**Parte 1 Abstract**

[Slide 1: Introduzione]

Salve a tutti, mi presento sono Andrea Bondanini.

La tesi da me realizzata concerne lo studio e lo sviluppo di un agente conversazionale applicato in ambiente sanitario, in un progetto realizzato in cooperazione con un’azienda del territorio, ovvero Onit Group.

Questa azienda svolge la propria attività nel settore dell’ICT in diversi segmenti di mercato, tra i quali quello medico.

La volontà di Onit di erogare i propri servizi attraverso l’utilizzo di nuove tecnologie mi ha permesso di poter lavorare alla progettazione di un chatbot chiamato OnAssistant, per studiare la fattibilità della progettazione, realizzazione e validazione di un assistente intelligente applicato in sanità.

Nel dettaglio è stato realizzato un prototipo che permette ad un paziente di comunicare attraverso una chat testuale, utilizzando il linguaggio naturale, ovvero quello che le persone utilizzano tra di loro nella comunicazione di tutti i giorni.

L’intento è quello di semplificare l’accesso alle prestazioni mediche erogate dal *Servizio Sanitario Nazionale* o SSN, che identifica il complesso delle funzioni e dei servizi assistenziali gestiti ed erogati dallo stato italiano.

Tempo:

**Parte 2 Introduzione del Problema**

[Slide 2: mostra possibilità attuali di prenotazione del SSN]

Attualmente la difficoltà maggiore di un cittadino che deve essere sottoposto a prestazioni mediche del SSN è rappresentata dalla possibilità d’accesso ad esse, dovuta a metodologie di prenotazioni obsolete e poco efficienti.

In Italia per richiedere un appuntamento è necessario rivolgersi al CUP (Centro Unico di Prenotazione) nelle seguenti modalità:

* Prenotazione allo sportello
  + è necessario presentarsi di persona nei vari uffici, spesso rimanere in fila svariati minuti e inoltre gli sportelli sono aperti per un numero limitato di giorni ed ore settimanali.
* Call center
  + risolvono la complicazione della prenotazione eseguita di persona, ma hanno diversi problemi per quanto riguarda l’efficienza: spesso si rimane in attesa diverso tempo al telefono senza riuscire a parlare con un operatore e si è costretti a richiamare.
  + Anche qui sono presenti vincoli temporali.
* CUP Online
  + è possibile accedere ai portali web messi a disposizione, tuttavia disponibili solamente per alcune regioni.
  + Inoltre presentano una eterogeneità di utilizzo con tipi di registrazioni e usabilità differenti tra loro.

L’Agenzia di Tutela della Salute di Milano ha analizzato le prenotazioni di prestazioni mediche del SSN ed è emerso che, nel 2018, più dell’80% delle visite mediche sono state fissate presentandosi di persona.

Tempo:

**Parte 3 Chatbot come soluzione? Analisi requisiti e criticità**

[Slide 3: Mostra vantaggi chatbot]

Perché è stato scelto un chatbot come soluzione? Con la realizzazione di OnAssistant ho cercato di implementare un sistema che potesse estrarre gli aspetti positivi delle modalità appena descritte tralasciandone le criticità. I vantaggi principali riguardano:

* Usabilità:
  + Data dalla facilità di utilizzo della chat in connubio con la possibilità di dialogare naturalmente.
  + Adatto a tutti i tipi di utente. Non è richiesta nessuna conoscenza tecnologica. È necessario emulare i passaggi effettuati quando si manda un messaggio ad un amico o parente.
* Praticità:
  + in produzione il sistema si interfaccerà con le applicazioni di messaggistica principali (Telegram, Facebook Messenger etc.)
* Efficienza:
  + abbiamo un sistema sempre funzionante, in ogni ora del giorno per tutti i giorni dell’anno.
  + Il paziente può eseguire l’operazione tranquillamente in ogni momento della sua giornata.

Durante l’analisi dei requisiti è stato importante valutare criticità del progetto e della sua applicazione, per individuare i vari ostacoli. In particolare:

* Prestazioni non prenotabili telematicamente: sebbene siano un numero limitato, alcune prestazioni richiedono obbligatoriamente l’intervento di un operatore per la prenotazione.

Tempo:

**Parte 4 Scenari utilizzo**

Come funziona OnAssistant e quali sono le funzionalità? Il bot offre:

* Possibilità di prenotare una prestazione medica in regime SSN.
* Sistema di FAQ, ovvero domande frequenti poste dagli utenti; per esempio possono richiedere informazioni su una visita specialistica.

Ora vediamo qualche scenario di utilizzo:

[Slide 4: Scenari di utilizzo]

**Scenario 1**

Per poter usufruire del servizio di prenotazione occorre conoscere il codice fiscale del paziente e il codice della ricetta elettronica, che è un documento elettronico rilasciato dal medico al quale sono associate prestazioni.

