

# Catania Discovery

Stefano Borzi - X81000003

Davide Costa - X81000007

Danilo Tascone - X81000050

## Requisiti:

Catania Discovery è una web application, in particolare, essa è una **single-page-app (SPA)** scritta in **MEAN.js**, quindi sviluppata in MongoDB, Expressjs, Angularjs, Node.js utilizzando la piattaforma Microsoft di Cloud Computing denominata **Azure**. La funzione principale dell'applicazione è fornire un servizio, efficiente ed affidabile, che riesca a mettere in contatto più agevolmente le persone. Il settore in cui si concentra l'applicazione è quello turistico, nello specifico, si mira a fornire una piattaforma in cui avviene una netta distinzione tra gli utenti che ne fanno uso. L'utente finale potrà infatti sfruttare i servizi offerti dall'applicazione sia sotto le spoglie di un turista, sia come un possibile venditore.

Sarà quindi disponibile un servizio di *autenticazione* che fornisca all'utente la possibilità di scegliere a quale delle due categorie appartenere.

Più dettagliatamente, un turista avrà la possibilità di:

- Scoprire offerte di vario genere, a lui dedicate.
- Scegliere di acquistare una delle offerte disponibili.
- Mettersi in contatto con il venditore dell'offerta a cui è interessato.
- Gestire le proprie offerte tramite una pagina dedicata.
- Gestire un profilo, con varie informazioni personali.

Dall'altra parte un venditore avrà invece la possibilità di:

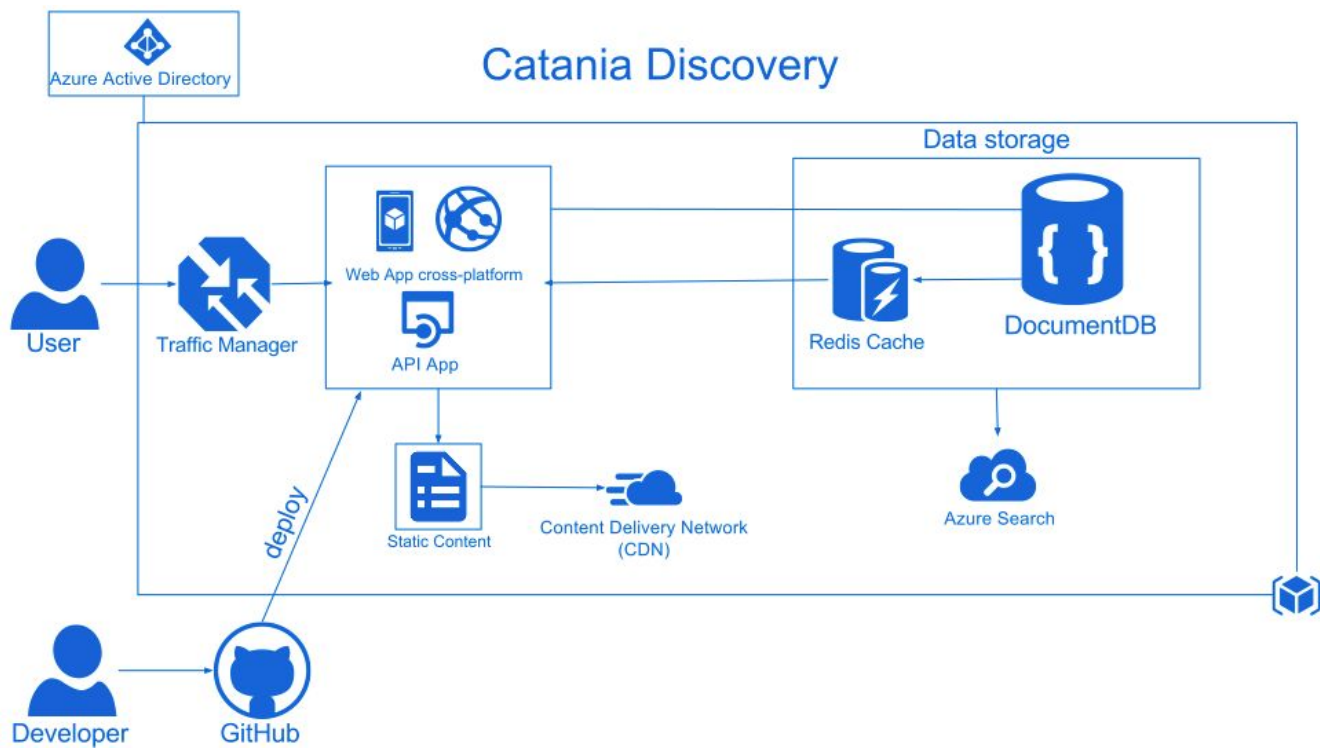
- Scoprire le offerte messe a disposizione dalla piattaforma.
- Pubblicare un'offerta da lui personalizzabile tramite una pagina dedicata.
- Accedere alle proprie offerte messe a disposizione degli utenti ed eventualmente modificarle.
- Gestire un profilo, con varie informazioni personali.

