



UNIVERSITÀ DEGLI STUDI DI TRENTO  
FACOLTÀ DI INFORMATICA  
Corso di Ingegneria del Software

# FixMi Analisi dei Requisiti

Giovanni Santini  
Riginel Ungureanu  
Valerio Asaro

Anno Accademico 2023/2024  
Trento

## CONTENTS

0.1	Scopo del documento . . . . .	3
0.2	Informazioni del Documento . . . . .	3
<b>1</b>	<b>Obiettivo del progetto</b>	<b>4</b>
1.1	Obiettivo Principale . . . . .	4
1.2	Obiettivi Secondari . . . . .	4
<b>2</b>	<b>Requisiti</b>	<b>6</b>
2.1	Requisiti Funzionali . . . . .	6
2.1.1	Utente Non Autenticato . . . . .	6
2.1.2	Cliente . . . . .	9
2.1.3	Dipendente . . . . .	11
2.1.4	Manager . . . . .	13
2.2	Requisiti Non Funzionali . . . . .	13
<b>3</b>	<b>Design</b>	<b>18</b>
3.1	Design del Front End . . . . .	18
3.2	Design del Back End . . . . .	19
3.2.1	Divisione in Microservizi . . . . .	19
3.2.2	Sistemi Esterni . . . . .	19
3.2.3	Descrizione Microservizi . . . . .	20
3.2.4	Grafici . . . . .	21
<b>4</b>	<b>Ruoli e Responsabilità</b>	<b>22</b>
<b>5</b>	<b>Il sistema Task</b>	<b>23</b>

**6 Tecniche e Strumentazione**

**25**

## 0.1 Scopo del documento

Il seguente documento riporta la definizione e l'analisi del progetto "FixMi" in linguaggio naturale. In particolare il documento mira a:

1. Stabilire gli obiettivi del progetto.
2. Definire i requisiti funzionali e non funzionali.
3. Presentare i requisiti di Front-End del progetto.
4. Presentare i requisiti di Back-End del progetto.
5. Stabilire ruoli e funzioni dei singoli membri del team di sviluppo.
6. Definire tecniche e strumenti utili alla realizzazione del progetto.

## 0.2 Informazioni del Documento

Campo	Valore
Titolo del Documento	Analisi dei Requisiti
Titolo del Progetto	FixMi
Autori del Documento	Giovanni Santini Riginel Ungureanu Valerio Asaro
Responsabile del Progetto	Riginel Ungureanu
Versione del documento	1.2

## OBIETTIVO DEL PROGETTO

Di seguito vengono illustrati in maniera discorsiva gli obiettivi principali e secondari del progetto in questione.

### 1.1 Obiettivo Principale

L'obiettivo del progetto è la realizzazione di una Web-App denominata "Fix Mi": un'applicazione web gestionale sviluppata per un negozio di articoli informatici ed elettronici "retro". Tale software svolgerà il ruolo di supporto all'attività di commercio e riparazione dei dispositivi elettronici mettendo in comunicazione datore di lavoro, dipendenti e clienti.

### 1.2 Obiettivi Secondari

Successivamente vengono elencati gli obiettivi secondari del progetto, numerati tramite acronimo OSx (Obiettivi Secondari), dove x è il numero che identifica inequivocabilmente un obiettivo da un altro. Questi obiettivi estendono quello principale e non sono da considerare di minore importanza durante l'analisi dei requisiti.

#### **OS1 Autenticazione**

L'applicazione dispone di un sistema di distinzione dei ruoli Manager, Dipendenti e Clienti. Tale sistema permetterà diverse funzionalità in base al livello di

autorizzazione.

### **OS2 Magazzino**

Il sito presenta ai dipendenti e al manager un sistema di gestione del magazzino con il quale si potrà aggiungere e rimuovere merce, oltre a controllarne la disponibilità.

### **OS3 Negozio**

Il sito offre ai clienti uno shop, con catalogo interfacciato con il magazzino, che permette loro di effettuare acquisti. I dipendenti potranno vedere gli ordini e le richieste di vendita in modo da gestirle.

