**Progetto easyNet**

Sommario

[**Obiettivo del progetto** 1](#_Toc135030489)

[**Descrizione Generale** 1](#_Toc135030490)

[**Modello E/R database relazionale** 2](#_Toc135030491)

[**Dizionario** 2](#_Toc135030492)

[**Modello logico** 3](#_Toc135030493)

[**Impostazione modello non relazionale** 4](#_Toc135030494)

[Bot 4](#_Toc135030495)

[**Descrizione tipologie d’utente** 5](#_Toc135030496)

[**Tecnologie utilizzate per il progetto:** 7](#_Toc135030497)

[API: 8](#_Toc135030498)

[Frontend: 8](#_Toc135030499)

[**Sottosistema Bot** 8](#_Toc135030500)

[DESCRIZIONE 8](#_Toc135030501)

[**Sottosistema contenuti** 9](#_Toc135030502)

[DESCRIZIONE 9](#_Toc135030503)

[CASI D'USO: 9](#_Toc135030504)

[**Interfaccia utente** 11](#_Toc135030505)

# **Obiettivo del progetto**

L'obiettivo del progetto è creare un social aziendale dove le aziende e gli utenti possano condividere post contenenti immagini, video o testo. Le aziende potranno mettere a disposizione dei bot a cui l’utente potrà porgere delle domande, a cui corrisponderanno delle risposte predefinite.  
Il focus principale di questo social è migliorare l’immagine dell’azienda a persone esterne ed interne ad essa. L'obiettivo è dare alle aziende e utenti una piattaforma dove mostrarsi a futuri dipendenti/datori di lavoro.

# **Descrizione Generale**

L’idea è di realizzare un’applicazione web composta da più pagine, che richiede la registrazione dell’utente per poter usufruire di buona parte delle funzionalità sviluppate.

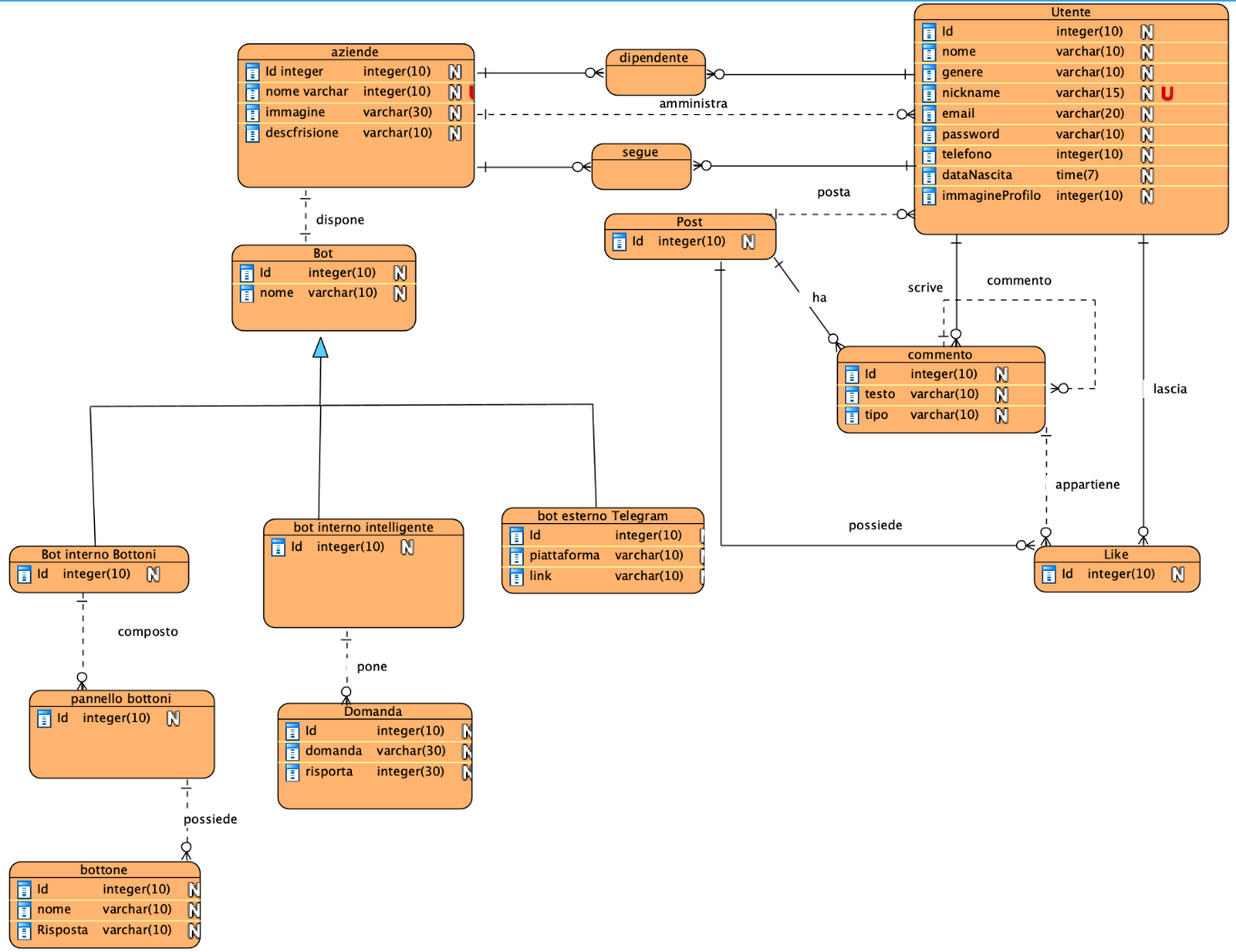
L’applicazione sarà focalizzata sulle interazioni tra dipendenti e aziende.

