot - Scheda emulatore

Come si può notare dalla figura 3.7, oltre ad emulare l'app sul dispositivo selezionato, la scheda *Emulate* permette funzioni come il cambio di orientazione, la simulazione dell'accelerometro, la Geo-localizzazione ecc.. Un ulteriore passo in avanti rispetto ad *Emulate* è compiuto dalla scheda *Test*, la quale permette, previa installazione sul proprio dispositivo reale dell'applicazione "Intel App Preview" (disponibile su tutti gli store dei sistemi operativi mobile), di provare l'app che si sta sviluppando su un dispositivo reale.

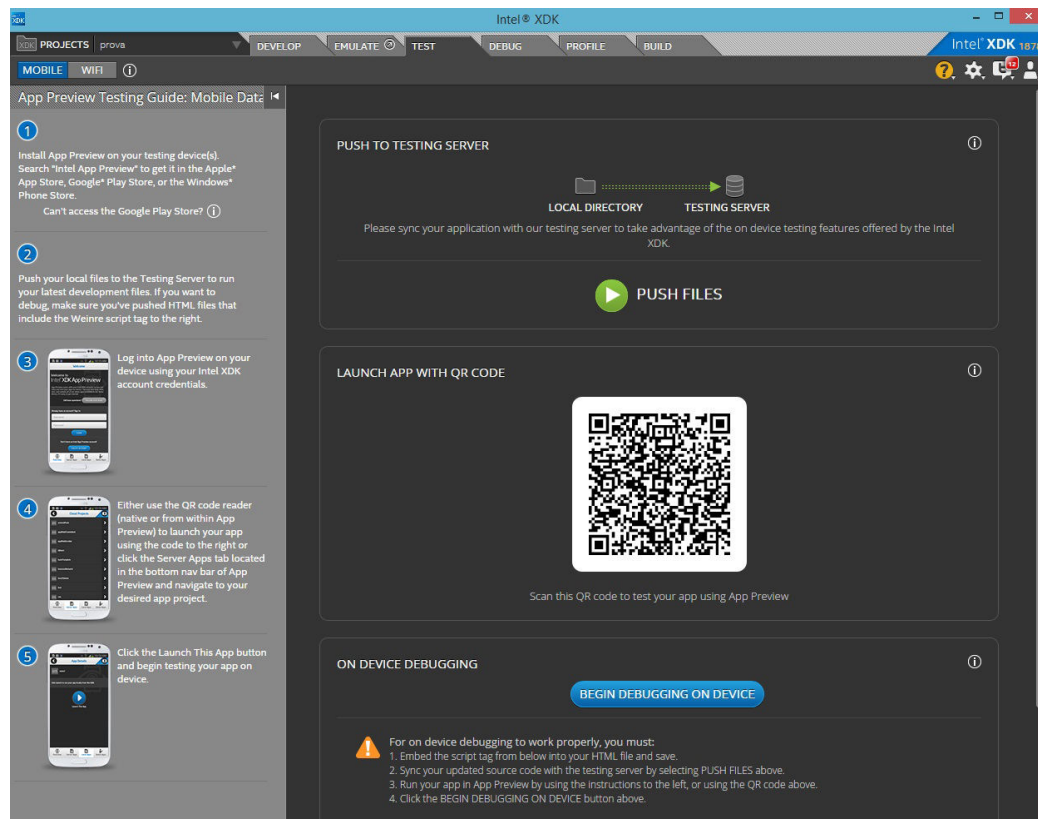


Figura 3.8: Screenshot - Scheda Test

L'applicazione, presente su Intel XDK, potrà essere inviata ad Intel App Preview tramite il comando *push* (che la caricherà su un server) e, successivamente, potrà essere recuperata tramite l'App, per poi venire testata sullo smartphone, oppure tramite il QR code utilizzando il lettore QR presente nell'app.

Debug è una scheda con funzionalità del tutto simile a quella precedentemente descritta; all'interno di essa sarà possibile avviare il debugging dell'applicazione e vedere il suo comportamento sul dispositivo reale, utilizzando sempre Intel App Preview. Pur essendo del tutto simile alla scheda precedente, differisce nel fatto che, invece di servirsi di internet o server, è necessario collegare fisicamente lo smartphone al pc.

Profile è una scheda dedicata ai dispositivi Android in cui è possibile una ulteriore configurazione.

Infine vi è la scheda *Build*: essa permette la compilazione dell'intero progetto per generare un pacchetto di installazione dell'applicazione realizzata.

