Voicefork

Human Computer Interaction On The Web Project

A.Y. 2022-2023

Faculty of Ingegneria dell'informazione, informatica e statistica
Department of Informatica

Group Members:
Corsi Danilo - 1742375
Lucciola Alessio - 1823638
Scarcelli Domiziano - 1872664

Voicefork

Voicefork

- Voice interface for making reservations in restaurants

Voicefork

- Voice interface for making reservations in restaurants
- Developed as Alexa skill using NodeJS

Voicefork

- Voice interface for making reservations in restaurants
- Developed as Alexa skill using NodeJS
- Collects information to simplify and speed up the booking process

Voicefork

- Voice interface for making reservations in restaurants
- Developed as Alexa skill using NodeJS
- Collects information to simplify and speed up the booking process
- Use a "Tripadvisor European Restaurants" dataset to test the system
  - Focus on Italian restaurants to simulate realistic reservation scenarios

Collection of requirements

Collection of requirements

- Questionnaire, analysis of competitors and
  online papers to make decisions on system implementation

Collection of requirements

- Questionnaire, analysis of competitors and online papers to make decisions on system implementation
- **Main concerns:**

Collection of requirements

- Questionnaire, analysis of competitors and online papers to make decisions on system implementation
- **Main concerns:**
    - Avoid errors during the booking process for a positive user experience

Collection of requirements

- Questionnaire, analysis of competitors and online papers to make decisions on system implementation
- **Main concerns:**
    - Avoid errors during the booking process for a positive user experience
    - Implement a feedback mechanism to provide clear and accurate information
        - Enable the user to have control and awareness of the booking status

Collection of requirements

- Questionnaire, analysis of competitors and online papers to make decisions on system implementation
- **Main concerns:**
    - Avoid errors during the booking process for a positive user experience
    - Implement a feedback mechanism to provide clear and accurate information
        - Enable the user to have control and awareness of the booking status
    - Smooth and natural conversation with use of feedback only at crucial moments for both experienced and casual users

Collection of requirements

- Questionnaire, analysis of competitors and online papers to make decisions on system implementation
- **Main concerns:**
  - Avoid errors during the booking process for a positive user experience
  - Implement a feedback mechanism to provide clear and accurate information
    - Enable the user to have control and awareness of the booking status
  - Smooth and natural conversation with use of feedback only at crucial moments for both experienced and casual users
  - Reduce disambiguation between restaurants with similar names

Alexa skill

Alexa skill

- Using the Alexa Developer Console platform for skill development

Alexa skill

- Using the Alexa Developer Console platform for skill development
- Activating the skill with the keyword "open restaurant reservation"

Alexa skill

- Using the Alexa Developer Console platform for skill development
- Activating the skill with the keyword "open restaurant reservation"
- Step-by-step user's guide to making a reservation
    - **Essential fields:** restaurant name, date, time and number of people

Alexa skill

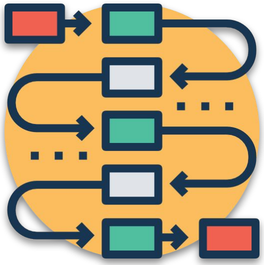
- Using the Alexa Developer Console platform for skill development
- Activating the skill with the keyword "open restaurant reservation"
- Step-by-step user's guide to making a reservation
  - **Essential fields:** restaurant name, date, time and number of people
- Ability to provide reservation information in any order

Alexa skill

- Using the Alexa Developer Console platform for skill development
- Activating the skill with the keyword "open restaurant reservation"
- Step-by-step user's guide to making a reservation
  - **Essential fields:** restaurant name, date, time and number of people
- Ability to provide reservation information in any order
- Using Alexa's error handling system to correct invalid answers

Alexa skill

- Using the Alexa Developer Console platform for skill development
- Activating the skill with the keyword "open restaurant reservation"
- Step-by-step user's guide to making a reservation
  - **Essential fields:** restaurant name, date, time and number of people
- Ability to provide reservation information in any order
- Using Alexa's error handling system to correct invalid answers
- Lambda functions to handle skill:
  - **LaunchRequestHandler:** captures the "invocation phrase"
  - **MakeReservationIntent:** collects the essential fields
  - **ReservationContextResponseHandler:** manage the reservation



Alexa skill

- Reservation management function that calculates user context and handles disambiguation

Alexa skill

- Reservation management function that calculates user context and handles disambiguation
- Need to provide the user's location to narrow the search for restaurants
  - Using the user's coordinates to search for nearby restaurants
  - Ability to specify a different location even if you have the coordinates

Alexa skill

- Reservation management function that calculates user context and handles disambiguation
- Need to provide the user's location to narrow the search for restaurants
  - Using the user's coordinates to search for nearby restaurants
  - Ability to specify a different location even if you have the coordinates
- Request to specify a location in case you do not have the coordinates to continue the reservation process

Restaurant Score

We aim to obtain a score in [0,1] for each restaurant.