Il cuore pulsante dell'applicazione è l'offerta. Ad essa sono dedicate le rispettive pagine di visualizzazione per utenti e venditori, e le relative operazioni.

Un'offerta infatti si compone di:

- Titolo, indicativo della tipologia dell'offerta.
- Prezzo.
- Quantità disponibile.
- Descrizione, con informazioni maggiormente dettagliate riguardo cosa viene effettivamente offerto all'utente.
- Immagine descrittiva.

## Architettura e Design:

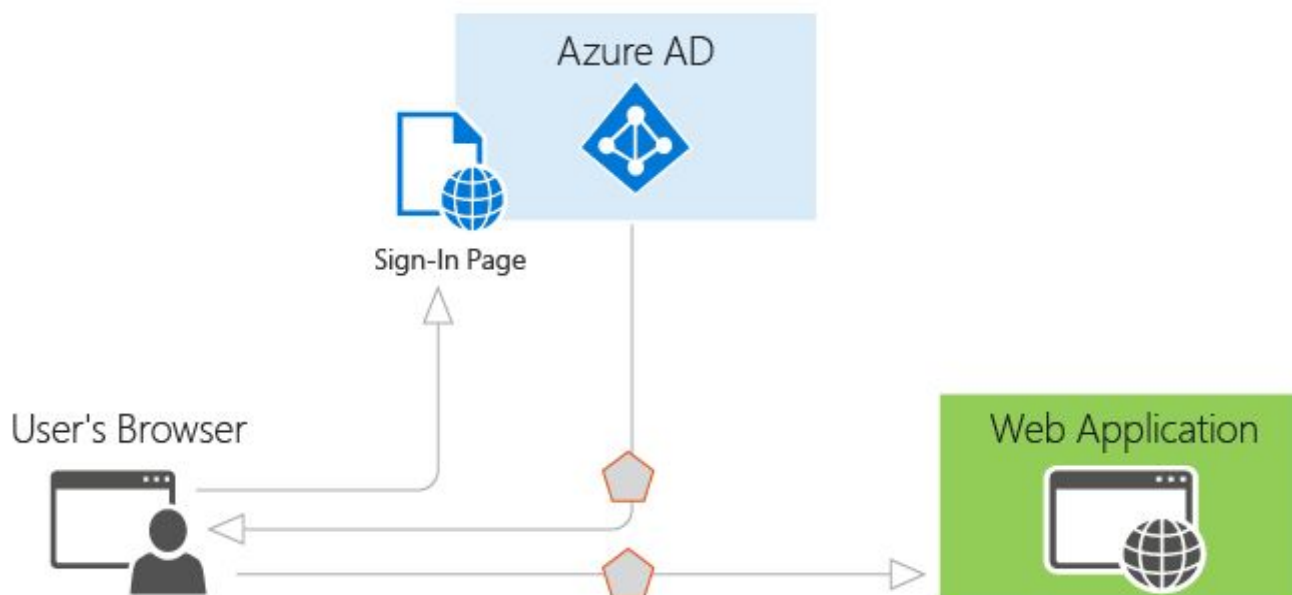


Analisi Top-Down dei servizi di Azure che sono stati utilizzati per lo sviluppo dell'applicazione:

### Azure Active Directory



E' il servizio offerto da Azure di gestione delle identità.





