

「αode」



Analisi dei requisiti

2025-03-10

| | |
|--------------|---------------|
| Responsabile | Nicolò Bovo |
| Redattori | Romeo Calero |
| Verificatori | Elia Leonetti |

Registro delle modifiche

| Vers. | Data | Descrizione | Autore | Verificatore |
|-------|------------|--|--------------|---------------|
| 0.2.0 | 2025-03-17 | Aggiunta sezione casi d'uso e architettura | Romeo Calero | Elia Leonetti |
| 0.1.0 | 2025-03-10 | Bozza del documento | Romeo Calero | Elia Leonetti |

Indice

| | |
|--|----------|
| I. Introduzione | 5 |
| I - 1. Scopo del documento | 5 |
| I - 2. Scopo del prodotto | 5 |
| I - 3. Riferimenti | 5 |
| I - 3.1. Riferimenti tecnici | 5 |
| I - 3.2. Riferimenti implementativi | 5 |
| II. Descrizione del prodotto | 5 |
| II - 1. Obiettivi del prodotto | 5 |
| II - 1.1. Obiettivi primari: | 6 |
| II - 1.2. Obiettivi secondari: | 6 |
| II - 2. Funzionalità e vincoli del prodotto | 6 |
| II - 2.1. Funzionalità implementate: | 6 |
| II - 2.2. Vincoli tecnici: | 7 |
| II - 3. Utenti e loro caratteristiche | 7 |
| II - 3.1. Tipologie di utenti: | 7 |
| III. Architettura del sistema | 7 |
| III - 1. Componenti Core | 7 |
| III - 2. Flusso Dati | 8 |
| IV. Casi d'uso | 8 |
| IV - 1. Introduzione | 8 |
| IV - 2. Attori | 8 |
| IV - 3. Elenco dei casi d'uso | 9 |
| IV - 3.1. UC1 - Autenticazione utente alla dashboard | 9 |
| IV - 3.2. UC2 - Inserimento Username | 9 |
| IV - 3.3. UC3 - Inserimento Password | 9 |
| IV - 3.4. UC4 - Visualizzazione Errore Credenziali | 10 |
| IV - 3.5. UC5 - Visualizzazione Dashboard | 10 |
| IV - 3.6. UC6 - Visualizzazione profilo utente nella dashboard | 11 |
| IV - 3.7. UC7 - Visualizzazione statistiche utente nella dashboard | 11 |
| IV - 3.8. UC8 - Visualizzazione promozioni recenti nella dashboard | 11 |
| IV - 3.9. UC9 - Filtra punti di interesse per categoria | 12 |
| IV - 3.10. UC10 - Messaggio di errore «Nessun risultato trovato» | 12 |

Lista delle figure

Lista delle tabelle

I. Introduzione

I - 1. Scopo del documento

Il presente documento di Analisi dei Requisiti fornisce una descrizione dettagliata e completa del sistema **NearYou**, una piattaforma di notifiche personalizzate basate sulla posizione che utilizza intelligenza artificiale generativa (AI generativa) per creare messaggi contestuali e rilevanti per gli utenti.

Il documento costituisce la base per lo sviluppo del sistema, definendo chiaramente funzionalità, vincoli, casi d'uso e requisiti tecnici derivanti dall'analisi del codice sorgente e dell'architettura implementata.

I - 2. Scopo del prodotto

NearYou è una piattaforma tecnologica che combina **tracking della posizione in tempo reale**, **database geospaziali**, **intelligenza artificiale generativa** e **analytics avanzate** per fornire notifiche personalizzate agli utenti quando si trovano in prossimità di negozi o punti di interesse.