* Quando l’utente si unisce alla conversazione, il bot manda un messaggio di benvenuto dichiarando le funzionalità per cui è predisposto.
* L’utente richiede di prenotare un nuovo appuntamento.
* Inserisce il cf, il codice della ricetta.
* Vengono mostrate le prestazioni associate (es: visita dermatologica)
* A questo punto l’utente può inserire i filtri di ricerca
* Vengono mostrate le disponibilità in ordine cronologico con le info. (sede, data, prezzo …)
* Conferma prenotazione

**Scenario 2**

Qui viene riportato un esempio dove l’utente richiede informazioni su una radiografia. Quindi viene fornita una breve descrizione e dei consigli su come il paziente di deve preparare alla visita.

Tempo:

**Parte 5 Progettazione e sviluppo**

[Slide 5: Architettura generale bot framework]

Il prototipo è stato realizzato utilizzando una piattaforma fornita da Microsoft.

Bot Framework SDK permette la progettazione di bot intelligenti in ambiente .Net.

In questa slide è possibile vedere l’infrastruttura generale.

Canali di messaggistica utilizzabili

Dialog manager, componente che si occupa della gestione dello stato e del flusso della conversazione. Progetto che ho implementato concretamente, sviluppato in C#

Bot framework service è un servizio messo a disposizione da Microsoft che permette l’interfacciamento tra il bot e i canali di messaggistica.

Moduli LUIS che garantiscono l’intelligenza del sistema

Per tutti i processi di prenotazione, il sistema usa delle API di un altro software aziendale (On.Health). Per il primo utilizzo l’utente dovrà registrarsi su questo applicativo.

[Slide 6: Luis.ai (Intent e entities)]

LUIS (Language Understanding Intelligent Service)

Permette all’agente conversazionale di elaborare il linguaggio dell’utente, per comprendere il significato di una frase ed abilitarlo all’estrazione delle entità all’interno delle frasi stesse.

Questa tecnologia si basa su 3 concetti principali:

* Intent: rappresenta l’intenzione dell’utente quando invia un messaggio.
* Entities: strumento per l’estrazione delle parole chiave all’interno di una proposizione.
* Ruoli: descrivono il ruolo che assume una entità.

[Slide 7: moduli LUIS]

Questo framework mi ha permesso di creare dei moduli LUIS, ognuno formato da un certo numero di intents, entitites in base alle funzionalità che il bot deve gestire.

Il testo inserito dall’utente sarà processato da questi moduli e restituito al bot in formato JSON.

[Slide 8: stack dialog manager]

Il Dialog Manager si occupa della logica e del flusso della conversazione.

L’elemento concettualmente più interessante è la gestione dei Dialogs.

**Dialoghi**

Sono gli elementi base che costituiscono la conversazione. Sono classi C# che estendono l’interfaccia Dialog.

Sono presenti dei dialoghi base precostruiti, oppure è possibile eseguire l’override dei seguenti metodi per creare dialoghi personalizzati più complessi.

Come possiamo notare vengono gestiti attraverso uno stack, per indirizzare la conversazione a seconda degli input forniti dall’utente.

[Slide 9: architettura Bot framework]

Qui è possibile vedere in modo schematico l’architettura del framework.

Il Web Server elabora il body della chiamata POST

Il componente Adapter riceve l’attività, crea il contesto di turno che permette di mandare messaggi all’utente sul canale di messaggistica.

Le attività ricevute sono di due tipologie principali:

* ConversationUpdate: ricevuta dal bot quando un utente si unisce alla conversazione.
* Message: ricevuta quando l’utente effettivamente un messaggio.

Tempo:

**Parte 6 Validazione**

[Slide 10: slide validazione]

Il prototipo è stato testato da circa una ventina di miei colleghi esterni al progetto, per capire se un utilizzatore fosse fin dal primo utilizzo in grado di portare a termine autonomamente il flusso di prenotazione.

I Test sono stati eseguiti con il Bot Framework Emulator, ovvero uno strumento che emula una applicazione di messaggistica e permette di eseguire i test con il bot in esecuzione locale.

I test eseguiti hanno confermato questo, e le per le prime prenotazioni è stato impiegato un tempo massimo di due minuti, mentre per quelle successive, quindi dopo aver acquisito confidenza con il sistema, il tempo medio è stato di circa un minuto.

**Parte 7 Conclusioni**

[Slide 11: slide validazione]

Come anticipato nell’analisi dei requisiti la volontà era quella di implementare un sistema che facilitasse e velocizzate il paziente per la prenotazione di prestazioni mediche, e gli permettesse di comunicare naturalmente con il bot h24.

Oltre a questo, il bot è ampiamente estendibile, sia per quanto riguarda le funzionalità (poiché attualmente sto lavorando implementare un sistema di notifica proattivo del bot e realizzare un servizio per la gestione dei propri referti medici.), ma estendibile anche per quanto riguarda il mercato target.

Infatti Onit, oltre alla sanità pubblica presa in esame, offre servizi anche per quanto riguarda la sanità privata (cliniche private, poliambulatori etc) e OnAssistant è facilmente migrabile anche per questo settore di mercato.

Grazie per l’attenzione, ho concluso.