## Traffic Manager

Questo servizio è stato utilizzato in fase di sviluppo poiché consente di controllare la distribuzione del traffico utente per gli endpoint di servizio in diversi data center. Questo per avere una maggiore velocità di risposta e prestazioni più elevate.

Inoltre essendo l'applicazione scalabile per qualsiasi città/regione, si presenta così facilmente espandibile e ottimizzata a tale operazione. Un possibile scenario è quello in cui vengono ampliate le istanze del servizio. L'applicazione verrà quindi ospitata in diverse aree di Azure. Verrà quindi facilmente creato un profilo di gestione del traffico per distribuire il traffico client all'endpoint più vicino.

---

## Data Storage



### DocumentDB

Il servizio ha lo scopo di connettere il DB utilizzato direttamente all'applicazione. Per lo storage di dati è stato scelto un database non relazionale **MongoDB**. Esso si presenta con la seguente struttura:

- CITY:
  - *city*: nome della città in formato stringa.
  - *province*: provincia della città in formato stringa.
- OFFER:
  - *user\_id*: chiave associata ad ogni utente.
  - *user*: nickname scelto dall'utente in formato stringa.
  - *title*: titolo dell'offerta in formato stringa.
  - *price*: prezzo dell'offerta in formato numerico.
  - *quantity*: quantità offerta in formato numerico.
  - *description*: breve descrizione indicativa dell'offerta in formato stringa.
  - *img\_path*: immagine associata all'offerta in formato stringa.
  - *data*: data in cui viene pubblicata l'offerta nel formato date.
  - *clients*: interessati all'offerta in formato stringa.

- OFFERTYPE:
    - *date\_start*: data d'inizio dell'offerta in formato date.
    - *date\_end*: data di fine dell'offerta in formato date.
    - *quantity*: quantità offerta in formato numerico.
    - *user\_id*: chiave associata all'utente proprietario dell'offerta.
  - USER:
    - *nickname*: identificativo scelto dall'utente in formato stringa.
    - *name*: nome proprio dell'utente in formato stringa.
    - *surname*: cognome dell'utente in formato stringa.
    - *password*: password associata all'account dell'utente in formato stringa.
    - *email*: email dell'utente in formato stringa.
    - *phone*: numero di telefono in formato stringa.
    - *date*: data di nascita dell'utente in formato date.
    - *propic*: immagine del profilo dell'utente in formato stringa.
    - *seller*: flag per distinguere l'utente tra venditore e offerente in formato Boolean.
    - *blocked*: flag per segnalare un utente bloccato in formato Boolean.
    - *description*: breve descrizione presente nel profilo utente in formato stringa.
- 

## Redis Cache



La gestione dello Storage dell'applicazione potrebbe essere ottimizzata utilizzando il servizio offerto da Azure di Redis Cache che consente l'accesso sicuro e dedicato a una cache gestita da Microsoft e accessibile da qualsiasi applicazione in Azure. Nonostante le già ottime prestazioni del database MongoDB il servizio di redis cache migliorerebbe la reattività dell'applicazione con l'aumentare del numero di utenti, sfruttando le caratteristiche proprie di Redis.

---

## Azure Search



Strettamente collegato con la gestione dei dati è il servizio di ricerca offerto da Azure. Esso consiste in una soluzione di ricerca distribuita come servizio cloud che delega la gestione a Microsoft. Oltre alla ricerca convenzionale di testo o comunque dati inerenti l'applicazione il servizio potrebbe essere esteso, grazie alle funzionalità che offre, ad una funzione specifica per il nostro caso d'uso, ovvero la ricerca geografica. Consentire quindi agli utenti di ricercare un'offerta in base alla sua posizione geografica per migliorare la loro esperienza.