### **OS4 Riparazione**

L'applicazione permette ai clienti di richiedere la riparazione di un proprio articolo ed organizzare l'attività di riparazione di tale articolo tra i dipendenti.

### **OS5 Assistenza**

Il sito presenta ai clienti la lista dei contatti dell'azienda, nonché la possibilità di inserire una richiesta di assistenza relativa al sito, un ordine, o una riparazione. I dipendenti potranno gestire le richieste e comunicare direttamente con i clienti.

### **OS6 Feedback**

Gli utenti potranno inviare feedback sulla qualità del servizio offertogli.

### **OS7 Gestione Dipendenti**

Il manager può gestire i profili dei dipendenti. Ciò include l'aggiunta di un nuovo profilo, la rimozione o la modifica. Il sito inoltre permette di vedere le statistiche dei dipendenti.

### **OS8 Gestione Task**

L'applicazione presenta ai dipendenti e al manager un sistema di gestione delle varie tipologie di task, ossia le varie attività che i dipendenti devono svolgere, assegnando ad ogni dipendente delle task appropriate alle proprie abilità.

## REQUISITI

Posto l'obiettivo del progetto segue la definizione dei requisiti del sistema, suddivisi in Requisiti Funzionali (RF) e Requisiti Non Funzionali (RNF).

### **2.1 Requisiti Funzionali**

Segue la lista dei Requisiti Funzionali (RF) individuati per il progetto.

#### **RF0 Requisiti Generali**

Il sito deve manifestare una distinzione esplicita tra i tre tipi di utente, ossia Cliente, Dipendente e Manager, ognuno con accesso a funzionalità e privilegi diversi. In vista di questo, la seguente sezione verrà divisa in base al tipo di utente interessato.

##### **2.1.1 Utente Non Autenticato**

#### **RF1 login**

Il sito deve presentare una pagina di login:

1. La pagina richiede due campi "e-mail" e "password" e presenta un bottone "login".
2. L'utente verrà autenticato se e solo se i campi "e-mail" e "password" inseriti corrispondono entrambi ad un profilo già registrato nel sistema.

3. Nel caso il login fallisse, ossia la e-mail o la password non corrispondessero ad un utente registrato, l'applicazione mostrerà una nuova pagina di login.
4. L'utente deve essere in grado di raggiungere la pagina di registrazione specificata in RF2 in qualsiasi momento dalla pagina di login.

### **RF1.2 2FA**

La pagina di login deve implementare una tecnologia 2FA (Two Factor Authentication) secondo le seguenti modalità:

1. L'utente dovrà inserire un codice OTP (One-Time Password), generato sul momento e ottenuto attraverso e-mail, prima di completare ogni accesso all'applicazione.
2. L'autenticazione non potrà procedere finché l'utente non avrà inserito il corretto codice OTP.
3. Un codice OTP non sarà più valido dopo minuti 5 dopo la sua generazione.
4. L'utente avrà sempre la possibilità di ricevere un nuovo OTP tramite e-mail, il codice precedente non sarà più valido per effettuare l'autenticazione.

### **RF1.3 Cambio Password**

Il sito deve offrire la possibilità all'utente di cambiare password:

1. L'applicazione deve inviare un link all'utente attraverso e-mail che porterà ad una nuova pagina. La pagina presenterà un campo "nuova password".
2. La nuova password dovrà essere digitata due volte per ridurre la probabilità di errori di battitura.
3. La nuova password deve rispettare i vincoli specificati in RNF2.
4. Qualora non vi fosse alcun profilo associato alla mail inserita, il processo fallirà e verrà caricata la pagina di login.

### **RF2 Registrazione**

Il sito deve dare la possibilità ad un utente di registrarsi:

1. La pagina presenta i campi "e-mail", "password" (due volte), "username" e "nazionalità".



2. La password deve essere conforme come specificato in RNF2.

### **RF2.1 Verifica E-mail**

Per terminare la registrazione con successo, l'applicazione deve verificare la mail dell'utente

1. L'applicazione deve inviare un codice di 6 cifre, generato randomicamente sul momento, alla e-mail specificata dall'utente e mostrare un campo "inserisci codice".
2. L'applicazione deve dare la possibilità all'utente di inviare un nuovo codice di verifica del profilo. Il codice precedente non sarà più valido per completare la registrazione.