Il sistema è progettato per:

- **Tracciare la posizione** degli utenti in tempo reale attraverso simulazione GPS
- **Generare messaggi personalizzati** utilizzando LLM (Large Language Models) quando l'utente è vicino a un negozio
- **Fornire una dashboard interattiva** con visualizzazione su mappa delle posizioni, percorsi e notifiche
- **Memorizzare e analizzare** tutti i dati per fornire insights attraverso dashboard analytics
- **Aggiornare automaticamente** il database dei negozi tramite ETL da OpenStreetMap

I - 3. Riferimenti

I - 3.1. Riferimenti tecnici

- Docker Compose: `docker-compose.yml`
- Configurazione ambiente: `.env`
- Documentazione API: FastAPI autodocs (`/docs`)
- Database schema: Script di inizializzazione ClickHouse e PostgreSQL

I - 3.2. Riferimenti implementativi

- Repository GitHub del progetto
- Dockerfile e script di deployment
- Configurazioni di monitoraggio (Grafana, Prometheus)

II. Descrizione del prodotto

II - 1. Obiettivi del prodotto

Il sistema NearYou ha come obiettivo principale la **creazione di un'esperienza utente personalizzata e contestuale** attraverso l'integrazione di diverse tecnologie avanzate:

II - 1.1. Obiettivi primari:

1. **Personalizzazione intelligente:** Utilizzare profili utente (età, professione, interessi) per generare messaggi rilevanti tramite LLM
2. **Geolocalizzazione precisa:** Implementare tracking in tempo reale con calcolo di distanze attraverso PostGIS
3. **Scalabilità e performance:** Gestire flussi di dati in tempo reale attraverso Apache Kafka e Bytewax
4. **Analytics avanzate:** Fornire insights attraverso dashboard Grafana con metriche di business e sistema
5. **Automazione:** Mantenere aggiornato il database negozi attraverso ETL automatici da OpenStreetMap

II - 1.2. Obiettivi secondari:

- **Caching intelligente:** Ridurre latenza e costi LLM attraverso cache Redis
- **Monitoraggio completo:** Garantire osservabilità del sistema con Prometheus, Loki e metriche custom
- **Facilità di deployment:** Fornire ambiente containerizzato completo con Docker Compose

II - 2. Funzionalità e vincoli del prodotto

II - 2.1. Funzionalità implementate:

Data Pipeline e Streaming:

- Producer Kafka che simula movimenti utenti su percorsi reali di Milano usando OSRM
- Consumer Bytewax per elaborazione stream dati con arricchimento in tempo reale
- Integrazione con PostGIS per calcolo distanze e identificazione negozi vicini

Generazione Messaggi Intelligente:

- Servizio microservice FastAPI per generazione messaggi personalizzati
- Integrazione con LLM (supporto Groq, OpenAI, provider multipli)
- Sistema di cache Redis per ottimizzazione performance e riduzione costi
- Template personalizzabili basati su profilo utente e tipologia negozio

Dashboard Utente:

- Interfaccia web responsive con mappa Leaflet interattiva
- Visualizzazione real-time di posizione utente, percorsi e negozi
- Sistema di notifiche in-app per messaggi personalizzati
- Autenticazione JWT con gestione sessioni sicura
- WebSocket per aggiornamenti posizione in tempo reale

Storage e Persistenza:

- ClickHouse per analytics e storage eventi temporali ad alta performance
- PostgreSQL/PostGIS per gestione dati geospaziali dei negozi
- Schema ottimizzato per query geospaziali e analytics

ETL e Data Management:

- Apache Airflow per orchestrazione ETL automatici
- ETL adattivo che si auto-configura in base ai cambiamenti nei dati

- Integrazione con OpenStreetMap per aggiornamento negozi Milano
- Sistema di tracking cambiamenti con logging completo

Monitoring e Observability:

- Dashboard Grafana con pannelli personalizzati per metriche business e sistema
- Prometheus per raccolta metriche con esportatori per tutti i componenti
- Loki per gestione log centralizzata
- Metriche custom per tracking performance LLM, cache hit rate, eventi utente

II - 2.2. Vincoli tecnici:

- **Geografici:** Sistema focalizzato su Milano (coordinate definite in configurazione)
- **Performance:** Target <200ms per generazione messaggi (con cache)
- **Scalabilità:** Supporto multi-worker Bytewax per elaborazione parallela
- **Sicurezza:** Kafka con SSL/TLS, JWT per autenticazione, password protette
- **Disponibilità:** Health checks e restart automatici per tutti i servizi