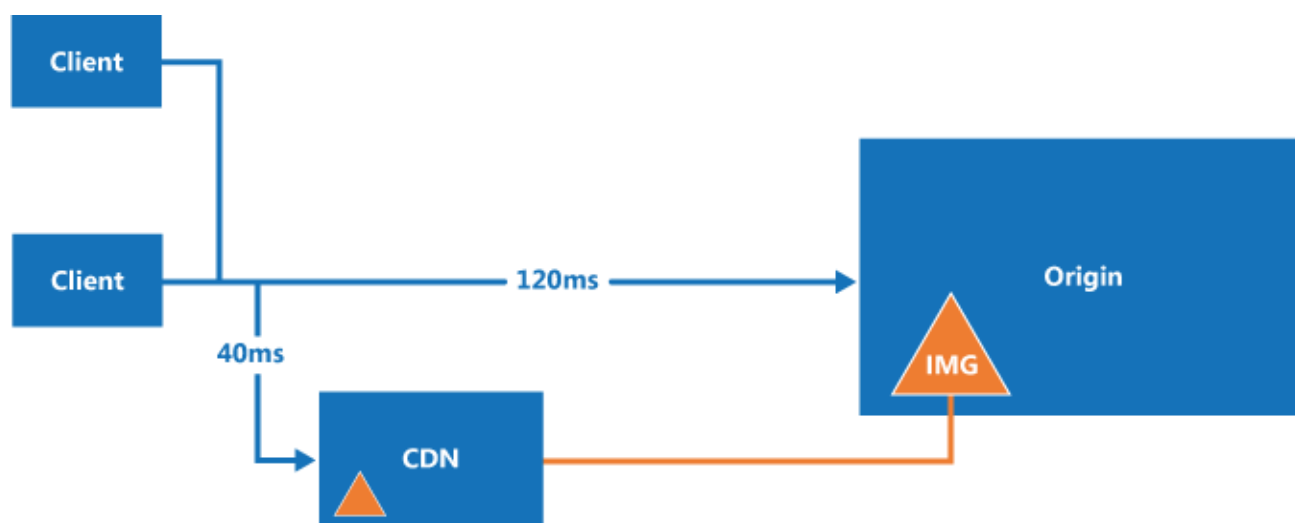
## API app

Il servizio offre un ampio supporto a **Node.js** utilizzato per lo sviluppo lato back-end dell'applicazione. Azure permette infatti di includere nel dominio dell'applicazione più "app" specializzate, ad esempio un'applicazione essendo costituita da più tipologie di servizi app, come in questo caso può distribuire il codice front-end a un'app WEB e il codice back-end a un'app per le API. Quest'ultima permette di semplificare lo sviluppo, più sicurezza, controllo di accesso semplificato, e semplicità nell'includere nell'applicazione app per la logica offerte da Azure stesso.



## CDN

Azure offre una soluzione per la distribuzione dei contenuti dell'app. Memorizzando infatti degli oggetti è possibile trattenerli in una cache in posizione strategica per migliorare la velocità di risposta dell'applicazione e l'esperienza utente. L'utilizzo del servizio permette la riduzione dei tempi di caricamento dei contenuti dell'app sia statici che dinamici. Nello sviluppo dell'applicazione è stata fornito un possibile esempio di utilizzo con dei dati statici, ma largamente espandibile con dati dinamici, come ad esempio video interattivi inerenti le offerte.