The score is based on:

- Distance between query and restaurant name
- The current context
- Geographical distance between restaurant and the user

Restaurant Score - Pipeline

1. User inserts all the necessary information (restaurant name, reservation day, number of people…)
2. The system search the most similar restaurants using the given restaurant name (query)
3. For each restaurant we have a $\mathbf{d_n} \in [0, 1]$ (the lower, the better);
4. Restaurants with a $\mathbf{d_n}$ higher than a threshold are discarded
5. Generate a `ReservationContext` object for each restaurant and a $\mathbf{d_c} \in [0, \infty]$

Restaurant Score - What is the Context?

It models the context in which the user is.

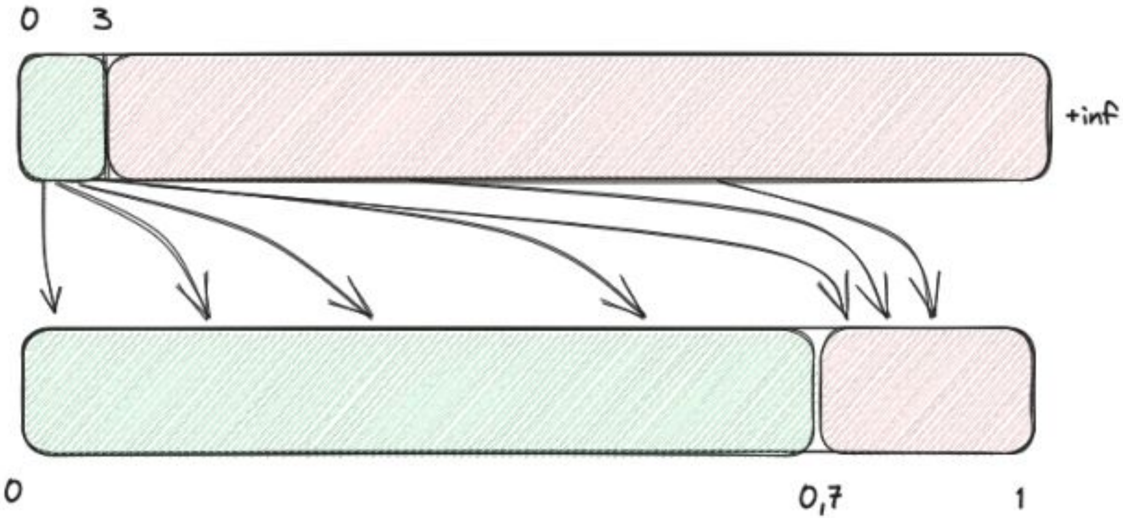
Contains the following information:

- Current day and hour
- Day and hour of the reservation
- Location when the user made the reservation
- Number of people

Distance between current and average context

Restaurant Score - Normalize Context Distance

To compute the aggregate score, we have to normalize the values of $\mathbf{d}_c$



Restaurant Score - Aggregated Score

Now we are ready to aggregate the distances into a single score:

Three cases:

1. Both $\mathbf{d_n}$ and $\mathbf{d_c}$:

$$\text{score} = 1 - ((1 - w) \cdot d_n + w \cdot \text{normalized}(d_c))$$

2. Only $\mathbf{d_n}$ ($\mathbf{d_c}$ = **`null`**):

$$\text{score} = 1 - \min(\max(d_n, 0.05)^{0.5}, 1)$$

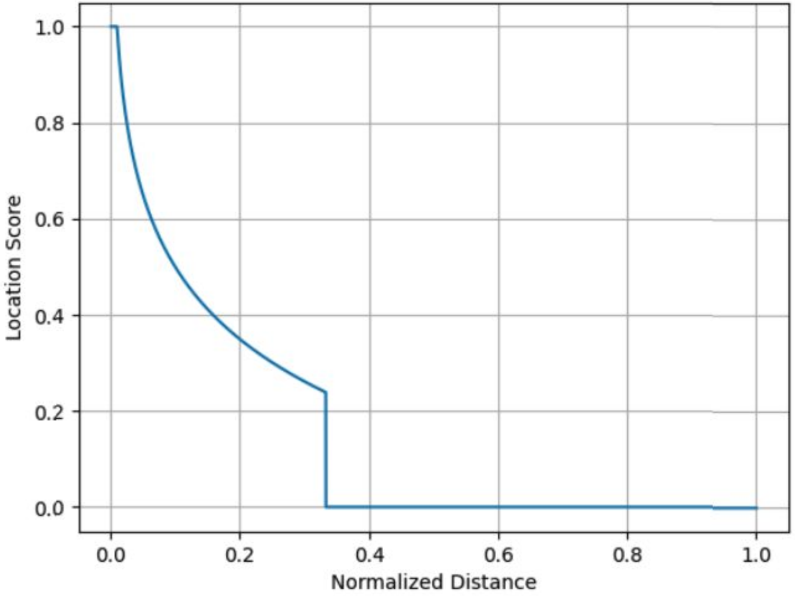
3. Each restaurant have $\mathbf{d_c}$ = **`null`**:

$$\text{score} = 1 - d_n$$

Restaurant Score - Location Boost

Score boost to places near the user

$$\begin{cases} s_l = \min\left(\log_{100}\left(\frac{1}{\text{normalized}(d_l)}\right), 1\right) & \text{if } d_l < \frac{50{,}000}{3} \\ s_l = 0 & \text{otherwise} \end{cases}$$



Restaurant Disambiguation - Buckets

- Each restaurant has a score from 0 to 1 (the higher the score, the higher the confidence that the restaurant is the desired one)
- Restaurants divided in **buckets** based on the scores:
    1. High confidence bucket: Scores in the range [0.6, 1]
    2. Medium confidence bucket: Scores in the range [0.4, 0.6)
    3. Low confidence bucket: Score in the range [0.1, 0.4)
- **We only select the first bucket that has at least one restaurant** and we try to disambiguate on that bucket

Restaurant Disambiguation - Pipeline

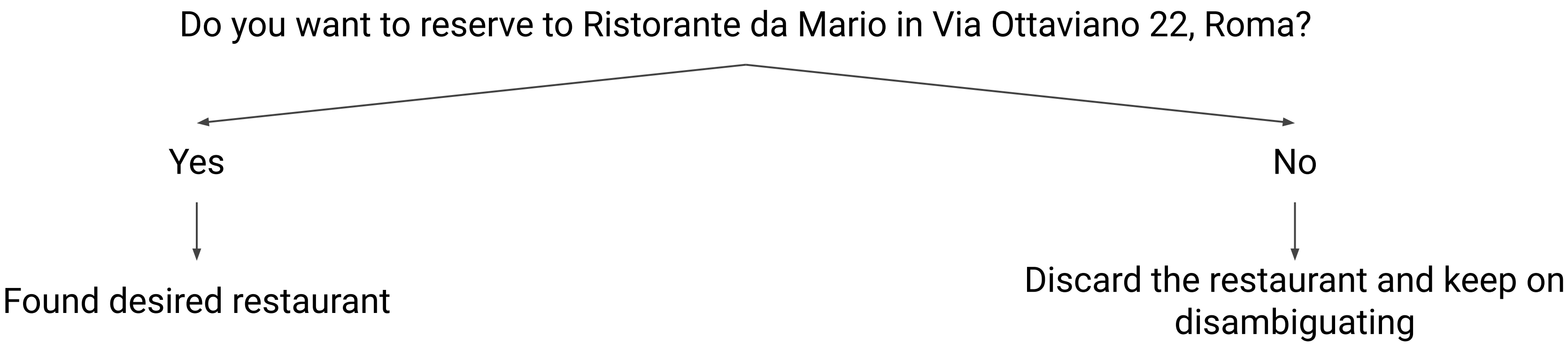
Ristorante da Mario, Roma
Ristorante da Marione, Roma
Ristorante da Mario, Tivoli

We may still have multiple restaurants in the selected bucket. If the number of restaurants is:

- **One**: We found the desired restaurant (end)
- **Two**: Immediately ask the user if they want to reserve to the one with the highest score
- **More than two**: Iterative process to try to guess the desired restaurant making some questions about the **distance**, the **city**, the **type of cuisine** and the **average score**

Restaurant Disambiguation - LatLon

LatLon (Distance): Take the position of the best restaurant and ask if that’s the desired one.