II - 3. Utenti e loro caratteristiche

II - 3.1. Tipologie di utenti:

Utente finale:

- **Descrizione:** Persona che utilizza l'applicazione per ricevere notifiche personalizzate
- **Caratteristiche:** Ha un profilo definito (età, professione, interessi) memorizzato nel sistema
- **Interazioni:** Accede alla dashboard, visualizza la mappa, riceve notifiche, naviga lo storico
- **Competenze richieste:** Utilizzo base di applicazioni web e mobile

Amministratore di sistema:

- **Descrizione:** Operatore responsabile del monitoraggio e manutenzione del sistema
- **Caratteristiche:** Competenze tecniche per interpretare metriche e log
- **Interazioni:** Accesso a Grafana, Prometheus, log Loki, configurazione Airflow
- **Competenze richieste:** Conoscenza sistemi distribuiti, database, monitoring

Sviluppatore/DevOps:

- **Descrizione:** Persona che sviluppa, deploy e mantiene il sistema
- **Caratteristiche:** Competenze tecniche avanzate su architetture microservizi
- **Interazioni:** Deploy con Docker Compose, configurazione environment, debug
- **Competenze richieste:** Docker, Kafka, database, cloud native technologies

III. Architettura del sistema

Il sistema NearYou implementa un'**architettura microservizi event-driven** con i seguenti componenti principali:

III - 1. Componenti Core

Streaming Data Pipeline:

- **Apache Kafka:** Message broker per streaming eventi GPS con SSL
- **Bytewax:** Stream processor per elaborazione real-time con supporto multi-worker

- **Producer:** Simulatore movimenti utenti su percorsi OSRM Milano
- **Consumer:** Processore eventi con arricchimento dati e generazione messaggi

API Services:

- **Message Generator:** Microservizio FastAPI per generazione messaggi LLM
- **Dashboard Service:** Servizio FastAPI per dashboard utente con WebSocket
- **Authentication:** Sistema JWT per gestione autenticazione e autorizzazione

Storage Layer:

- **ClickHouse:** Database colonnare per analytics e storage eventi temporali
- **PostgreSQL/PostGIS:** Database relazionale con estensioni geospaziali per negozi
- **Redis:** Cache distribuita per messaggi LLM e sessioni

ETL e Data Management:

- **Apache Airflow:** Orchestratore per ETL automatici da OpenStreetMap
- **OSRM:** Servizio routing per calcolo percorsi realistici su Milano

Monitoring Stack:

- **Grafana:** Dashboard per visualizzazione metriche business e sistema
- **Prometheus:** Raccolta metriche con esportatori specializzati
- **Loki:** Gestione log centralizzata
- **Exporters:** Componenti per esposizione metriche database e servizi

III - 2. Flusso Dati

1. **Generazione Eventi:** Producer simula movimenti utenti su percorsi OSRM
2. **Stream Processing:** Bytewax consumer elabora eventi, arricchisce con dati negozi
3. **Proximity Detection:** Calcolo distanze con PostGIS, trigger per messaggi
4. **Message Generation:** Chiamata servizio LLM con cache Redis
5. **Storage:** Persistenza eventi arricchiti in ClickHouse
6. **Real-time Updates:** Notifica utenti via WebSocket
7. **Analytics:** Visualizzazione dati in dashboard Grafana

IV. Casi d'uso

IV - 1. Introduzione

I casi d'uso del sistema NearYou sono strutturati attorno alle funzionalità core implementate nel codice sorgente. Ogni caso d'uso è identificato da un codice univoco UC[numero] e descrive le interazioni tra attori e sistema.

IV - 2. Attori

Utente Autenticato: Persona che utilizza la dashboard per visualizzare la propria posizione e ricevere notifiche personalizzate, avendo eseguito l'accesso.

Utente non Autenticato: Utente che non ha ancora eseguito l'accesso.