### **RF2.2 Terminazione Registrazione**

1. Qualora il codice inserito nel campo "inserisci codice" corrispondesse a quello inviato tramite e-mail, e i campi "e-mail" e "password" fossero validi, la registrazione terminerà con successo e l'applicazione caricherà automaticamente la pagina di login (RF1).
2. Se una delle condizioni del punto precedente non venisse soddisfatta, la registrazione non proseguirà e la pagina permetterà di effettuare modifiche ai dati inseriti nei campi
3. Il livello di permessi del nuovo profilo successivamente alla registrazione è di tipo Cliente.
4. La registrazione non deve terminare con successo se un utente tenta di registrarsi con una e-mail a cui è associato un profilo esistente.

### **RF3 Negozio**

Il sito presenta all'utente non autenticato una pagina "Negozio":

1. La pagina permette di visualizzare gli oggetti presenti nel magazzino al momento del caricamento della stessa
2. L'utente deve essere in grado di ordinare gli oggetti per nome o per prezzo tramite un bottone, ricercare una stringa tra tutti gli oggetti nel magazzino tramite una barra di ricerca.

3. Ogni oggetto deve avere una pagina dedicata accessibile tramite un click sull'oggetto

## **RF4 Informazioni / Contatti**

L'applicazione deve offrire all'utente una pagina dove sono visibili dati utili riguardo l'azienda:

1. Descrizione dell'azienda.
2. Una lista contenente varie Frequently Asked Questions (FAQ) riguardo il funzionamento del sito.
3. Contatti dell'impresa: numeri di telefono, e-mail e indirizzo.
4. Il sito deve mostrare inoltre un riquadro con una mappa dove è possibile visualizzare la posizione del negozio.

### **2.1.2 Cliente**

Il sito deve dare la possibilità al cliente di autenticarsi attraverso la pagina di login come specificato in RF1 inserendo le proprie credenziali (e-mail e password) inserite in fase di registrazione, in modo da poter accedere alle funzionalità elencate di seguito:

## **RF5 Negozio**

Il sito deve permettere al cliente, oltre alle funzionalità definite in RF3 per gli utenti non autenticati, le seguenti funzionalità:

### **RF5.1 Carrello**

Il sito deve avere una pagina dedicata al carrello del cliente. Il sito deve permettere al cliente autenticato, nella pagina dedicata a un specifico articolo, di inserirlo nel proprio carrello attraverso un pulsante.

1. La pagina deve avere un elenco di tutti i prodotti che sono stati immessi nel carrello. Ogni prodotto nell'elenco deve avere il nome, il prezzo, una foto e un pulsante "rimuovi"
2. La pagina deve permettere al cliente di rimuovere un elemento dal proprio carrello premendo il pulsante "rimuovi" dell'elemento.

3. Ogni prodotto deve portare alla pagina dedicata allo stesso cliccando sul nome nell'elenco.
4. La pagina deve mostrare il prezzo totale, ossia la somma dei prezzi degli elementi del carrello.
5. La pagina deve avere un tasto "procedi al pagamento" con il quale si procede alla pagina di pagamento specificata sotto.
6. L'utente deve avere la possibilità di vedere i prodotti salvati nel proprio carrello anche dopo aver chiuso la pagina.

### **RF5.2 Pagamento**

La pagina deve fornire diversi metodi di pagamento:

1. La pagina deve fornire il pagamento tramite Paypal e Bancomat con circuiti Visa e Mastercard.
2. Una volta effettuato il pagamento con successo, il sito deve fornire un riepilogo dell'acquisto e un messaggio che invita il cliente a recarsi in negozio per il ritiro dei prodotti.
3. Il sito deve creare per i dipendenti una task "magazzino" (RF9), dettagliando i prodotti richiesti.
4. Il sito deve inviare i dettagli del pagamento al software di accounting scelto.
5. Il sito deve svuotare gli elementi del carrello dell'utente una volta effettuato il pagamento.