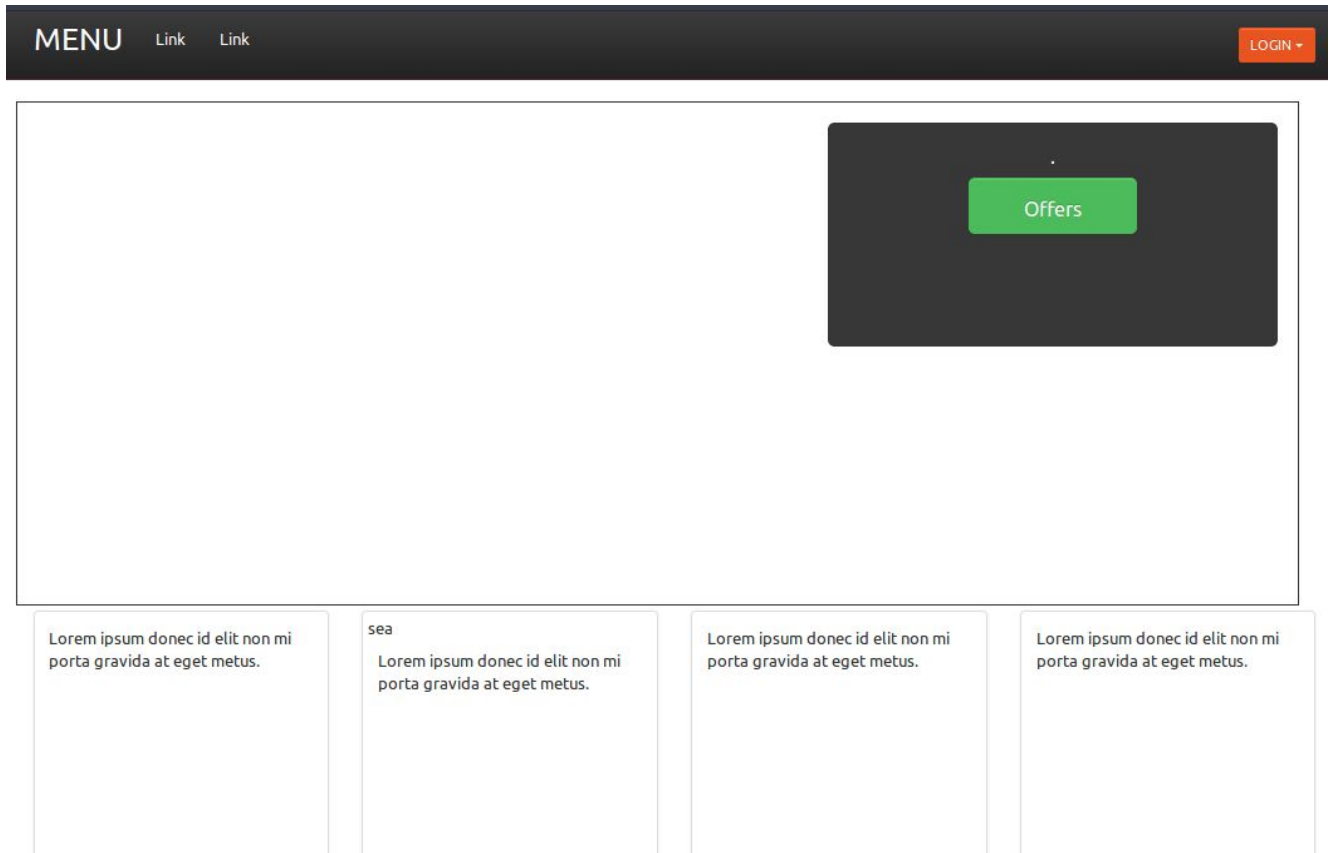


## GitHub

Utilizzato per coordinare il lavoro degli sviluppatori, come piattaforma di condivisione del codice, di pianificazione dei task da compiere in cui è stato diviso il progetto principale, e come unica repository del codice sviluppato. Pienamente supportato da Azure.