Do you want to reserve to Ristorante da Mario in Via Ottaviano 22, Roma?

Yes

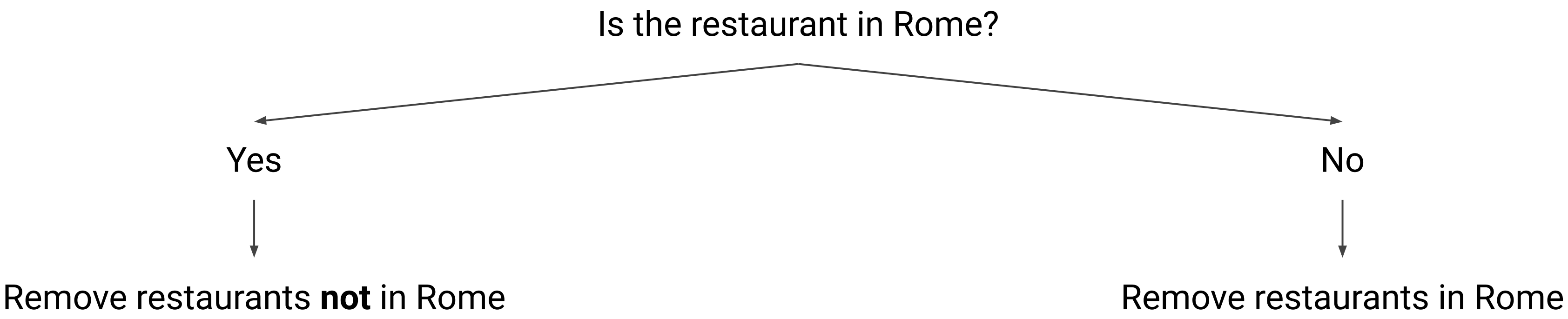
Found desired restaurant

No

Discard the restaurant and keep on disambiguating

Restaurant Disambiguation - City

City: Take the city of the best restaurant and ask if the restaurant is in that city. Useful to remove restaurants in multiple cities.



Is the restaurant in Rome?

Yes

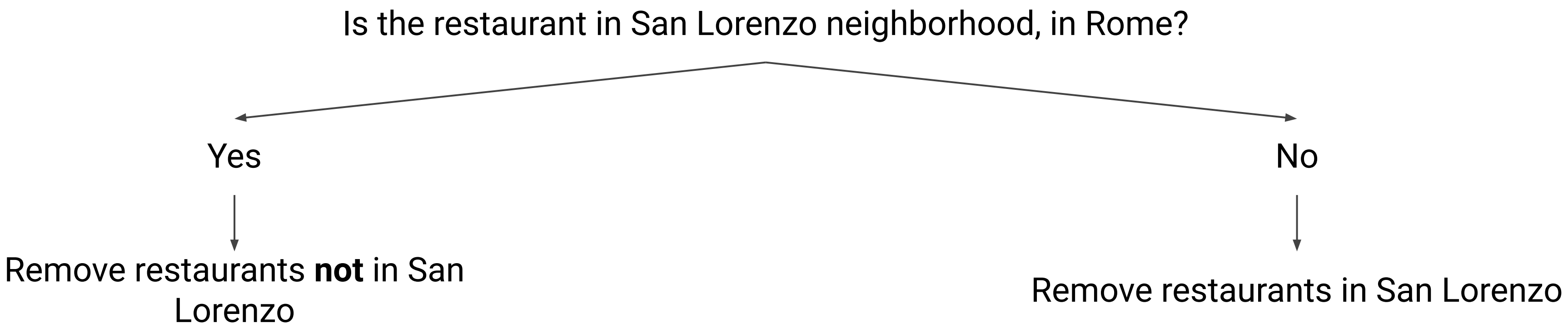
No

Remove restaurants **not** in Rome

Remove restaurants in Rome

Restaurant Disambiguation - City (zone)

City (zone): It works with neighborhoods too. Useful if restaurants are all in the same city but in different neighborhoods.