### **RF6 Riparazione**

Il sito deve offrire al cliente la possibilità di inviare, attraverso un'apposito form, una richiesta di riparazione per un proprio articolo di elettronica.

1. Il form ha i campi "nome", "cognome", "e-mail", "numero di telefono", "descrizione del problema", "foto" e un bottone "Invia".
2. Successivamente all'invio della richiesta da parte del cliente, l'applicazione deve generare una task di tipo "riparazione". Il sistema di task è specificato in RF9

3. Qualora il cliente abbia inviato una riparazione, il sito deve mostrare al cliente lo stato della propria riparazione in un'apposita pagina "stato riparazioni", interfacciandosi con il sistema di task specificato in RF9. In caso non vi siano riparazioni, la pagina deve mostrare la scritta "Nessuna riparazione registrata".

### **RF7 Assistenza**

Il sito deve permettere al cliente di richiedere assistenza e comunicare con l'azienda attraverso un form:

1. Il form ha i campi "e-mail", "descrizione richiesta" e un bottone "Invia".
2. Successivamente all'invio della richiesta di assistenza da parte del cliente, l'applicazione deve generare una task di tipo "assistenza". Il sistema di task è specificato in RF9.

### **RF8 Feedback**

Il sito deve dare la possibilità al cliente di inviare anonimamente del feedback attraverso una pagina dedicata.

1. La pagina contiene i campi "disponibilità azienda", "velocità riparazione", "soddisfazione riparazione", "soddisfazione sito web", "idee per migliorare" e un bottone "Invia".
2. Il sito deve creare una task assistenza con i contenuti dei campi inseriti dall'utente.

#### **2.1.3 Dipendente**

Il sito deve dare la possibilità al dipendente di autenticarsi attraverso la pagina di login (RF1) inserendo le proprie credenziali (e-mail e password) fornitegli dal manager, in modo da poter accedere alle funzionalità dei clienti e quelle riservate ai dipendenti elencate qui sotto.

### **RF9 Pagina Task**

Il sito deve offrire una pagina "Task" ai dipendenti autenticati, il sistema delle *task* è definito nel capitolo 5 a pagina 23, si consiglia di leggere tale capitolo prima di continuare.

1. La pagina permetterà al dipendente di visualizzare una lista di task e i corrispettivi *task-tag*.
2. Le task visualizzate sono automaticamente selezionate per il dipendente autenticato dall'applicazione in base ai *work-tag* associati al proprio profilo.
3. Le task visualizzate sono solo quelle con lo stato "Da eseguire" oppure "In pausa".

#### RF9.1 Interagire con le tasks

Nella pagina "Task", il dipendente deve essere in grado di:

1. Selezionare una task dalla lista da eseguire o da quelle messe in pausa, contrassegnandola come in lavorazione, solo nel caso in cui il dipendente non presenti alcuna task in lavorazione.
2. Mettere in pausa la task in esecuzione, la task verrà contrassegnata come "In pausa" e il dipendente potrà selezionare una nuova task dalla lista.
3. Contrassegnare una task come "completata" e salvarla in una lista di task completate, insieme al nome del / dei dipendente/i responsabili del completamente e alla data, oppure come "fallita" nel caso ci fossero stati dei problemi. Il dipendente allora potrà scegliere una nuova task dalla lista.
4. Creare una nuova task di qualsiasi tipologia, impostando *task-tag* e la descrizione (anche vuota).

#### RF10 Magazzino

Il dipendente deve aver la possibilità di interagire con il magazzino in un'apposita pagina, in particolare:

1. Visualizzare i prodotti all'interno del magazzino ordinati per nome o per data di arrivo, in modo crescente o decrescente.
2. Aggiungere articoli.
3. Rimuovere articoli.

### 2.1.4 Manager

Il sito deve dare la possibilità di autenticazione del manager nella pagina di login, fornendo e-mail e password del profilo manager.

## RF11

Se un profilo è di tipo manager, deve poter accedere a tutte le funzionalità del profilo “dipendente” specificate sopra.