Nello specifico, i dipendenti avranno il compito di creare contenuti per conto delle aziende.

L’utente non autenticato visualizzerà un’anteprima che mostra gli ultimi tre post delle aziende, la pagina di registrazione e quella di accesso, quest’ultima comprende anche la possibilità di effettuare il recupero della password.

L’utente che effettua l’accesso può visualizzare la versione completa dell’applicazione web, che comprende la pagina dei post delle aziende e la pagina dei bot. Gli utenti associati ad un’azienda possono liberamente interagire anche su contenuti appartenenti ad un’altra azienda. I bot servono come supporto per le necessità dei dipendenti.

# **Modello E/R database relazionale**



# **Dizionario**

|  |  |  |  |
| --- | --- | --- | --- |
| **Entità** | **Descrizione** | **Attributi** | **Identificatori** |
| Utente | Utente autenticato | Id, nome, genere, nickname, email, password, telefono, dataNascita, immagineProfilo | Id |
| Azienda | Azienda creata da un utente | Id, nome, immagine, descrizione | Id |
| Post | Post creato da un utente | Id, url, descrizione | Id |
| Like | Like lasciato da un utente ad un post | Id | Id |
| Commento | Commento di un utente ad un post o ad un altro commento | Id, testo, tipo | Id |
| BotInternoIntelligente | Bot intelligente interno al sistema | Id, nome, chiamata, credenziali | Id |
| BotBottoni | Bot interno al sistema con bottoni predefiniti | Id | Id |
| Pannello | Entità che contiene i bottoni del bot a bottoni | Id | Id |
| Bottone | Chiamate predefinite | Id, nome, risposta | Id |
| Domanda | Domanda che viene posta al bot | Id | Id |

|  |  |  |
| --- | --- | --- |
| **Relazioni** | **Descrizione** | **Entità coinvolte** |
| Scrive | Utente scrive un commento | Utente(1, N),Commento(1,1) |
| Posta | Utente posta un post | Utente(1, N), Post(1,1) |
| Ha | Un post ha commenti | Post(1,N), Commento(1,1) |
| Commento | Commento ad un commento | Commento(1,N), Commento(1,1) |
| Appartiene | Un like appartiene ad un commento | Like(1,N), Commento(1,1) |
| Lascia | Utente lascia un like | Utente(1,N), Like(1,1) |
| Possiede | Un post possiede dei like | Post(1,N), Like(1,1) |
| Amministra | Un utente amministratore amministra l'azienda | Utente(1,1), Azienda(1,N) |
| Segue | Un utente segue un azienda | Utente(1,N), Azienda(1,N) |
| Dipendente | Un dipendente fa parte di un azienda | Utente(1,N), Azienda(1,N) |
| Dispone | Un azienda dispone di un solo bot | Bot(1,1), Azienda(1,1) |
| Composto | Un bot utilizza un pannello | Botbottoni(1,N), Pannello(1,1) |
| Pone | Bot interno pone la domanda ad un AI | BotInternoIntelligente(1,N), Doamanda(1,1) |
| Possiede | Un pannello possiede più bottoni | Pannello(1,N), Bottone(1,1) |

# **Modello logico**

Utente (Id, nome, genere, nickname, email, password, telefono, dataNascita, immagineProfilo)

Azienda (Id, nome, immagine, descrizione, FK\_amministratore\*, FK\_bot\*)