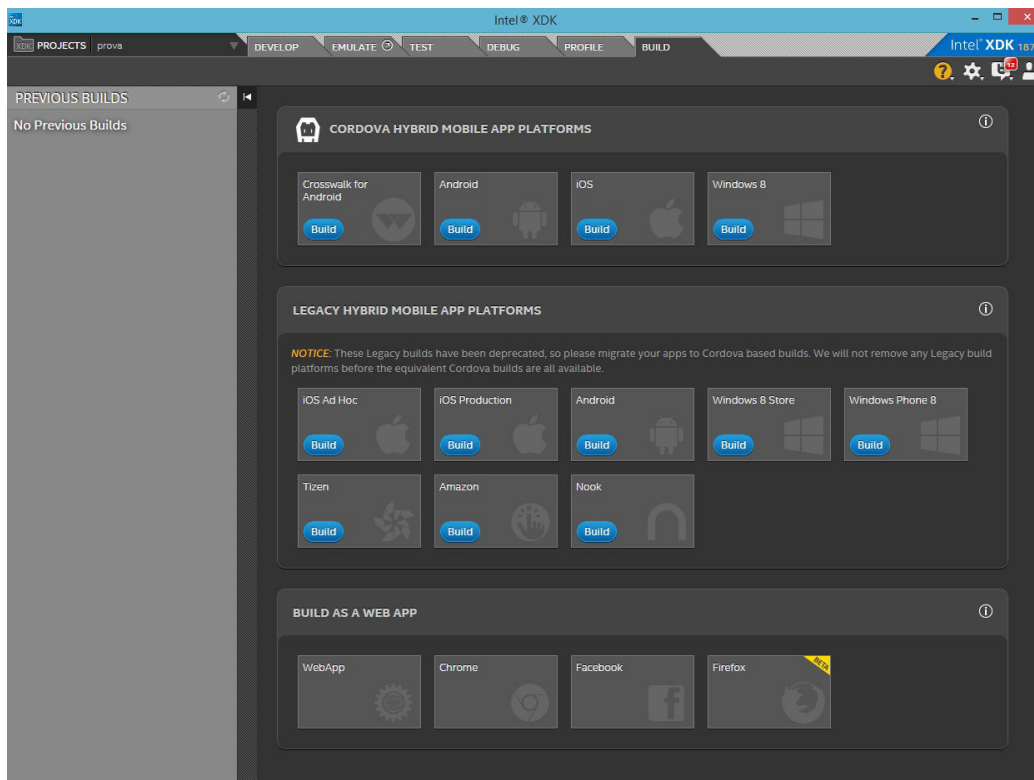


Figura 3.9: Screenshot - Scheda Build

All'interno della scheda, figura 3.9, è possibile scegliere tra i vari Os mobile oppure la piattaforma ibrida. Tuttavia, abbiamo scelto di non avvalerci della funzionalità di quest'ultima scheda per non essere troppo legati al software; quindi, per gli ultimi test e la creazione del pacchetto di installazione, sono stati utilizzati Xcode ed Eclipse, rispettivamente per iOS e Android.

3.3 Xcode

Ambiente di sviluppo integrato (IDE, Integrated Development Environment) realizzato da Apple Inc. per agevolare la creazione di software per Mac OS X e iOS. È fornito gratuitamente in bundle con il sistema operativo a partire da Mac OS X 10.3. Xcode[10] lavora in congiunzione con Interface builder, un tool grafico atto a realizzare interfacce grafiche, del quale, tuttavia, non faremo uso poiché non utilizzeremo l'approccio nativo (pertanto non impiegheremo neanche il linguaggio Objective-C).

Xcode è un utile strumento per riprodurre su emulatore o dispositivo reale il nostro software, però per riuscire ad effettuare il deploy su dispositivo reale è necessario registrarsi al *developer program* e pagare un costo annuale di 99€. Inoltre, Xcode risulta essenziale per poter caricare su *Itunes Connect* l'app, in quanto questo sito web è destinato agli sviluppatori per inserire le app che si vogliono pubblicare e gestirle dopo la loro pubblicazione. Una volta inviata la richiesta di pubblicazione sullo store sarà necessario un periodo di attesa variabile tra i sette e i quattordici giorni in cui l'Apple provvederà ad analizzare l'app per appurare che non vi siano violazioni dei propri regolamenti. Al termine di questa verifica, in caso di approvazione, l'app sarà caricata sull'Apple Store.

3.4 Eclipse

Ambiente di sviluppo integrato multi-linguaggio e multiplatforma in cui è indispensabile inserire manualmente, all'interno del progetto, le librerie e l'SDK di Android. Come nel caso di Xcode, perfino Eclipse[9] fornisce un emulatore per la simulazione dell'app (anche se non sempre ciò che viene mostrato rispetta il reale comportamento dell'app sul dispositivo mobile) ed inoltre è possibile effettuare il deploy su dispositivo reale senza registrarsi o pagare un costo annuale.

Un'altra differenza sostanziale con Xcode risiede nel fatto che, terminata l'app, non viene caricata automaticamente sul nostro account di google play, ma è necessario creare con Eclipse un file di estensione .apk e salvarlo sulla propria macchina locale, per poi in un secondo momento caricarlo sul proprio account e pubblicarlo sullo store.

Rispetto alla pubblicazione su Apple Store, su Google Play non vi è una verifica preliminare approfondita, quindi il tempo tra la richiesta di pubblicazione e l'effettiva messa in commercio, non supera mai le ventiquattro ore.

3.5 Architettura di Assistenza Interventi Tecnici

Essendo un'app stand-alone l'architettura risiede sul dispositivo ed è organizzata in due macro blocchi, uno scritto in codice nativo e uno sviluppato secondo le tecnologie web.

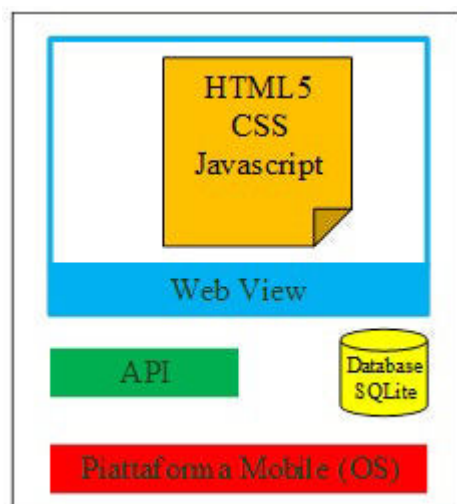


Figura 3.10: Architettura app

Scendendo in dettaglio, è possibile individuare tre componenti indispensabili all'esecuzione dell'applicazione: il sistema operativo del dispositivo, le API necessarie all'applicazione per accedere alle funzionalità tipiche di uno smartphone e i database locali su cui memorizzare i dati. Poiché le API rappresentano di fatto il ponte per la comunicazione tra il sistema operativo e l'applicazione, è possibile raggruppare queste componenti in una struttura composta da tre livelli, come mostrato in figura 3.10.

3.6 Database SQLite

L'uso di database è stato utile per effettuare il cosiddetto local storage, cioè salvare i dati in modo locale per aumentare l'usabilità e le prestazioni del dispositivo. Il database utilizzato è SQLite[5], una libreria software scritta in linguaggio C che implementa un DBMS SQL di tipo ACID.