### RF11.1 Gestione dei Dipendenti

Il manager deve essere in grado di:

1. Aggiungere profili “dipendente” dall’applicazione inserendo l’e-mail del profilo in un apposito campo
2. Aggiungere dati Anagrafici associati all’account dipendente , ”nome”, ”cognome”, ”data di nascita”, ”giorno assunzione”.
3. Rimuovere dipendenti tramite un bottone che permette di selezionare la e-mail del dipendente e rimuoverla dalla lista dipendenti.
4. Assegnare e rimuovere una o più *work-tag* ai dipendenti.
5. Visualizzare informazioni dei dipendenti in apposita pagina, in particolare:
  - Informazioni Biografiche.
  - Storico con attività e record delle task svolte / in svolgimento. Lo storico deve poter essere filtrato e ordinato tramite data di inizio e fine task, nome dipendente, lista di task-tag.
6. Il profilo manager ha tutte le *work-tag* assegnate per il proprio profilo.

## 2.2 Requisiti Non Funzionali

Di seguito vengono elencati i requisiti non funzionali dell’applicazione, numerati con acronimo RNF.

Obiettivi	Requisiti
OS1 Autenticazione	RF1, RF2
OS2 Magazzino	RF9, RF10
OS3 Negozio	RF3, RF9, RF10
OS4 Riparazione	RF6, RF9
OS5 Assistenza	RF7, RF9
OS6 Feedback	RF8, RF9
OS7 Gestione dei Dipendenti	RF11
OS8 Gestione Task	RF9

Table 2.1: Relazione tra Obiettivi e Requisiti Funzionali

### RNF1 Intuitività e Accessibilità

Il sito deve apparire semplice, sia sotto l'aspetto visivo che sotto l'aspetto pratico:

1. In media l'utente deve essere in grado di capire le funzionalità con una sola lettura della descrizione.
2. Il sito deve essere disponibile sia in lingua italiana che in lingua inglese, l'utente con un livello di lingua A1 è in grado di leggere e comprendere il contenuto.
3. Il sito deve avere un design consistente nella sua interezza, utilizzando un singolo font e una palette fissa di colori.

### RNF2 Sicurezza

Il sito deve gestire ogni comunicazione in maniera sicura, sia tra utenti e server, che tra i servizi esterni di cui usufruisce:

1. Il sito deve garantire che i dati sensibili siano crittografati e inaccessibili a soggetti indesiderati.
  - Il sito deve utilizzare un algoritmo di hashing di tipo SHA-3 per il controllo e la memorizzazione delle passwords.
  - Il sito deve utilizzare i protocolli tls e https per ogni comunicazione tra utenti e servizi.

2. Il sito deve verificare l'identità dell'utente attraverso il 2FA come specificato nel RF1.
3. Il sito deve, in fase di registrazione e di modifica password, controllare la sicurezza della password, ossia assicurarsi che la password abbia almeno una lunghezza pari a 10 caratteri e presenti almeno un numero, una lettera maiuscola e un carattere speciale dalla lista

! ? \$ % ^ & \* ( ) \_ - + = { [ ] } : ; @ # | \ < , > .

### RNF3 Privacy

L'applicativo deve essere conforme alle principali direttive sull'utilizzo e la gestione dei dati personali del cliente, del dipendente e del manager. In particolare il sito dovrà rispettare le normative del GDPR:

Il GDPR (Regolamento Generale sulla Protezione dei Dati) è una legge europea che riguarda la protezione dei dati personali. Le principali aree di focus del GDPR includono il consenso esplicito per la raccolta dei dati, la trasparenza nell'uso dei dati, la possibilità di accesso e cancellazione dei dati personali da parte dell'individuo, e misure di sicurezza per proteggere tali dati.

### RNF4 Affidabilità e Disponibilità

1. la probabilità che il sito fornisca i risultati desiderati senza interruzioni o tempi di inattività deve essere maggiore del 99%
2. la probabilità che il sito rimanga operativo in un determinato momento indipendentemente dal numero di guasti già subiti dal sistema deve essere maggiore del 99%

### RNF5 Performante

Il sito deve offrire velocità di accesso ai servizi offerti in tempi rapidi:

1. Il sito deve aggiornare la lista degli articoli presenti in negozio, in caso di modifica al magazzino, in meno di un secondo.
2. Il sito deve inviare le mail di 2FA(specificata nell'RF1) in meno di 5 secondi.
3. Il sito deve aggiornare la lista delle task (in caso di completamento, aggiunta, messa in pausa) in meno di un secondo.