Segue (FK\_utente\*, FK\_azienda\*)

Dipendente (FK\_utente\*, FK\_azienda\*)

Post (Id, url, descrizione, FK\_utente\*)

Like (Id, FK\_utente\*, FK\_Post\*, FK\_Commento\*)

Commento (Id, testo, tipo, FK\_utente\*, FK\_post\*, FK\_commento\*)

BotInternoBottoni (Id)

Pannello (Id, FK\_BotInternoBottoni\*)

Bottone (Id, nome, risposta, FK\_pannello\*)

BotInternoIntelligente (Id)

Domanda (Id, domanda, risposta, FK\_ BotInternoIntelligente\*)

BotEsterno (Id, piattaforma, link)

# **Impostazione modello non relazionale**

## Bot

I bot hanno dei campi condivisi, ogni bot ha poi dei suoi campi specifici che si aggiungono a quelli esistenti.

Campi condivisi:

{

    "id":"",

    "nomeAzienda":""

}

Campi solo BotBottoni:

{

    "Q&A":

[

        {

            "domanda":"",

            "risposta":""

        }

    ]

}

Campi solo BotEsterno:

{

    "piattaforma":"link alla piattaforma"

}

Campi BotUtente:

{

    "utenti":{

        "id":"",

        "post":[

            "id\_post"

        ],

        "lista\_seguiti":[

            "id\_utente"

        ],

        "lista\_follower":[

            "id\_utente"

        ],

        "post\_piaciuti":[

            "id\_post"

        ],

        "post\_salvati":[

            "id\_post"

        ],

        "post\_menzionati":[

            "id\_post"

        ]

    }

}

Campi Post:

{

"post":{

"id":"",

"id\_utente":"",

"nome\_utente":"",

"like":[

"id\_utente"

],

"commenti":[

{

"id\_utente":"",

"nome\_utente":"",

"commenti":"",

"like":[

"id\_utente"

],

"risposte":[

{

"id\_utente":"",

"nome\_utente":"",

"commento":"",

"like":[

"id\_utente"

]

}

]

}

],

"contenuto":"pagina MD",

"hastag":[

"parola"

],

"menzioni":[

"id\_utente"

]

}

}

# **Descrizione tipologie d’utente**

Per registrarsi, l'utente inserisce i dati richiesti e successivamente completa la procedura di verifica della mail.

A seguito del login potrà cambiare la password e recuperarla in caso di perdita. Questo avviene, come per la registrazione, attraverso una mail di conferma.

L'utente può richiedere i dati relativi alla sua attività salvati dall’applicazione e, se lo desidera, inviare una richiesta di cancellazione.

Ogni utente registrato può chiedere di creare un’azienda e diventarne amministratore. Per fare questo, sarà necessario inviare dei documenti che certificano l’esistenza dell’azienda e la sua partecipazione ad essa.

I documenti verranno poi verificati da un moderatore che confermerà oppure no la richiesta.

L'amministratore dell'azienda può promuovere altri utenti a dipendenti di quella stessa azienda, decidendo cosa possono o non possono fare per suo conto.

I moderatori possono decidere di espellere alcuni utenti dalla piattaforma in base a delle policy comportamentali da definire.

Registrazione:

* L'utente si registra al sistema inserendo mail, password e le sue generalità;
* L'account non viene attivato fino a quando la mail non viene verificata, seguendo la procedura descritta nel caso d'uso "Verifica e-mail".

Verifica e-mail:

* Il sistema invia una mail di conferma all'utente per verificare che la mail sia effettivamente sua.

Invio mail di recupero:

* L'utente avvia la procedura di recupero password, cliccando sull’apposito pulsante;
* Il sistema invia una mail all’indirizzo con cui l’utente si è registrato contenente il link alla pagina di recupero password.

Recupero della password:

* L'utente inserisce una nuova password;
* Il sistema modifica la password nel database.

Consultazione del profilo:

* Il sistema mostra le informazioni sul profilo all'utente.

Cancellazione utente:

* L'utente chiede di essere cancellato;
* Il sistema elimina i dati dell'utente dal database.

Richiesta dati utente:

* Il sistema mostra tutti i dati salvati che riguardano l'utente.

Login:

* L'utente inserisce e-mail e password;
* Il sistema verifica che mail e password siano corretti;
* Il sistema autentica l'utente.