Amministratore: Persona che usa la dashboard per visualizzare le analisi di marketing e di monitoraggio del sistema.

IV - 3. Elenco dei casi d'uso

IV - 3.1. UC1 - Autenticazione utente alla dashboard

Attore principale:

- Utente non autenticato

Precondizioni:

- L'utente ha credenziali valide nel sistema

Postcondizioni:

- L'utente è autenticato e ha accesso alla dashboard

Scenario principale:

1. L'utente accede alla pagina di login
2. Inserisce username e password
3. Il sistema valida le credenziali e autentica l'utente consentendogli l'accesso alla dashboard

Inclusioni:

- UC2: Inserimento Username
- UC3: Inserimento Password

Estensioni:

- UC4: Visualizzazione Errore Credenziali

IV - 3.2. UC2 - Inserimento Username

Attore principale:

- Utente non autenticato

Precondizioni:

- L'utente ha uno username valido nel sistema
- L'utente sta effettuando l'autenticazione per la pagina di dashboard

Postcondizioni:

- Il sistema accetta lo username inserito

Scenario principale:

1. L'utente accede alla pagina di login
2. L'utente inserisce lo username

IV - 3.3. UC3 - Inserimento Password

Attore principale:

- Utente non autenticato

Precondizioni:

- L'utente ha una password valida nel sistema
- L'utente sta effettuando l'autenticazione per la pagina di dashboard

Postcondizioni:

- Il sistema accetta la password inserita

Scenario principale:

1. L'utente accede alla pagina di login
2. L'utente inserisce la password

IV - 3.4. UC4 - Visualizzazione Errore Credenziali

Attore principale:

- Utente non autenticato

Precondizioni:

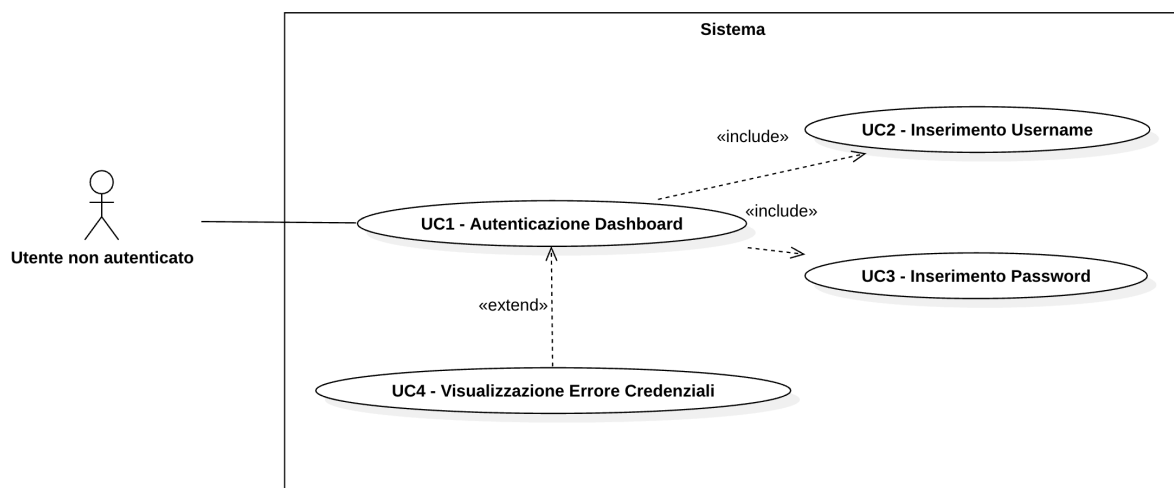
- L'utente sta effettuando l'autenticazione per la pagina di dashboard

Postcondizioni:

- Il sistema rifiuta le credenziali d'accesso inserite dall'utente poiché discrepanti con quelle memorizzate

Scenario principale:

1. L'utente accede alla pagina di login
2. L'utente inserisce lo username
3. L'utente inserisce la password
4. L'utente riceve un messaggio di errore che lo notifica delle credenziali errate