## RNF6 Compatibilità e Portabilità

L'applicazione deve essere supportata su una molteplice quantità di dispositivi.

Tra i principali vi sono:

1. Dal lato client, computer e dispositivi mobili aventi un browser che supporta:
  - html5
  - https
  - tls 1.2
2. Dal lato server, computer che supporti:
  - Node js 18.18.0 LTS
  - MongoDB 7.0

Il sito deve essere "responsive", ossia deve potersi adattare alla dimensione degli schermi dei principali dispositivi in commercio, quali:

1. TV e monitor di PC e Laptop
  - Aspect Ratio da 4:3, 16:9, 21:9
2. Smartphone
  - Iphone X,XR,11,..., 14
  - Tutti i modelli Xiaomi dal 2018 in poi
  - Tutti i modelli Samsung dal 2018 in poi
  - Tutti i modelli Motorola dal 2018 in poi
  - Tutti i modelli Huawei dal 2018 in poi
3. Tablet
  - Ipad Air, Pro dal 2018 in poi
  - Tutti i modelli Xiaomi dal 2018 in poi
  - Tutti i modelli Samsung dal 2018 in poi

**RNF7 Mantenibilità e Scalabilità**

1. Al sito deve essere affiancato, prima e dopo il rilascio ufficiale, un team di manutenzione che si occupi di testare ogni funzionalità periodicamente e che, su richiesta qualora ci siano problemi, sia pronto a intervenire tempestivamente
2. Il sito, al fine di essere facilmente mantenibile, deve possedere un codice sorgente con le seguenti caratteristiche:
  - Il codice sorgente del back-end deve essere modulare, utilizzando un'architettura a microservizi come descritto nel capitolo 3.2 - Design del back-end.
  - Il codice sorgente deve rispettare le linee guida ufficiali del linguaggio scelto.

**RNF8 Conformità**

L'applicazione deve essere conforme alle normative di legge in materia di siti web imposti dall'Unione europea, inoltre vengono rispettati i seguenti:

1. GDPR
2. W3C WAI

### 3.1 Design del Front End

Nella seguente sezione verranno mostrate alcune grafiche al fine di visualizzare come il sito mostrerà le proprie funzionalità all'utente e ai dipendenti.

#### Schermata di Login

In seguito viene mostrata la schermata di Login di Fix Mi, esso rispecchia le caratteristiche descritte dal (RF1) e viene mostrato attraverso le tre tipologie di schermo discusse nel (RNF6).

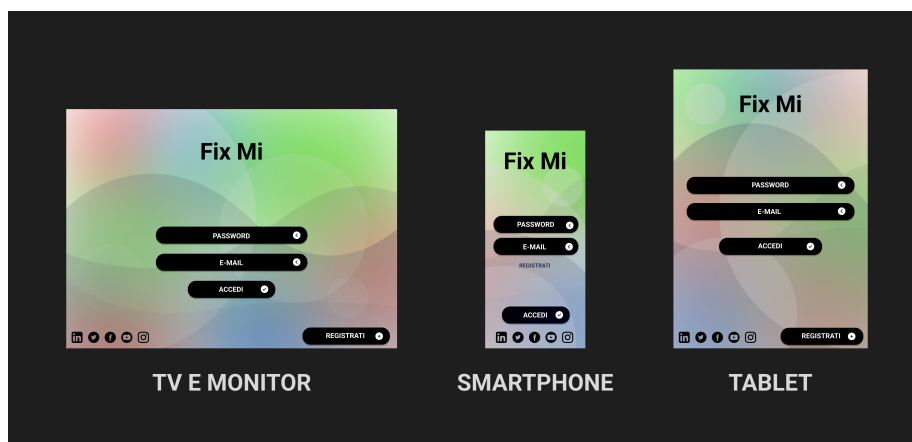


Figure 3.1: La schermata di Login di Fix Mi

## 3.2 Design del Back End

Nella seguente sezione vengono riportate le specifiche generali per quanto riguarda la back-end dell'applicazione.

