

# Ingegneria dei requisiti

## 1. Introduzione

**1.1 Obiettivo.** Questo documento stabilisce i requisiti di un'applicazione di messaggistica seguendo le normative imposte dalla AAC tramite l'utilizzo di pittogrammi. Questo documento serve anche come punto di partenza per la fase di design.

**1.2 Definizioni, acronimi e abbreviazioni.**

**AAC** (Augmentative Alternative Communication) si riferisce agli strumenti e alle strategie utilizzati per supportare o sostituire la parola per le persone con esigenze comunicative complesse

---

**CCN** (Complex Communication Needs Users)

---

**CP** (Communication Partners)

---

**1.4 Riferimenti.** Project Plan: App AAC, Documento 2: Gestione del progetto, Documento 3: Requisiti, Documento 4: Design, Documento 5: Testing, Documento 6: Maintenance

**1.5 Panoramica.** La sezione 2 del documento dà una panoramica generale sulla composizione dell'applicazione. La sezione 3 dà una descrizione più specifica sui requisiti per le funzioni offerte.

## 2. Descrizione panoramica

**2.1 Prospettiva del prodotto.** L'applicazione si appoggia a un database Firestore Database, dove vengono salvati in modo sicuro tutti i dati relativi alle chat tra gli utenti. Le credenziali di login invece si appoggiano sul servizio Authentication presente sempre sulla piattaforma di Firestore.

**2.2 Funzioni del prodotto.** Il sistema prevede due tipi di funzioni:

- Funzioni in cui gli utenti usano l'applicazione e si scambiano i messaggi. I requisiti specifici sono indicati nella sezione 3.1.1 e 3.1.4
- Funzioni in cui i dati vengono aggiornati e amministrati. Una descrizione più dettagliata è data nella sezione 3.1.2 e 3.1.3

**2.3 Caratteristiche dell'utente.** L'applicazione prevede la distinzione in tre categorie di utenti, con ruoli diversi

- Tutor (Sezione 3.2.1)
- CP (Sezione 3.2.2)
- CCN (Sezione 3.2.3)

**2.4 Vincoli.** Sono presenti dei vincoli per gli utenti di tipo CCN, con limitazioni sulle categorie di pittogrammi che sono autorizzati a usare e la creazione di nuove chat con

altre persone. Il database remoto di Firebase dispone di uno spazio limitato con la versione gratuita utilizzata per questo progetto.

### 3. Requisiti specifici

La fase di elicitazione è stata condotta attraverso interviste dirette con il committente (docenti) e l'analisi della documentazione fornita, inclusi i requisiti per progetti di ingegneria del software AA 2025/26.

#### 3.1 Requisiti dell'interfaccia esterna

**3.1.1 Interfaccia utente.** Per una rappresentazione grafica del flusso di navigazione tra le interfacce, si rimanda allo *State Machine Diagram* allegato nel Documento 4. Il sistema deve fornire interfacce utente distinte e adatte in base al ruolo (Tutor, CP, CCN) e allo stato dell'autenticazione, rispettando i seguenti requisiti funzionali di interfaccia:

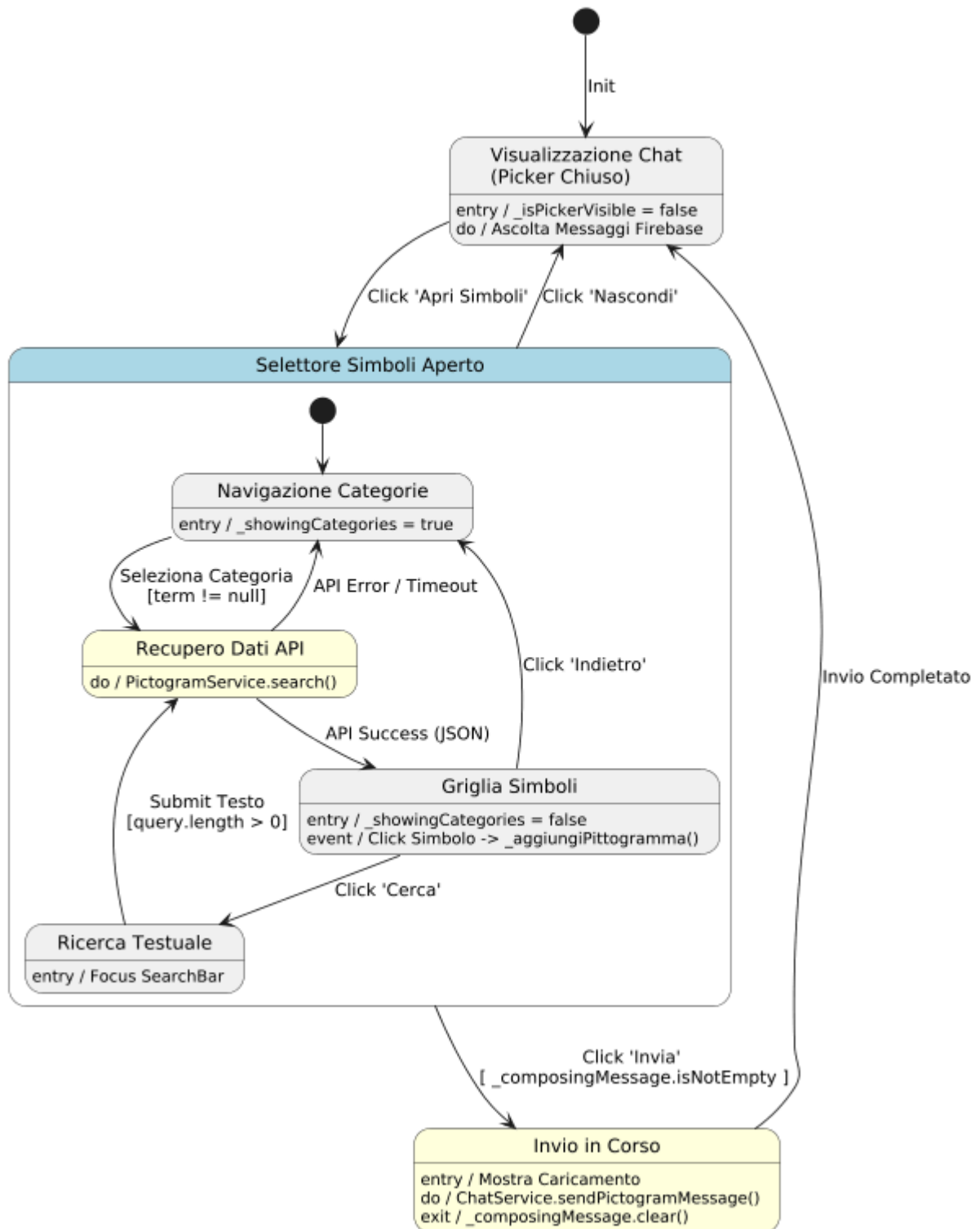
- Accesso e registrazione: Il sistema deve presentare, all'avvio in assenza di una sessione attiva, un'interfaccia di autenticazione che permetta all'utente di effettuare il login o di registrare un nuovo account (limitatamente ai ruoli Tutor e CP)
- Navigazione principale: Il sistema deve garantire una navigazione persistente tra le sezioni principali dell'applicazione tramite una barra di navigazione che consenta l'accesso rapido a:
  - Lista delle conversazioni attive
  - Pannello di gestione utenti CCN (visibile esclusivamente al ruolo Tutor)
  - Impostazioni dell'applicazione
- Lista chat e ricerca: L'interfaccia principale deve visualizzare l'elenco cronologico delle conversazioni esistenti. Il sistema deve fornire un meccanismo di ricerca testuale che permetta di filtrare le chat visibili in base al nome dell'interlocutore; in assenza di riscontri, il sistema deve fornire un feedback visivo di "nessun risultato"
- Chat: La schermata di conversazione deve permettere la visualizzazione dello storico dei messaggi tramite scorrimento verticale. L'utente deve avere la possibilità di interagire con i singoli messaggi:
  - Click sul messaggio: attivare la lettura audio del messaggio, il sintetizzatore deve leggere la descrizione dei pittogrammi che compongono il messaggio
  - Pressione prolungata sul messaggio:
    - *Caso utente proprietario della chat:* se l'azione avviene massimo 30 secondi dopo l'invio e il messaggio è stato inviato dall'utente: mostrare un avviso di conferma di eliminazione del messaggio, se l'utente procede alla eliminazione eliminare dal database il messaggio e mostrare un messaggio di conferma al termine. se l'azione avviene su un messaggio inviato dall'utente ma dopo i 30 secondi dall'invio allora mostrare un messaggio di impossibilità di eliminazione. Nel caso in cui il messaggio non sia stato inviato dall'utente non fare nulla.