IV - 3.5. UC5 - Visualizzazione Dashboard

Attore principale:

- Utente Autenticato

Precondizioni:

- L'utente ha effettuato il login con successo

Postcondizioni:

- La dashboard principale è completamente caricata e funzionale
- Tutte le informazioni dell'utente sono visualizzate

Scenario principale:

1. L'utente accede alla dashboard
2. Il sistema carica la dashboard principale
3. Il sistema carica la mappa con i punti di interesse

4. Il sistema inizializza tutti i componenti della dashboard

Inclusioni:

- Visualizzazione profilo utente nella dashboard (UC6)
- Visualizzazione statistiche utente nella dashboard (UC7)
- Visualizzazione promozioni recenti nella dashboard (UC8)

IV - 3.6. UC6 - Visualizzazione profilo utente nella dashboard**Attore principale:**

- Utente Autenticato

Precondizioni:

- L'utente è autenticato nel sistema e quindi visualizza la dashboard
- Il sistema mantiene le informazioni del profilo utente

Postcondizioni:

- Le informazioni del profilo utente sono visualizzate nella sidebar sinistra

Scenario principale:

1. L'utente accede alla dashboard principale
2. Il sistema carica e visualizza le informazioni del profilo nella sidebar (ID, Età, Professione, Interessi)

IV - 3.7. UC7 - Visualizzazione statistiche utente nella dashboard**Attore principale:**

- Utente Autenticato

Precondizioni:

- L'utente è autenticato nel sistema e quindi visualizza la dashboard
- Il sistema traccia e memorizza le attività dell'utente

Postcondizioni:

- Le statistiche dell'utente sono visualizzate nella sezione dedicata della sidebar

Scenario principale:

1. L'utente accede alla dashboard principale
2. Il sistema calcola e mostra le statistiche aggiornate (km percorsi, negozi vicini, notifiche, minuti attivi)

IV - 3.8. UC8 - Visualizzazione promozioni recenti nella dashboard**Attore principale:**

- Utente Autenticato

Precondizioni:

- L'utente è autenticato nel sistema e quindi visualizza la dashboard
- Sono disponibili promozioni nel sistema

Postcondizioni:

- Le promozioni recenti vengono visualizzate nella sezione dedicata della sidebar

Scenario principale:

1. L'utente accede alla dashboard principale
2. Il sistema recupera le promozioni più recenti disponibili
3. Il sistema mostra le promozioni nella sezione «Promozioni Recenti» della sidebar



IV - 3.9. UC9 - Filtra punti di interesse per categoria

Attore principale:

- Utente Autenticato

Precondizioni:

- L'utente è autenticato e si trova nella dashboard principale
- Il sistema contiene punti di interesse di diverse categorie
- La barra di filtri è visibile in basso alla mappa

Postcondizioni:

- Vengono visualizzati sulla mappa solo i punti di interesse della categoria selezionata

Scenario principale:

1. L'utente accede alla dashboard principale
2. L'utente seleziona una categoria specifica (Ristoranti, Abbigliamento, Supermercati, Elettronica, Bar & Caffè)

Inclusioni:

- Visualizzazione nome del punto di interesse dal marker (UC12)
- Visualizzazione categoria dal punto di interesse dal marker (UC13)

Estensioni:

- UC10: Visualizzazione messaggio «Nessun risultato trovato»

IV - 3.10. UC10 - Messaggio di errore «Nessun risultato trovato»

Attore principale:

- Utente Autenticato

Precondizioni:

- L'utente è autenticato e si trova nella dashboard principale
- L'utente ha applicato un filtro per categoria
- Il sistema non dispone di punti di interesse inerenti alla categoria selezionata

Postcondizioni:

- Il sistema non carica alcun punto di interesse
- Il sistema mostra il messaggio di errore

Scenario principale:

1. L'utente accede alla dashboard principale
2. L'utente seleziona una categoria specifica
3. L'utente visualizza il messaggio di errore