### 3.2.1 Divisione in Microservizi

La backend presenta un'architettura basata su microservizi indipendenti che comunicano tramite API. In particolare i microservizi sono:

- Home
- Autenticazione
- Negozio
- Magazzino
- Task
- Mail Server
- Gestione Dipendenti

### 3.2.2 Sistemi Esterni

L'applicazione si interfaccerà con i seguenti sistemi esterni:

#### Sistemi di Pagamento

La backend si interfaccierà con Paypal Name-Value-Pair (NVP) e NEXI XPay per effettuare i pagamenti nella pagina "Negozio" (RF3).

#### Databases

La back-end utilizzerà il servizio "MongoDB Cluster Atlas" per interagire con databases distribuiti, in particolare i database per gli utenti registrati, le task e il magazzino.

#### OpenStreetMap

Nella pagina di informazioni, l'applicazione userà OpenStreetMap per visualizzare la posizione dell'azienda nella mappa.

**2FA**

Il sito utilizzerà un servizio esterno per effettuare l'autenticazione a 2 fattori.

**3.2.3 Descrizione Microservizi****Home**

Home presenta la pagina principale "Home" e permette di reindirizzare l'utente al microservizio desiderato tra Autenticazione, Negozio, Magazzino, Task, Gestione Dipendenti, in base al livello di autorizzazione, e alle pagine Feedback, Assistenza, Informazioni.

**Autenticazione**

Il microservizio gestisce la registrazione e l'autenticazione degli utenti. Si interfaccia ad un database non relazionale esterno per memorizzare i profili e le rispettive informazioni.

**Server Mail**

L'applicazione prevede di implementare un server SMTP per l'invio delle mail a tutti i tipi di utente e prevede una mail aziendale per i dipendenti e per i manager.

**Gestione delle Task**

Il microservizio si interfaccia con database esterno non relazionale per memorizzare le task, mostrarle ai dipendenti e salvarne lo stato (definito nel capitolo 6).

**Gestione dei Dipendenti**

Il microservizio è in grado di leggere e scrivere nel database delle task e degli utenti registrati.

**Magazzino**

Il microservizio utilizza un database non relazionale per memorizzare gli articoli contenuti all'interno del magazzino, oltre che alle operazioni svolte, come:

- l'aggiunta di un nuovo articolo

- la rimozione di un articolo esistente

### 3.2.4 Grafici

In seguito uno schema di alto livello dei collegamenti tra i diversi microservizi. In blu i microservizi, in giallo i sistemi esterni, in verde i databases.