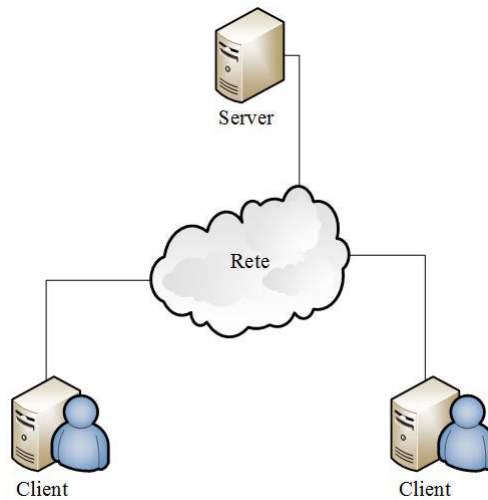


Figura 3.11: Struttura database Client/Server MySQL



Figura 3.12: Struttura database Client/Server SQLite

A differenza di molti altri database SQL, come MySQL, SQLite non ha un processo separato server, cioè non si basa su un'architettura Client/Server. Infatti, in un'architettura Client/Server l'applicazione di database e il database sono separati in due parti (Figura 3.11), cioè il front-end o parte client ed il back-end o parte server. Il client esegue l'applicazione di database che accede alle informazioni del database e interagisce con l'utente attraverso la tastiera e lo schermo mentre il server esegue il software, gestisce le funzioni necessarie e l'accesso condiviso ai dati. Invece SQLite legge e scrive direttamente sul file presente sul disco ordinario (Figura 3.12) e questo permette un accesso molto veloce ai dati, in alcuni casi superiore anche a MySQL.

SQLite è un database completo con tabelle, query, trigger e viste, inoltre, come già detto i database creati sono contenuti in un unico file binario il cui formato interno è multi-piattaforma, cioè indipendente dalla piattaforma e dal relativo ordine dei byte (si può liberamente copiare un database tra i 32-bit e 64-bit o tra architetture

big-endian e little-endian).

Phonegap dispone di una libreria apposita per poter accedere ad SQLite sul dispositivo, ed utilizza i seguenti metodi per la creazione e comunicazione con esso:

- *openDatabase*: apre o crea il database con il nome e la dimensione specificata;
- *transaction*: effettua una transazione sul database;
- *executeSQL*: esegue un'istruzione SQL.

Verrà così usato il primo comando per creare il database nel momento della prima installazione, e successivamente tutte le volte che avremo la necessità di operare sul database. Non sarà possibile usare i comandi *transaction* e *executeSQL* finché prima non si è aperto il database.

3.6.1 Tabelle Assistenza Interventi Tecnici

Il database della nostra app, come già presentato nel modello E-R (paragrafo 2.4), si compone di undici tabelle principali, oltre a queste è presente anche la tabella dove viene memorizzata la condizione generale di fornitura, richiamata soltanto all'atto della creazione del rapportino in pdf.