- *Caso Tutor visualizza la chat di un suo CCN(non proprietario):*  
Restituire un messaggio di impossibilità di eliminazione dei messaggi

Per la composizione dei messaggi, il sistema deve fornire un selettore di input ( *Picker* ) espandibile che:

- Visualizzi i pittogrammi organizzati per categorie semantiche.
- Mostri una sezione di accesso rapido contenente gli ultimi pittogrammi utilizzati (cronologia locale).
- Permetta la selezione sequenziale di più pittogrammi prima dell'invio.
- Per ogni pittogramma selezionato mostrare una schermata di suggerimenti contenente i 3 pittogrammi, (cliccabili) più recenti utilizzati in passato dopo quello selezionato ed aggiorni i suggerimenti del pittogramma precedente aggiungendo il pittogramma selezionato come suggerimento più recente (*esempio: Pittogramma Saluto ha come suggerimenti: 1. Persona, 2. Uomo, 3. Donna. Dopo aver selezionato il pittogramma Saluto verranno mostrati i suggerimenti 1. Persona, 2. Uomo, 3. Donna. Se si seleziona il pittogramma Amico, il sistema provvederà ad aggiornare i suggerimenti per Saluto in: 1. Amico, 2. Persona, 3. Uomo) e mostrerà i suggerimenti per Amico, se non presenti non compariranno.*

### Macchina a Stati: Composizione Messaggio CAA (ChatPage)



- Pagina impostazioni e accessibilità: Il sistema deve fornire un'interfaccia di configurazione divisa in tre sezioni logiche:

1. **Accessibilità CAA:** Deve permettere la regolazione della dimensione della griglia dei pittogrammi e l'attivazione/disattivazione delle etichette testuali (label) sotto le immagini.
  2. **Sintesi Vocale (TTS):** Deve permettere l'abilitazione della lettura automatica dei messaggi e la regolazione della velocità di riproduzione.
  3. **App & Sistema:** Deve consentire la commutazione del tema visivo (Chiaro/Scuvo).
- Pagina di gestione utenti CCN: L'interfaccia dedicata ai Tutor deve permettere la visualizzazione della lista degli utenti CCN associati. Per ogni utente gestito, il sistema deve fornire controlli per:
- Modificare le categorie di pittogrammi visibili (abilitazione/disabilitazione).
  - Registrare un nuovo account CCN (funzionalità "Nuovo CCN").
  - Eliminare o disattivare un account esistente.
  - Visualizzare le chat del CCN

3.1.2 *Interfacce hardware.* L'interfaccia utente è screen-oriented

3.1.3 *Interfacce software.* L'interfaccia con il software avviene sulla piattaforma cloud di Firestore Firebase. I dati di autenticazione sono archiviati in modo sicuro sulla piattaforma Authentication già presente all'interno di Firebase. Tutte le chat sono salvate su Firestore Database all'interno della collezione "chats", dove si trova l'id degli utenti a cui la chat è visibile. Tutti i dati di personalizzazione e il ruolo dell'utente è salvato su Firestore Database all'interno della collezione "users" sotto l'id specifico dedicato all'utente. Per il salvataggio dei suggerimenti dei pittogrammi per ogni singolo utente sarà presente un file in locale nella cartella privata dell'utente chiamato suggerimenti.json, accessibile tramite operazioni I/O.

3.1.4 Interfacce di comunicazione

### 3.2 *Richieste funzionali*

#### 3.2.1 *Tutor*

3.2.1.1 *Aggiunta di nuovi utenti CCN.* Il tutor deve essere in grado di poter registrare un nuovo utente CCN mediante l'utilizzo di nome, cognome, email e password

3.2.1.2 *Gestione degli utenti CCN.* Un tutor deve essere in grado tramite lista di selezionare l'utente CCN che segue e poter modificare le categorie di pittogrammi che quell'utente CCN sarà in grado di utilizzare. Il tutor può anche disabilitare l'account dell'utente CCN selezionato e questo comporterà un logout immediato dall'applicazione finché non verrà riattivato l'account da parte del tutor.

3.2.1.3 *Registrazione come tutor.* Un utente tutor in fase di registrazione deve poter specificare il suo ruolo così da poter accedere poi alla schermata per gestire gli utenti CCN a lui associati

3.2.1.4 *Creazione nuova chat.* L'utente tutor può creare una nuova chat tramite l'apposito bottone situato in basso a destra che gli consente di scegliere l'utente con la quale comunicare

3.2.1.5 *Controllare richieste per una nuova chat dell'utente CCN.* Il tutor deve poter essere in grado di gestire le richieste in entrata di creazione di una nuova chat ricevute dall'utente CCN selezionato

*3.2.1.6 Creazione nuova chat tra utente CCN e un altro utente.* Il tutor selezionando un utente CCN da lui gestito deve poter creare una nuova chat tra lui e un altro utente.

*3.2.1.7 Controllo chat CCN.* Un utente tutor deve essere in grado di controllare le chat dei propri utenti CCN che gestisce senza aver possibilità di inviare o cancellare i messaggi nelle chat che visualizza.