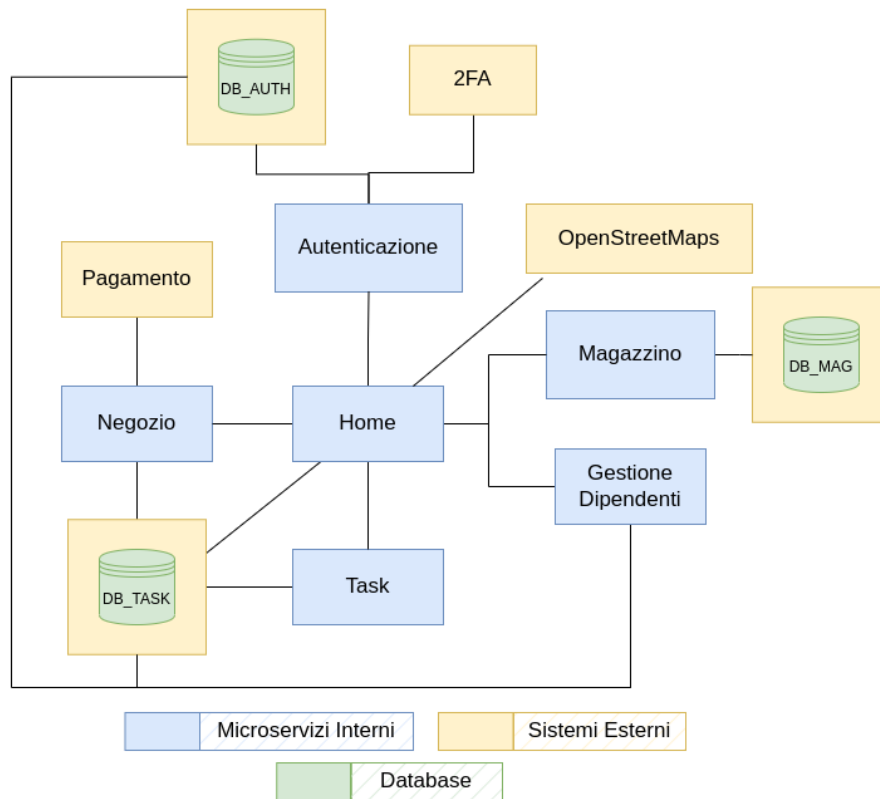


Figure 3.2: Schema di alto livello della back end

## RUOLI E RESPONSABILITÀ

Il team di sviluppo ha suddiviso l'obiettivo del progetto in obiettivi secondari di cui sono stati successivamente individuati ruoli e responsabilità:

Ruolo	Membro
Sviluppo dell'obiettivo	Il gruppo di sviluppo
Leader del team di sviluppo	Riginel Ungureanu
Supervisore Back-End	Giovanni Santini
Supervisore Front-End	Valerio Asaro

Table 4.1: Relazione tra Ruoli e Membri

## IL SISTEMA TASK

Il sistema delle task è centrale alla gestione dell'attività aziendale e al funzionamento dell'applicazione. In seguito vi è una descrizione del suo funzionamento

### **Definizione Task e Task-tag**

Una task è un'attività sulla quale un dipendente o un manager può decidere di lavorare. Esistono diverse tipologie di task, caratterizzate da una *task-tag* e una descrizione di massimo 200 caratteri.

Una *task-tag* è una e una sola stringa tra le seguenti:

- Magazzino
- Negozio
- Assistenza e Supporto
- Riparazione

Le tasks possono essere in uno dei seguenti stati:

- Da eseguire
- In lavorazione
- In pausa
- Completata
- Fallita



### Significato delle singole tasks

In seguito vi è una descrizione del significato di ogni tipologia di task:

1. Magazzino: il dipendente accede alla pagina del magazzino (RF10) per apportare eventuali modifiche o visualizzare i vari articoli.
2. Negozio: il dipendente si assume il compito di rimanere alla cassa per interagire con eventuali clienti.
3. Assistenza: il dipendente visualizza la richiesta di supporto dal cliente tramite mail. Il dipendente potrà rispondere per mail al cliente in base alle esigenze del caso, secondo la propria discrezione.
4. Riparazione: il dipendente potrà prendersi carico di una riparazione selezionando e avviando la task. Qualora ci fossero problemi e/o ritardi, il dipendente potrà mettere la task in pausa e riprenderla in qualsiasi momento. Una volta che la riparazione viene terminata, il dipendente segnerà la task come completata o fallita nel caso la riparazione non avesse successo.

### Definizione work-tag

Ad ogni profilo dipendente sono associate una o più *work-tag*. Una *work-tag* è una stringa dello stesso tipo di una *task-tag* (Magazzino, Negozio, Assistenza, Riparazione).

Ogni dipendente potrà lavorare solo sulle task che hanno gli stessi o meno *task-tag* rispetto ai *work-tag* del dipendente.

## TECNICHE E STRUMENTAZIONE

Diversi strumenti e tecniche sono state scelte accuratamente in base alla situazione posta dall'obiettivo. Di seguito un elenco degli strumenti utilizzati per la realizzazione del progetto:

- Trello      <https://trello.com/>
- GitHub     <https://github.com/>
- Google Suite    <https://workspace.google.it/intl/it/>
- LaTeX       <https://www.latex-project.org/>
- VSCode      <https://code.visualstudio.com/>
- Figma       <https://www.figma.com/>
- Coolors      <https://coolors.co/>
- Diagrams     <https://app.diagrams.net/>