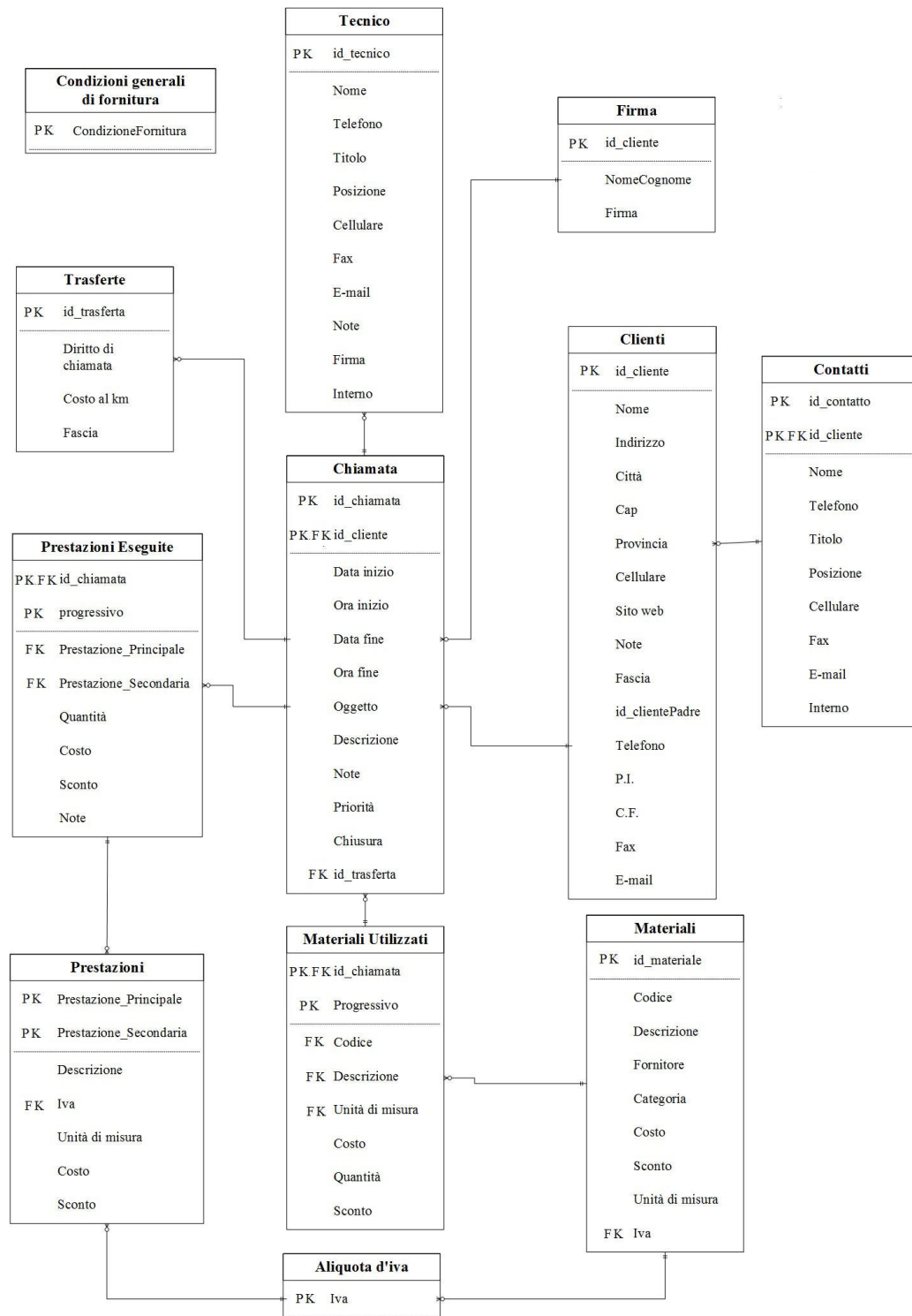


Figura 3.13: Tabelle Assistenza Interventi Tecnici

Andremo ora a descrivere in dettaglio le tabelle:

- **Clienti**: questa tabella contiene i salvataggi dei dati dei clienti. Come primary key viene usato l'*id_cliente*, un numero intero che si auto-incrementa ad ogni insert. Il valore dell'id parte da 2 poiché la prima posizione della tabella (ovvero con l'*id_cliente=1*) viene utilizzata per salvare i dati dell'azienda. Questo ci esenta dal dover creare una tabella del tutto uguale a quella dei clienti ma avente un solo record (i dati dell'azienda);
- **Contatti**: questa tabella contiene tutti i contatti relativi all'azienda e ai clienti. Infatti, le primary key sono sia l'*id_contatto* (intero che si auto-incrementa) sia l'*id_cliente*. Ciò permette di associare sempre ad ogni contatto il giusto cliente;
- **Trasferte**: questa tabella, avente come primary key l'*id_trasferta* (intero che si auto-incrementa), consente il salvataggio dei costi che l'azienda applica alle trasferte. Tuttavia, avremo però delle posizioni bloccate: quando l'id è uguale a uno saremo in presenza del *diritto di chiamata*; quando l'id è uguale a due corrisponderà al *costo per km*. Nonostante vi sia già la colonna preposta a questi due dati, si è scelto anche di bloccarli su id=1 e id=2, in modo da poterne disporre più velocemente nel momento delle query;
- **Tecnico**: questa tabella contiene i dati dei tecnici dell'azienda e la primary key sarà l'*id_tecnico*, un intero che si auto-incrementa;
- **Materiali**: questa tabella, con primary key *id_materiali* intero auto-incrementante, contiene i dati dei materiali salvati. È strettamente legata alla tabella dell'aliquota d'iva, poiché non verrà salvato nessun dato relativo ai materiali se prima non è presente almeno un'aliquota da associarvi;
- **Prestazioni**: questa tabella contiene le prestazioni che l'azienda può effettuare. Come primary key si utilizzano *prestazione principale* e *prestazione secondaria*, ovvero il nome completo della prestazione. La tabella è strettamente legata all'aliquota d'iva, poiché ad ogni prestazione deve essere sempre assegnato un valore d'iva;
- **Aliquota d'iva**: questa tabella contiene tutte le aliquote che l'azienda applica ai materiali o alle prestazioni. La primary key sarà chiaramente direttamente il valore dell'iva;
- **Chiamata**: questa è la tabella principale dell'app, in cui vengono salvate i dati di tutte le chiamate, sia aperte che chiuse (cioè presenti nello storico). Le primary key sono l'*id_chiamata* (un intero che si auto-incrementa) e l'*id_cliente*. Ogni chiamata, pertanto, sarà associata ad un solo cliente;

- **Prestazioni eseguite**: questa tabella contiene le prestazioni eseguite in ogni chiamata, con la relativa quantità e il costo. Come primary key troveremo l'*id_chiamata* e il *progressivo*, che identifica l'ordine di salvataggio delle prestazioni relative alla chiamata;
- **Materiali utilizzati**: questa tabella contiene tutti i materiali utilizzati, la quantità e il costo. Le primary key sono, come nel caso precedente, l'*id_chiamata* e il *progressivo*, che identifica l'ordine di salvataggio dei materiali relativi alla chiamata;
- **Firma**: questa tabella contiene il nome, le note e la firma del contatto cliente relativo alla chiamata. La primary key sarà l'*id_cliente*, poiché viene ipotizzata una firma per cliente; in caso di contatto diverso verrà sovrascritta la vecchia firma;
- **Condizioni generali di fornitura**: questa tabella contiene le condizioni di fornitura dell'azienda. Avendo una sola colonna, la primary key è *condizioneFornitura*.

Capitolo 4

Caratteristiche avanzate dell'App

In questo capitolo verranno presentate le caratteristiche avanzate presenti sull'app Assistenza Interventi Tecnici in modo da aumentarne le funzionalità e la facilità d'uso. In particolare verranno descritti:

1. Fotocamera/Galleria;
2. La firma;
3. La gestione multilingua;
4. Il rapportino di lavoro;
5. L'invio di e-mail;
6. Collegamento a Dropbox.

4.1 Fotocamera/Galleria

Il comando da console per aggiungere il plugin di Phonegap per la gestione della fotocamera e galleria è: *phonegap plugin add org.apache.cordova.camera*. Una volta installata la libreria avremo a disposizione la funzione *camera.getPicture(cameraSuccess, cameraError, cameraOptions)*[6], la quale permette l'avvio della fotocamera (Figura 4.1) o della galleria (Figura 4.2).

