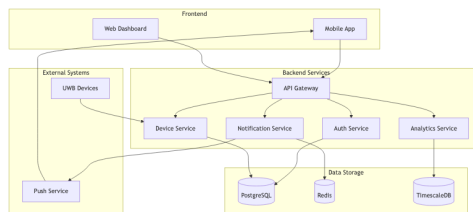


# MATE HOME 2

## Analisi Funzionale Dettagliata



December 28, 2024

# Contents

1	MATE HOME 2 - Analisi Funzionale Dettagliata	2
1.1	1. Panoramica del Sistema	2
1.1.1	1.1 Architettura del Sistema	2
1.1.2	1.2 Flusso dei Dati	2
1.2	2. Funzionalità Core	3
1.2.1	2.1 Dashboard Principale	3
1.2.2	2.2 Sistema di Notifiche	4
1.2.3	2.3 Monitoraggio Bagno	4
1.2.4	2.4 Gestione Dispositivi	5
1.3	3. Aspetti Tecnici	5
1.3.1	3.1 Sicurezza	5
1.3.2	3.2 Performance	6
1.4	4. Roadmap di Sviluppo	6
1.4.1	4.1 Fase Attuale	6
1.4.2	4.2 Prossimi Sviluppi	7
1.5	5. Note Implementative	7
1.5.1	5.1 Stack Tecnologico	7
1.5.2	5.2 Best Practices	7

## Architettura del Sistema

Figure 1: Architettura del Sistema

# 1 MATE HOME 2 - Analisi Funzionale Dettagliata

## 1.1 1. Panoramica del Sistema

MATE HOME 2 è un sistema avanzato di monitoraggio assistenziale progettato per offrire un supporto completo nella cura e nell'assistenza di persone anziane o che necessitano di supervisione costante. Il sistema si distingue per la sua capacità di fornire informazioni in tempo reale sullo stato e le attività dell'assistito, garantendo al contempo privacy e dignità.

### 1.1.1 1.1 Architettura del Sistema

L'architettura del sistema è stata progettata secondo i principi di scalabilità, affidabilità e sicurezza, articolandosi in tre componenti principali:

**App Mobile (iOS/Android)** L'applicazione mobile rappresenta l'interfaccia principale per caregiver e familiari. È stata sviluppata utilizzando React Native per garantire: - **Prestazioni Native:** L'app offre un'esperienza fluida e reattiva su entrambe le piattaforme - **Interfaccia Intuitiva:** Design user-friendly che permette un accesso immediato alle informazioni critiche - **Notifiche in Tempo Reale:** Sistema di notifiche push per eventi significativi - **Monitoraggio Continuo:** Visualizzazione dello stato attuale e storico delle attività

**Backend Server** Il cuore del sistema, responsabile dell'elaborazione dati e della logica di business: - **Microservizi:** Architettura modulare per maggiore scalabilità e manutenibilità - **API Gateway:** Punto di accesso unificato e sicuro per tutte le richieste - **Sistema di Notifiche:** Elaborazione e distribuzione intelligente delle notifiche - **Gestione Dati:** Storage sicuro e efficiente dei dati sensibili

**Sensori e Dispositivi** La rete di sensori radar UWB (Ultra-Wide Band) fornisce: - **Rilevamento Presenza:** Monitoraggio non invasivo della presenza nelle stanze - **Analisi Movimento:** Identificazione di pattern di movimento e potenziali anomalie - **Parametri Ambientali:** Monitoraggio di temperatura, umidità e luminosità - **Privacy Preservata:** Nessuna telecamera o dispositivo invasivo

### 1.1.2 1.2 Flusso dei Dati

Il flusso dei dati è stato ottimizzato per garantire: - **Latenza Minima:** Tempo di risposta inferiore a 200ms per le informazioni critiche - **Affidabilità:** Sistema di retry automatico in

## Flusso Dati

Figure 2: Flusso Dati

caso di errori di comunicazione - **Sicurezza:** Crittografia end-to-end di tutti i dati sensibili - **Efficienza:** Ottimizzazione del consumo di banda e risorse

## 1.2 2. Funzionalità Core

### 1.2.1 2.1 Dashboard Principale

La dashboard è il punto di accesso principale alle informazioni del sistema. È stata progettata per fornire una visione immediata e completa dello stato dell'assistito.