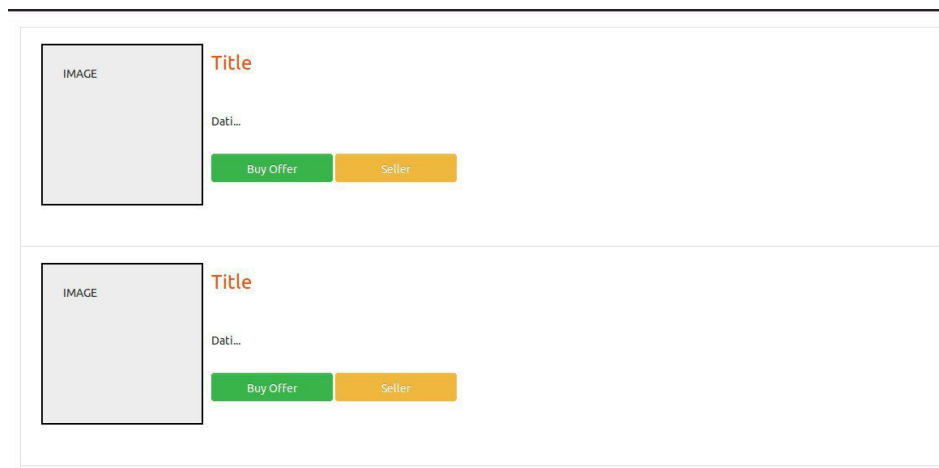
# Mockup

## Home page



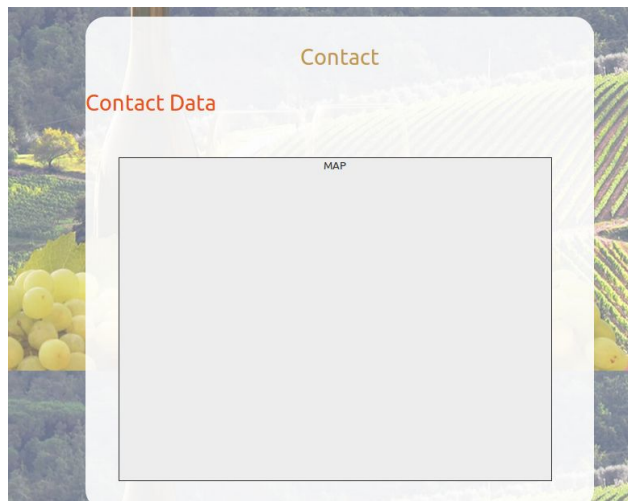
*Layout: barra di navigazione superiore con link di riferimento, dropdown a link per registrazione e per login da parte dell'utente, visualizzazione dei contenuti più rilevanti nel pannello principale, visualizzazione di contenuti secondari nei pannelli sottostanti.*

## Offer Page



*Layout: Classico shopping-cart model di visualizzazione delle offerte disponibili, ogni offerta viene visualizzata con la relativa immagine il titolo posto in rilievo, una breve descrizione a seguire e le possibilità di acquistare o contattare il venditore.*

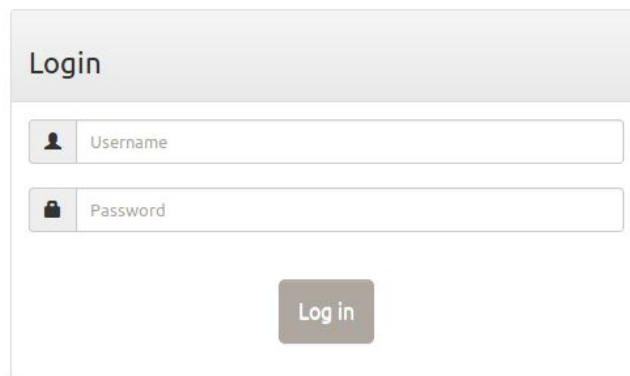
## Contact Page



Layout: Semplice container con le informazioni su come contattare gli sviluppatori e relativa mappa.

---

## Login

A mockup of a login form. It has a light gray header with the word "Login". Below the header are two input fields: the first is labeled "Username" with a person icon, and the second is labeled "Password" with a lock icon. A "Log in" button is centered below the fields.

Layout: Schermata di Login con immissione delle credenziali

---

## Profilo Utente

A mockup of a user profile page. On the left is a vertical light gray bar containing the text "utente" at the top, a placeholder box labeled "IMAGE", and a list of user data: "Dati utente", "- email", "- nome", and "- cognome". To the right of this bar is a section titled "Descrizione utente" with two horizontal lines, one labeled "Turista/Seller" and the other "Descrizione utente".

Layout: Barra di informazione posta lateralmente a sinistra, con nome utente, immagine del profilo e dati rilevanti a seguire. Completano la pagina un pannello con una descrizione più dettagliata dell'utente, l'informazione se si tratta di un venditore o di un turista, e le eventuali attività.