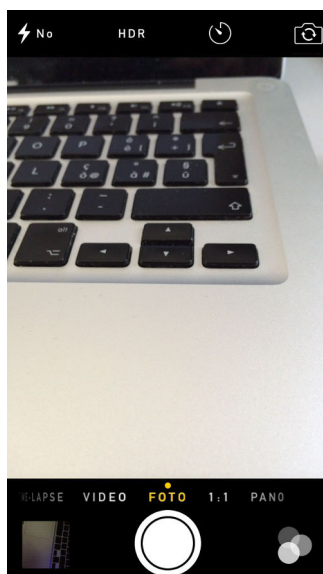


Figura 4.1: Fotocamera nativa



Figura 4.2: Galleria

Una volta acquisita l'immagine (fotocamera o galleria), il valore di ritorno viene inviato alla funzione *cameraSuccess* come string con codifica base64, o come file URI (che rappresenta il percorso del file immagine sullo storage locale). *CameraError* è una funzione chiamata in caso di errore o mal funzionamento da parte della fotocamera/galleria. L'ultima funzione, *cameraOptions*, viene utilizzata per poter personalizzare le impostazioni della fotocamera/galleria, che sono: *quality*, *destinationType*, *sourceType*, *allowEdit*, *encodingType*, *targetWidth*, *targetHeight*. *Quality* è un numero intero, un intervallo da 0-100, per esprimere la qualità dell'immagine. Se si vuole evitare di inviare file di grosse dimensioni, considerando anche che le fotocamere di oggi stanno diventando sempre più ad alta risoluzione, è bene mantenere questo valore sotto la soglia massima. Il valore predefinito è 50. *DestinationType* è il valore di ritorno che viene poi inviato alla funzione *cameraSuccess*, e sarà *camera.DestinationType.DATA_URL* per il base64, e *camera.DestinationType.FILE_URI*, valore predefinito. Per poter avviare la fotocamera è necessario che il valore *sourceType* sia uguale a *camera.PictureSourceType.CAMERA*, oppure non sia presente poiché come impostazione predefinita vi è appunto l'avvio della fotocamera, mentre per avviare la galleria sarà obbligatorio inserire *camera.PictureSourceType.PHOTOLIBRARY*. Una volta che l'utente scatta la foto, o seleziona un'immagine, l'applicazione della fotocamera, o galleria, si chiude e si ritorna così alla pagina dell'app. Per la fotocamera si possono anche impostare: *allowEdit*, valore booleano che permette una semplice modifica dell'immagine dopo l'acquisizione; *encodingType*, è un numero e vale 0 per JPEG (valore predefinito) e 1 per PNG; infine *targetWidth* e

targetHeight che sono da utilizzare insieme, ed è possibile inserire il numero di pixel per scalare l'immagine. L'app offre la possibilità di associare alla propria

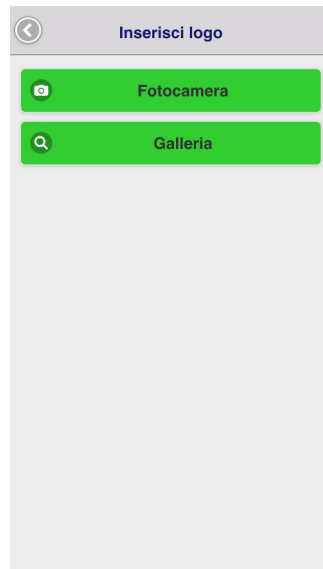


Figura 4.3: Scelta tra fotocamera e galleria

azienda il suo logo, o tramite fotocamera o accedendo alla galleria delle immagini (Figura 4.2), il quale poi sarà inserito nel rapportino finale di lavoro (capitolo 4.4).

4.2 Firma

La firma, utile per dare la possibilità al cliente di validare i dati inseriti dal tecnico e a quest'ultimo di confermare l'esecuzione dell'intervento, viene realizzata attraverso una funzione `Signature` di una classe javascript e l'elemento `canvas` di HTML5 (utilizzato per disegnare tramite scripting, di solito javascript). Verrà utilizzata la funzione `jSignature`[8] per creare il contenitore `<canvas>`[7] e disegnare effettivamente la grafica della firma. Sarà obbligatorio però creare un altro contenitore, normalmente in un `<div>`, dove verrà applicato il `canvas`, più specificatamente verrà creato:

```
<div id="sign"></div>
```

Nel file javascript relativo alla pagina HTML, all'apertura di quest'ultima, verrà inizializzata la funzione tramite il comando `$("#sign").jSignature()`, questo attiverà la funzione di signature sul div avente come `id=sign`, creando così il `canvas` e la possibilità di effettuare la firma. Per poter utilizzare la firma creata si farà uso

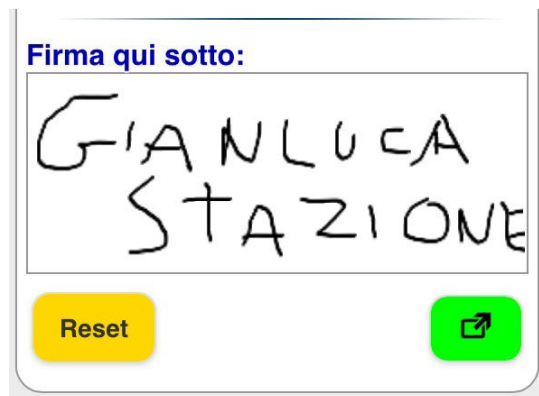


Figura 4.4: Esempio di firma

del comando `$("#sign").jSignature("getData")` che ritorna una stringa con codifica base64 (formato `data:image/png;base64,data`, quest'ultimo sono i dati nel formato ASCII che corrispondono all'immagine), che equivale all'immagine della firma. In caso si voglia rifare la firma sarà sufficiente cliccare su `reset`, il quale chiama la funzione `$("#sign").jSignature("reset")` e cancellerà il tutto.

Queste funzioni sono presenti sia nella creazione di un tecnico, sia nella pagina *Informazioni cliente*, ma a quest'ultimo è stata aggiunta una funzionalità, infatti, come si può notare dalla figura 4.4, vi è un pulsante verde che ci permette di aprire una nuova pagina a tutto schermo con funzionalità di `jSignature` per dare l'opportunità al cliente di avere uno spazio maggiore per apporre la propria firma.