#### Componenti Principali

**Status Card** Visualizza lo stato attuale dell'assistito:

```
1 interface StatusCard {
2   presence: boolean;           // Presenza nella stanza
3   activity: ActivityLevel;     // Livello di attività
4   lastMovement: Date;         // Ultimo movimento rilevato
5   environmentals: {
6     temperature: number;      // Temperatura ambiente
7     humidity: number;         // Umidità relativa
8     light: number;            // Livello di illuminazione
9   }
10 }
```

**Comportamento:** - Aggiornamento in tempo reale (polling ogni 30 secondi) - Indicatori visivi per cambiamenti di stato - Animazioni fluide per transizioni - Gesture per interazioni rapide

**Activity Timeline** Cronologia delle attività nelle ultime 24 ore:

```
1 interface ActivityEvent {
2   type: 'movement' | 'rest' | 'absence';
3   startTime: Date;
4   duration: number;
5   intensity?: number;
6 }
```

**Visualizzazione:** - Grafico interattivo con zoom e pan - Codifica colori per diversi tipi di attività - Dettagli on-demand con tap - Filtri per tipo di attività

### 1.2.2 2.2 Sistema di Notifiche

Il sistema di notifiche è stato progettato per mantenere i caregiver informati sugli eventi significativi, minimizzando al contempo il rischio di “alert fatigue”.

#### Tipi di Notifica

##### Notifiche di Movimento

```
1 interface MovementAlert {  
2   type: 'movement';  
3   severity: 'info' | 'warning' | 'critical';  
4   location: Room;  
5   timestamp: Date;  
6   description: string;  
7   metadata: {  
8     intensity: number;  
9     duration: number;  
10    pattern: string;  
11  }  
12 }
```

**Logica di Generazione:** - Analisi continua dei pattern di movimento - Rilevamento anomalie basato su ML - Contestualizzazione temporale (giorno/notte) - Personalizzazione per utente

##### Notifiche Ambientali

```
1 interface EnvironmentalAlert {  
2   type: 'environmental';  
3   parameter: 'temperature' | 'humidity' | 'light';  
4   value: number;  
5   threshold: {  
6     min: number;  
7     max: number;  
8   };  
9   timestamp: Date;  
10 }
```

**Gestione:** - Monitoraggio continuo dei parametri - Soglie configurabili - Aggregazione intelligente - Prioritizzazione automatica

### 1.2.3 2.3 Monitoraggio Bagno

Il modulo di monitoraggio bagno è stato sviluppato con particolare attenzione alla privacy e alla sicurezza dell'assistito.

#### Funzionalità Principali

### Rilevamento Occupazione

```

1 interface BathroomStatus {
2     occupied: boolean;
3     duration: number;
4     lastVisit: {
5         start: Date;
6         end: Date;
7         duration: number;
8     };
9     analytics: {
10         dailyVisits: number;
11         averageDuration: number;
12         unusualPatterns: boolean;
13     }
14 }

```

**Caratteristiche:** - Rilevamento non invasivo - Analisi temporale delle visite - Identificazione pattern anomali - Alert per durate eccessive

**Statistiche e Analisi** Il sistema mantiene statistiche dettagliate per: - Frequenza delle visite - Durata media - Pattern giornalieri - Anomalie comportamentali

#### 1.2.4 2.4 Gestione Dispositivi

Il modulo di gestione dispositivi garantisce il corretto funzionamento dell'infrastruttura hardware.

### Monitoraggio Sensori

```

1 interface DeviceStatus {
2     id: string;
3     type: 'radar' | 'environmental';
4     battery: number;
5     signal: number;
6     lastHeartbeat: Date;
7     errors: DeviceError[];
8     configuration: DeviceConfig;
9 }

```

**Funzionalità:** - Monitoraggio stato batteria - Qualità del segnale - Diagnostica remota - Aggiornamenti OTA

## 1.3 3. Aspetti Tecnici

### 1.3.1 3.1 Sicurezza

La sicurezza è una priorità fondamentale del sistema:

## Autenticazione

- OAuth 2.0 con refresh token
- Biometria opzionale
- Session management
- Rate limiting

## Protezione Dati

- Crittografia end-to-end
- Anonimizzazione
- Backup automatici
- Audit logging

### 1.3.2 3.2 Performance

Obiettivi di performance:

## Frontend

- First Load: < 2s
- Time to Interactive: < 1s
- Frame Rate: 60fps
- Offline Support

## Backend

- Latenza API: < 200ms
- Uptime: 99.9%
- Throughput: 1000 req/s
- Auto-scaling

## 1.4 4. Roadmap di Sviluppo

### 1.4.1 4.1 Fase Attuale

- Implementazione core features
- Testing con utenti pilota
- Ottimizzazione performance
- Documentazione tecnica

### 1.4.2 4.2 Prossimi Sviluppi

- Integrazione ML per pattern recognition
- Espansione analytics
- API pubbliche
- Mobile app 2.0

## 1.5 5. Note Implementative

### 1.5.1 5.1 Stack Tecnologico

- **Frontend:** React Native, TypeScript, Redux
- **Backend:** Node.js, Express, PostgreSQL
- **Infrastructure:** AWS, Docker, Kubernetes
- **Monitoring:** ELK Stack, Prometheus

### 1.5.2 5.2 Best Practices

- Clean Architecture
- Test-Driven Development
- Continuous Integration
- Documentation as Code