Creazione azienda:

* L'utente chiede di creare un’azienda e diventarne amministratore;
* L'utente carica i documenti che provano l’esistenza dell’azienda e la sua partecipazione ad essa;
* L'azienda non viene attivata fino a quando i documenti non vengono verificati secondo la procedura descritta nel caso d’uso “Verifica documenti”.

Verifica documenti:

* Il sistema mostra un elenco di richieste di verifica;
* Il moderatore sceglie una richiesta da verificare;
* Il moderatore verifica che i documenti caricati dall'utente siano validi;
* Il sistema registra l'azienda e rende l'utente amministratore.

Espulsione utente (ban):

* Il moderatore indica un utente da eliminare;
* Il sistema invia una e-mail di notifica all'utente;
* Il sistema elimina l'utente dal database.

Autorizzazione dipendenti:

* Il sistema mostra un elenco di utenti dotato di una funzione di ricerca;
* L'amministratore sceglie un utente che diventerà dipendente;
* L'amministratore sceglie il "livello" di autorizzazione del dipendente;
* Il sistema aggiunge l'utente ai dipendenti dell'azienda.

# **Tecnologie utilizzate per il progetto:**

Elenco:

* Databases
* Vue.js
* Asp NetCore
* JWT
* Bot Servizio
* Conversational API

BOT Service - Telegram API:

* Hosting;
* Interfaccia utente: useremo JavaScript e Vue.js. Ci saranno due fasi: la prima di verifica dell’identità del client con il JWT Token generato dalla Web API, poi l’invio della richiesta alla API di Telegram.

Conversational BOT API - Tecnologia Utilizzata:

* Google Dialogflow / Azure Language Understanding
* Hosting: Servizio cloud based.
* Benefici:
  + Riconoscimento degli intent mediante l'utilizzo del natural language understanding model di Google (Google Assistant) o Microsoft;
  + API: permettono l'interazione da qualsiasi applicazione, web o mobile;
  + Possibilità di riutilizzare gli intent.

Database:

Prevediamo di usare due database: uno SQL e uno NoSQL.

* Database SQL:
  + Distribuzione: MariaDB
  + Hosting: Azure
  + Opzione 1: Usato solo per la sola gestione degli account e l'autenticazione degli utenti
  + Opzione 2: Usato per il salvataggio di tutti i dati generati dagli utenti. (In questo caso l'applicazione sarebbe meno performante, in quanto sarebbero necessarie delle join per ottenere dati semplici come numero di seguiti e follower).Il DB NoSQL non sarebbe quindi necessario.
* Database NoSQL:
  + Distribuzione: MongoDB
  + Hosting: MongoDB Atlas
  + Usato per la gestione dei contenuti creati dagli utenti. Miglioreremmo le prestazioni generali e il carico di lavoro per i server

## API:

Le API per l'accesso ai dati saranno gestite usando un'applicazione ASP.NET Web API.

Questa si occuperà di autenticare gli utenti, di interagire con i database e di fornire i JWT.

L'interazione con i database avverrà utilizzando EntityFramework.

## Frontend:

Il frontend del sito verrà realizzato usando Vue.js, HTML e CSS.

Il sito si interfaccerà con le API dell'applicazione ASP.NET per ottenere i dati richiesti.

Verranno utilizzati le librerie di Bootstrap e JQuery.

# **Sottosistema Bot**

## DESCRIZIONE

Il sistema prevede tre tipi di bot che si differenziano in base all'intelligenza e alle tecnologie utilizzate e che saranno integrati nell’applicazione web.  
I tre tipi sono:  
- Bot a bottoni;  
- Bot intelligenti;  
- Bot Telegram.

I bot a bottoni mostreranno all'utente un elenco di bottoni con le varie azioni disponibili. L’utente avrà la possibilità di selezionarne uno e il bot risponderà con del testo o con un altro pannello di bottoni.  
I bot intelligenti utilizzeranno delle tecnologie di riconoscimento degli intent (Azure LUIS o Google DialogFlow) e permetteranno un'interazione tipo chat.  
I bot di Telegram saranno disponibili all'interno dell'applicazione attraverso una web view.

L'amministratore dell'azienda potrà scegliere quale tipo di bot creare.  
Nel caso dei bot a bottoni l'utente potrà definire, usando un linguaggio di markdown, il formato del pannello da mostrare all'utente. Definirà inoltre le risposte ai vari bottoni.  
I bot di Telegram saranno configurabili dall'applicazione web e apparterranno al sistema, non ai singoli amministratori delle aziende.  
Infine, per quanto riguarda i bot intelligenti, l'amministratore fornirà un esempio di domanda e la risposta. Useremo un servizio AI per generare delle domande simili a quella data dall'amministratore. Queste verranno usate per allenare il modello di riconoscimento del linguaggio attraverso le API.