4.3 Gestione multilingua

Assistenza Interventi Tecnici dà la possibilità di impostare (Figura 4.5) a piacere la lingua da utilizzare per l'intera navigazione, a scelta tra Italiano, Inglese, Francese, Tedesco, Spagnolo e Portoghese. L'opportunità viene data all'inizio dell'installazione dell'app, e successivamente nella pagina delle *impostazioni/cambio lingua*. In un file JSON verranno salvate tutte le traduzioni di ogni

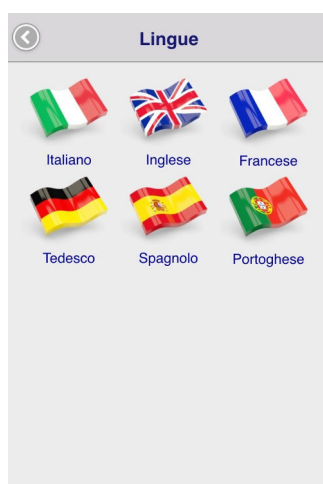


Figura 4.5: Impostazioni - Cambio lingua


pagina HTML così, a seconda della lingua selezionata, all'avvio della pagina viene caricato uno script javascript che legge il file json ed effettua le sostituzioni. La corrispondenza di ogni parola viene mantenuta con appositi id. Il file JSON si mostrerà così:

```
{
  "Traduzione": [
    {
      "id": "impostazioniHeader",
      "Italiano": "Impostazioni generali",
      "Inglese": "General settings",
      "Francese": "Réglages Généraux",
      "Tedesco": "Allgemeine Einstellungen",
      "Spagnolo": "Configuración General",
      "Portoghese": "Configurações Gerais"
    },
    ...
  ]
}
```

4.4 Rapportino di lavoro

Al fine di evitare la compilazione cartacea dell'intervento eseguito da parte del tecnico, si è pensato di creare direttamente il rapportino di lavoro (Figura 4.6), utilizzando le informazioni inserite dal tecnico nel momento della chiusura dell'intervento. Dopo la chiusura infatti, l'app esegue le seguenti operazioni:

1. Raccoglie tutti i dati dell'intervento (informazioni relative all'intervento e al cliente, prestazioni effettuate, merci e relative trasferte);
2. Carica la firma del cliente e del tecnico;
3. Carica le condizioni generali di fornitura;
4. Crea il pdf e lo invia per e-mail (verrà presentato nella sezione successiva);



Eurosystem2000
Via del mercato,1
41100 - Modena - (MO)
Telefono: 059123456
Fax: 059123456
E-mail: mail@es2000.it

Chiamata Nr: 4	Data: 08-04-2015 Ora: 10:00	Cliente: Azienda prova
Chiamata chiusa	Data: 08-04-2015 Ora: 9:36	Indirizzo: Viale campi 41100 - Modena - (MO) E-mail: mail@es2000.it Telefono: 059123456

Oggetto della chiamata:		Dettaglio chiamata	
Oggetto prova			
Descrizione della chiamata: Descrizione prova			

Prestazioni effettuate					
Prestazione	Quantità	Prezzo	Importo	Tecnico	Fatturazione
Assistenza Computer	H 2:30	47.80 €	108.75 €	Mario Rossi	Si
Descrizione: Descrizione prestazione					Totale prestazioni (Iva esclusa): 108.75 €

Merci utilizzate				
Descrizione	Quantità	Prezzo	Importo	Serial Number
Materiale prova	Nr 2.00	10.00 €	20.00 €	Si
Codice: 123 Note:Note sul materiale				Totale materiali (Iva esclusa): 20.00 €

Diritto di chiamata	Costo trasferta	km percorsi	Costo trasferta km
50,00 €	Fascia 1 (12,00 €)	150	48,00 €

MARIO
ROSSI

Tecnico: Mario Rossi

GIANLUCA
STAZIONE

Cliente: Gianluca Stazione

Condizioni generali di fornitura:
Condizioni di fornitura

Figura 4.6: Rapportino di intervento tecnico

Il rapportino è fondamentalmente diviso in otto parti. Nella prima, ovvero l'intestazione, è presente il logo aziendale (nel caso sia stato inserito, altrimenti rimarrà vuoto) e i dati principali dell'azienda (nome, indirizzo, telefono ed e-mail). Successivamente troviamo un riquadro in cui è presente il numero della chiamata e quando è stata aperta e chiusa, seguita dai dati del cliente (ed eventualmente seguita dai dati del cliente padre). La terza sezione invece raccoglie i dati della chiamata, ossia oggetto della chiamata e la sua descrizione (quelli inseriti nel momento della creazione della chiamata). La quarta e la quinta sezione invece sono tabelle che raccolgono le prestazioni effettuate e i materiali utilizzati e, nel caso non siano presenti, verrà mantenuto visualizzato il titolo della tabella e le intestazioni delle colonne. Dopodiché è presente la tabella delle trasferte, ed in caso una o più colonne non siano presenti, al contrario delle tabelle precedenti, verrà barrata la casella corrispondente. Per ultimo abbiamo la sezione delle firme del tecnico (a sinistra) e del cliente (a destra), il tutto termina con le condizioni generali di fornitura (nelle impostazioni sarà possibile specificare se inserirle o meno). Per creare questo pdf è stato utilizzato *jsPDF*[11], una libreria open-source che permette di generare file pdf utilizzando solamente javascript. Si crea così un oggetto della classe jsPDF:

```
var doc = new jsPDF();
```

Creata così una pagina pdf, sarà sufficiente utilizzare i metodi:

- *setFontSize(n)*: dimensione del font;
- *text(x,y,testo)*: per inserire del testo alle coordinate scelte;
- *addImage(img, tipo, x, y, width, height)*: inserire un'immagine (del tipo jpeg o png) alle coordinate e dimensioni inserite;
- *addPage()*: per inserire una nuova pagina;
- *roundedRect(x, y, width, height, r1, r2)*: per creare rettangoli con gli angoli arrotondati (r1 e r2 sono i raggi di curvatura).