### **3.2.2 CP**

*3.2.2.1 Creazione di una nuova chat.* Un utente CP deve poter creare una nuova chat con un altro utente (tutor o CCN).

*3.2.2.2 Scheda gestione CCN nascosta.* Un utente di tipo CP non deve essere in grado di accedere o visualizzare la pagina di gestione degli utenti CCN.

### **3.2.3 CCN**

*3.2.3.1 Creazione nuova chat.* Un utente CCN non deve essere in grado di visualizzare il pulsante per la creazione di una nuova chat

*3.2.3.2 Visualizzazione griglia pittogrammi.* l'utente CCN può visualizzare solamente le categorie di pittogrammi selezionate dal suo tutor

*3.2.3.3 Impostazioni accessibilità.* Un utente CCN deve essere in grado di accedere e modificare le impostazioni per l'ingrandimento della griglia dei pittogrammi e per l'attivazione della sintesi vocale

# Qualità del software

In questa sezione vengono definiti gli attributi di qualità del software che hanno guidato lo sviluppo del progetto "ChatByPics" e la metodologia organizzativa adottata per garantirli.

## Attributi di Qualità del Prodotto (ISO/IEC 25010)

Per la natura specifica dell'applicazione, destinata all'ambito della Comunicazione Aumentativa Alternativa (AAC), sono stati privilegiati i seguenti attributi di qualità:

- **Usabilità (Usability):** Dato il target di utenza che include persone con bisogni comunicativi complessi (CCN), l'usabilità è il requisito non funzionale primario. L'interfaccia è stata progettata per essere intuitiva, riducendo il carico cognitivo tramite l'uso predominante di immagini (pittogrammi ARASAAC) rispetto al testo. La navigazione è semplificata e priva di elementi di distrazione.
- **Sicurezza (Security):** L'applicazione gestisce dati sensibili relativi a minori o pazienti. Per garantire la riservatezza e l'integrità dei dati:
  - L'autenticazione è gestita tramite **Firebase Authentication**, che utilizza protocolli standard sicuri.
  - L'accesso ai dati nel database è regolato da regole di sicurezza (Firestore Security Rules) che impediscono l'accesso non autorizzato ai profili e alle chat private.
- **Affidabilità (Reliability):** Il sistema deve garantire continuità nel servizio di comunicazione. L'utilizzo di un backend *serverless* come Firebase garantisce un'alta disponibilità (High Availability) e riduce il rischio di *downtime* del server, assicurando che il servizio sia accessibile quando necessario.
- **Manutenibilità (Maintainability):** Il codice è stato scritto in **Dart/Flutter** seguendo un'architettura modulare. La separazione tra la logica di business (es. gestione stati, servizi Firebase) e l'interfaccia utente (Widget) permette aggiornamenti futuri e correzione di bug con un impatto minimo sulle altre parti del sistema.

## Garanzia della Qualità tramite il modello SWAT

Per garantire il raggiungimento degli obiettivi di qualità sopra descritti, il team di sviluppo ha adottato il modello organizzativo **SWAT (Skilled With Advanced Tools)**. Questa scelta ha influenzato positivamente la qualità del software nei seguenti modi:

1. **Utilizzo di Strumenti Avanzati (Advanced Tools):** Come previsto dal modello SWAT, il team ha fatto uso di strumenti di alto livello e generatori potenti per massimizzare l'efficienza e ridurre gli errori manuali:
  - **Flutter:** Framework cross-platform che ha permesso di sviluppare un'unica codebase di alta qualità per Android e iOS.
  - **Firebase:** Ha agito come acceleratore di sviluppo backend, fornendo database in tempo reale e autenticazione "out-of-the-box", permettendo al team di concentrarsi sulla qualità dell'interfaccia utente.
  - **GitHub:** Utilizzato per il versionamento e la code review continua.

2. **Sviluppo Incrementale:** Il team ha lavorato creando **versioni incrementali del sistema software**. Invece di rilasciare tutto alla fine, sono stati sviluppati e testati moduli separati (es. prima il Login, poi la Lista Chat, infine la Gestione CCN). Questo approccio iterativo ha permesso di identificare e risolvere i difetti di qualità nelle prime fasi, riducendo il costo delle correzioni.
3. **Comunicazione e Ruoli:** Essendo una squadra piccola, i canali di comunicazione sono stati mantenuti **molto brevi** e informali, permettendo un allineamento costante sui requisiti di qualità. I membri del team hanno agito sia come sviluppatori che come manager di se stessi, partecipando attivamente alle sessioni di *brainstorming* per risolvere problemi critici di usabilità e architettura.