I bot saranno disponibili solo agli utenti registrati e saranno tutti disponibili all'interno dell'applicazione web.  
Ogni azienda potrà avere al massimo un bot.

CASI D’USO:

* Interazione Bot
  + Attori: Utente registrato;
  + Descrizione (bot intelligente):
    - L'utente registrato scrive un messaggio nella chat;
    - Il bot interpreta la richiesta e risponde.
  + Descrizione (bot a bottone):
    - Il sistema mostra i bottoni disponibili;
    - L'utente clicca sul bottone desiderato;
    - Il bot risponde di conseguenza in base al bottone cliccato.
  + Descrizione (bot di Telegram):
    - Il sistema mostra la chat con il bot di Telegram attraverso una WebView;
    - L'utente scrive il comando;
    - Il sistema si interfaccia con il bot e fornisce la risposta

* Creazione Bot
  + Attori: Admin azienda
  + Descrizione: L'admin accede al menù per la creazione del bot;
    - Bot intelligenti:
      * L'admin inserisce una domanda esempio e la risposta;
      * Il sistema genera delle domande simili (usando il modello NLP) in modo da allenare il modello.
    - Bot a bottoni:
      * L'admin definisce la struttura del pannello dei bottoni usando il linguaggio md;
      * L'admin definisce la risposta ad ogni bottone.
    - Bot di Telegram:
      * L'admin definisce i comandi disponibili;
      * L'admin definisce la risposta ad ogni comando;
      * Il sistema configura il bot di Telegram.

# **Sottosistema contenuti**

## DESCRIZIONE

Gli amministratori e i dipendenti possono fare post per conto delle aziende.

Gli utenti registrati possono visualizzare i post e mettere like e commenti, possono ricevere likes sia i post sia i commenti.

All'utente viene mostrata la home page con gli ultimi post di aziende consigliate per lui, da cui, attraverso un filtro, può decidere se visualizzare solo post di pagine seguite.

Viene mostrata anche una pagina o casella di ricerca dove l'utente può cercare le pagine di aziende specifiche.

Cliccando sul bottone dei commenti vicino a un post viene visualizzata la lista dei commenti di quel post.

L'utente non registrato può visualizzare solo gli ultimi 4 post del profilo.

CASI D'USO:

* Anteprima:
  + L'utente sceglie un profilo;
  + Il sistema mostra gli ultimi 4 post di quel profilo e le sue informazioni generali.
* Visualizzazione profilo:
  + L'utente sceglie un profilo;
  + Il sistema mostra tutti i post di quel profilo e le informazioni generali del profilo.
* Filtro:
  + Il sistema mostra una lista di post;
  + L'utente sceglie un filtro da applicare;
  + Il sistema mostra una lista di post aggiornata.
* Ricerca:
  + L'utente applica una ricerca per nome, tag o per parole chiave;
  + Il sistema mostra una lista di aziende che potrebbero soddisfare la ricerca dell'utente;
  + L'utente sceglie un’azienda;
  + Il sistema mostra il profilo dell'azienda.
* Gestione pagina:
  + L'admin azienda o il dipendente modificano, la foto profilo, il nome e la descrizione della pagina.
* Post
  + L'admin o il dipendente inseriscono il contenuto del post, che può essere solo testo o un immagine/video con del testo;
  + Il sistema salva sul database il post come stringa in formato MD;
  + Il sistema mostra il post nel profilo dell'azienda.
* Seguire un’azienda
  + L'utente sceglie di seguire un’azienda;
  + Il sistema aggiunge la relazione nel database;
  + Il sistema invia una notifica all'azienda.
* Visualizzazione commenti
  + L'utente sceglie un post;
  + Il sistema mostra una lista dei commenti degli altri utenti riguardanti il post.
* Inserimento commenti
  + L'utente sceglie un post;
  + L'utente inserisce un commento;
  + Il sistema salva il commento nel database.
* Like sui post e sui commenti
  + L'utente sceglie un contenuto a cui mettere like, che può essere un post o un commento;
  + Il sistema aggiunge la relazione al database.
* Cancellazione contenuti
  + Il moderatore sceglie un contenuto non idoneo;
  + Il sistema lo cancella dal database.

# **Interfaccia utente**

<https://www.figma.com/proto/EfkIutUnxIslAAIMY1i67t/PROGETTO?type=design&node-id=44-331&scaling=scale-down&page-id=0%3A1>