Questi sono stati quelli più usati in questo progetto ovviamente ne esistono altri ancora. Una volta terminato si potrà utilizzare il metodo *output()*, più specificatamente:

```
var pdf = doc.output('datauri');
```

Avremo così il percorso del pdf, che verrà inserito come allegato nell'e-mail (ne parleremo nella sezione successiva).

4.5 Invio E-mail

Una volta conclusa l'assistenza, chiusa la chiamata (sezione chiusura) e creato il rapportino (capitolo 4.4), quest'ultimo verrà inviato tramite e-mail come allegato. La gestione dell'invio è gestita dalla funzione:

```
window.plugins.socialsharing.shareViaEmail(message, subject, mail, mailCC,  
mailBCC, files, onSuccess, onError)
```

Questa funzione chiama automaticamente l'applicazione delle e-mail del dispositivo su cui è installata l'app. Gli argomenti che vengono passati sono:

- *Message*: è una stringa e contiene il corpo della mail che si vuole scrivere;
- *Subject*: stringa che contiene l'oggetto della mail;
- *Mail,mailCC,mailBCC*: sono array che contengono gli indirizzi a cui inviare l'e-mail (CC e BCC sono la copia carbone normale e nascosta). In caso non si voglia metterli è obbligatorio scrivere null;
- *Files*: anch'esso è un array che contiene tutti i file che si vogliono allegare all'e-mail (null se non si invia niente). Nel nostro caso è il rapportino in pdf;
- *onSuccess,onError*: funzioni, la prima viene chiamata una volta che l'e-mail è stata inviata, mentre la seconda nel caso ci siano problemi con l'applicazione mail o l'invito dell'e-mail.



Figura 4.7: Screenshot- Apertura e-mail su iOS

4.6 Collegamento a Dropbox

Una caratteristica peculiare di quest'app è il collegamento a Dropbox[12] per cui il tecnico ha la possibilità, in questa versione gratuita, di caricare il backup dell'app su di esso.

Dropbox mette a disposizione un framework javascript offrendo così metodi per poter accedere al proprio dropbox. Prima di vedere in dettaglio i comandi da utilizzare, è necessario effettuare diverse operazioni preliminari. Oltre ad avere un account su dropbox, sarà indispensabile procedere nella sezione *developer*[13] (più precisamente in *App Console*) per creare l'integrazione tra l'app e Dropbox; questo produrrà una *APP_KEY*, una chiave alfanumerica che dovrà essere inserita nel file javascript per collegare l'app al proprio account developer. Questo procedimento però crea solamente una *testing app* con la possibilità di collegare solamente 100 utenti ai rispettivi account; per togliere questo limite sarà necessario richiedere, tramite mail, il passaggio da *testing app* a *business app*.

Una volta ottenuta la *app_key*, e l'account dell'app in versione business, passiamo a vedere i comandi da utilizzare. Per prima cosa si utilizza il comando:

```
var client = new Dropbox.Client(key:app_key);
```

Questo attiva il contatto con l'account developer (tramite l'utilizzo dell'*app_key* descritta in precedenza), mentre con:

```
client.authDriver(new Dropbox.AuthDriver.Cordova());
```

Essa servirà ad abilitare su Phonegap (che ricordiamo fa parte del progetto di Apache Cordova) il driver di autorizzazione dell'account dropbox, ovvero tramite *client.authenticate(auth_callback)* aprirà la pagina di autenticazione (Figura 4.8).

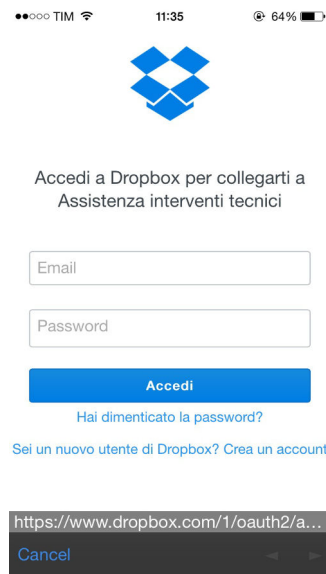


Figura 4.8: Pagina di autenticazione di Dropbox

Questo processo avviene una volta che nella pagina *Impostazioni* premiamo l'icona di collegamento, e dopo essersi autenticati si tornerà alla pagina principale delle impostazioni. Una volta autenticati, utilizzando sempre i comandi citati in precedenza, si potrà sempre comunicare con il proprio account senza dover effettuare nuovamente l'autenticazione.

4.6.1 Backup su Dropbox

Un'altra funzione dell'app è quella di fornire l'opportunità di effettuare backup, ovvero salvare i dati dei database locali, su dropbox.

Una volta premuto sull'icona *Backup*, il sistema provvederà ad estrarre i dati da tutti i database e a creare un array in cui salvarli. Appena l'array è stato creato, verrà utilizzato il seguente metodo presente nel tool javascript di dropbox:

```
client.writeFile(percorso,dati,function);
```

Dove *percorso* è una stringa contenente il percorso da seguire su dropbox (partendo ovviamente dalla radice) ed il nome del file stesso (*cartella/test.txt*), se per esempio volessimo salvare il file nella radice basterà semplicemente scrivere il nome del file (*test.txt*), *dati* corrisponde ad un array contenente ciò che si vuole inserire nel file, infine *function* è una funzione in cui si controlla l'andamento del metodo, ossia si gestisce cosa succede in caso di errore o in caso di successo. Come già discusso nell'introduzione, trattandosi della versione free si è data la possibilità

solo di salvare i backup, non di recuperarli, questa funzione sarà presente nella versione a pagamento.

Capitolo 5

Conclusioni e sviluppi futuri

Il progetto sperimentale ha portato alla realizzazione di un'app ibrida correttamente funzionante e disponibile sulle piattaforme Android ed iOS. Durante lo svolgimento del progetto sono state incontrate numerose difficoltà dovute principalmente alla eterogeneità dei dispositivi in gioco; è stato infatti necessario soffermarsi sulla parte grafica, in modo che le funzioni fossero ben visibili ed utilizzabili sia per smartphone di piccole dimensioni sia per tablet. Il superamento di tale difficoltà, tuttavia, ha permesso di ottenere un sistema particolarmente efficiente dal punto di vista grafica, riuscendo a realizzare un prodotto software in grado di essere facilmente fruibile. Questo sforzo sull'interfaccia grafica ha aperto un panorama nuovo sulle abitudini di lavoro e sulle possibilità di ottimizzazione dei processi aziendali; si pensi in primis all'enorme vantaggio di poter lavorare senza l'utilizzo di moduli cartacei e all'enorme risparmio di tempo nell'aver moduli già firmati ed inviati in tempo reale a tutti i destinatari.

È importante osservare come Phonegap si sia dimostrato all'altezza della situazione, rispondendo appieno a quelli che erano i requisiti iniziali del progetto; in particolar modo si è potuto sperimentare come il deploy dei package sulle varie piattaforme sia stato sempre in grado di convertire correttamente il codice sorgente (HTML5, Css e javascript) in applicazioni mobile. Si è osservato come l'utilizzo di questa tecnologia, curando molto gli aspetti grafici, si avvicini molto alla grafica e all'usabilità delle applicazioni native con l'enorme vantaggio però di poter scrivere il codice sorgente una sola volta e con un codice unico, anziché cominciare da capo per ogni piattaforma di sviluppo con linguaggi differenti. Infine da ricordare l'aiuto offerto dal software open-source Intel XDK per poter aumentare la velocità di scrittura delle app ibride tramite tool di visual editing. Come già accennato nell'introduzione, l'app sviluppata è una versione free, quindi la sua naturale evoluzione è rappresentata dalla versione a pagamento in cui vengono tolti i limiti imposti ed aggiunte varie funzioni.

Le limitazioni presenti nella versione free riguardano l'inserimento dei tecnici,

ossia è possibile salvarne solamente uno, mentre nella sezione clienti sarà possibile salvarne solo tre. Nelle impostazioni, invece, sarà possibile salvare i backup, ma non recuperarli. Queste limitazioni nella versione a pagamento saranno tolte, e sarà offerta la possibilità di recuperare i backup effettuati.

Le funzioni aggiuntive, invece, sono la possibilità di salvare o scaricare in formato .csv (file di testo utilizzato spesso da fogli elettronici come excel) l'azienda, i tecnici e i clienti, la presenza della pagina *Veicoli* (sezione *Azienda*) in cui si potranno salvare tutti i veicoli dell'azienda utilizzati per le trasferte, infine la presenza della pagina *Assegna chiamata* (*Home Page*) in cui si potrà gestire in modo più approfondito l'assegnazione delle chiamate, attribuendo più tecnici o collegare un particolare veicolo.

Un'ulteriore idea di sviluppo è quella di creare una versione in cui si agganci l'app ad un database centralizzato, permettendo l'accesso, tramite pagina di login, a più tecnici, in modo da formare una "squadra" di tecnici. Ogni tecnico così potrà vedere tutte le chiamate da assegnare ed accettare l'assegnazione, ovviamente potrà gestire solo le chiamate di cui è associato o è presente come membro della squadra. Infine sarà particolarmente interessante utilizzare anche le funzionalità della firma biometrica che alcuni smartphone di ultima generazione stanno iniziando a rendere disponibili per gestire l'autenticazione.

Bibliografia

- [1] Domenico Beneventano, Sonia Bergamaschi, Francesco Guerra , Maurizio Vincini *Progetto di Basi di Dati Relazionali*, Pitagora Editrice Bologna, 2007.
- [2] <http://www.xcubelabs.com/blog/native-web-and-hybrid-apps-understanding-the-difference>.
- [3] <http://chelramsey.com/phonegap-vs-native-builds-a-12-month-review/>.
- [4] <https://software.intel.com/it-it/html5/tools>.
- [5] <http://it.wikipedia.org/wiki/Sqlite>.
- [6] http://docs.phonegap.com/en/2.9.0/cordova_camera_camera.md.html#Camera.
- [7] <https://developer.mozilla.org/en-US/docs/Web/API/HTMLCanvasElement>.
- [8] <http://willowsystems.github.io/jSignature/#/about/>.
- [9] http://it.wikipedia.org/wiki/Eclipse_%28informatica%29.
- [10] <http://it.wikipedia.org/wiki/Xcode>.
- [11] <https://github.com/MrRio/jsPDF>.
- [12] <https://www.dropbox.com/>.
- [13] <https://www.dropbox.com/developers>.

Indice analitico

- ACID, 77
- Android, 8, 64, 67
- App ibrida, 63, 65
- App nativa, 65
- Appery, 68
- Apple, 76
- Application craft, 68
- Associazione, 59
- Attributi, 59

- Backup, 92

- Canvas, 86
- Codifica, 86
- CSS, 67
- Css, 7, 70

- DBMS, 77
- Debug, 74
- Deploy, 76, 95
- Dropbox, 91

- Eclipse, 76
- Emulate, 73
- Entità, 59

- Framework, 67, 71

- Html5, 7, 63, 65, 68, 70

- Intel XDK, 69
- iOS, 68, 76

- javascript, 7, 65, 67, 70
- jQuery Mobile, 63, 69, 71

- MySQL, 78

- Phonegap, 66, 68, 79

- SDK, 68, 76
- Signature, 86
- Software, 69, 76–78
- SQLite, 77

- Xcode, 76