Is the restaurant in San Lorenzo neighborhood, in Rome?

Yes

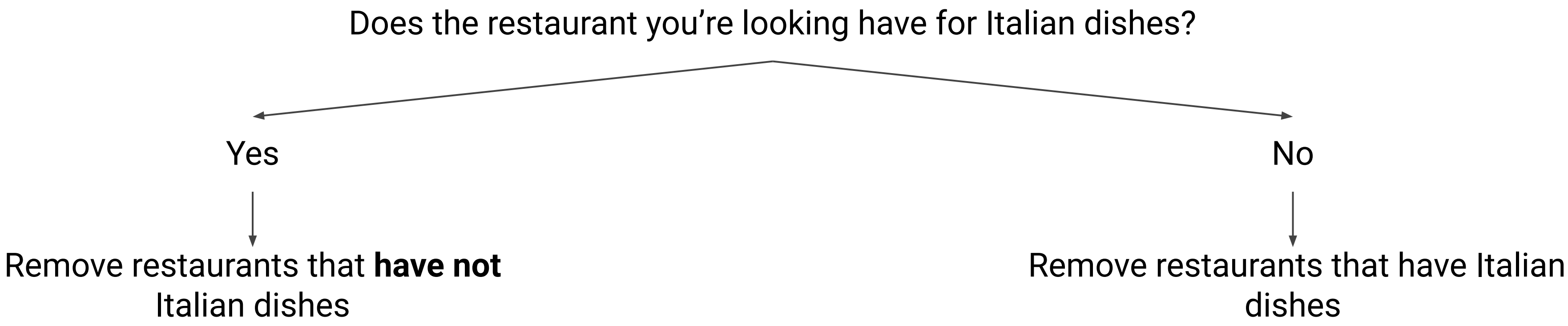
Remove restaurants **not** in San Lorenzo

No

Remove restaurants in San Lorenzo

Restaurant Disambiguation - Cuisine

Cuisine: Take the most discriminative cuisine and ask if the desired restaurant has that type of cuisine.



Does the restaurant you’re looking have for Italian dishes?

Yes

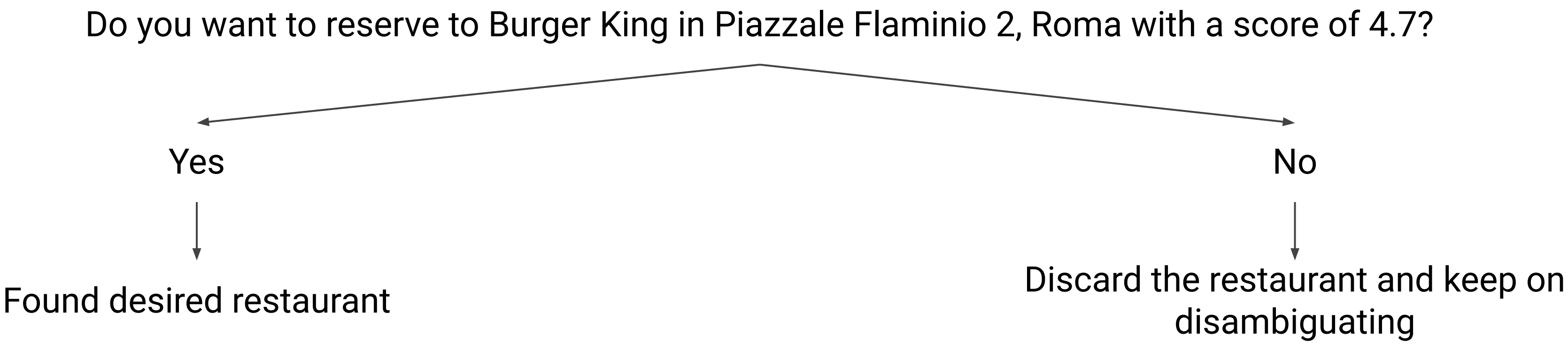
Remove restaurants that **have not** Italian dishes

No

Remove restaurants that have Italian dishes

Restaurant Disambiguation - AvgRating

Average rating: In case of high uncertainty, ask the user if he wants the restaurant with the best rating score.



Do you want to reserve to Burger King in Piazzale Flaminio 2, Roma with a score of 4.7?

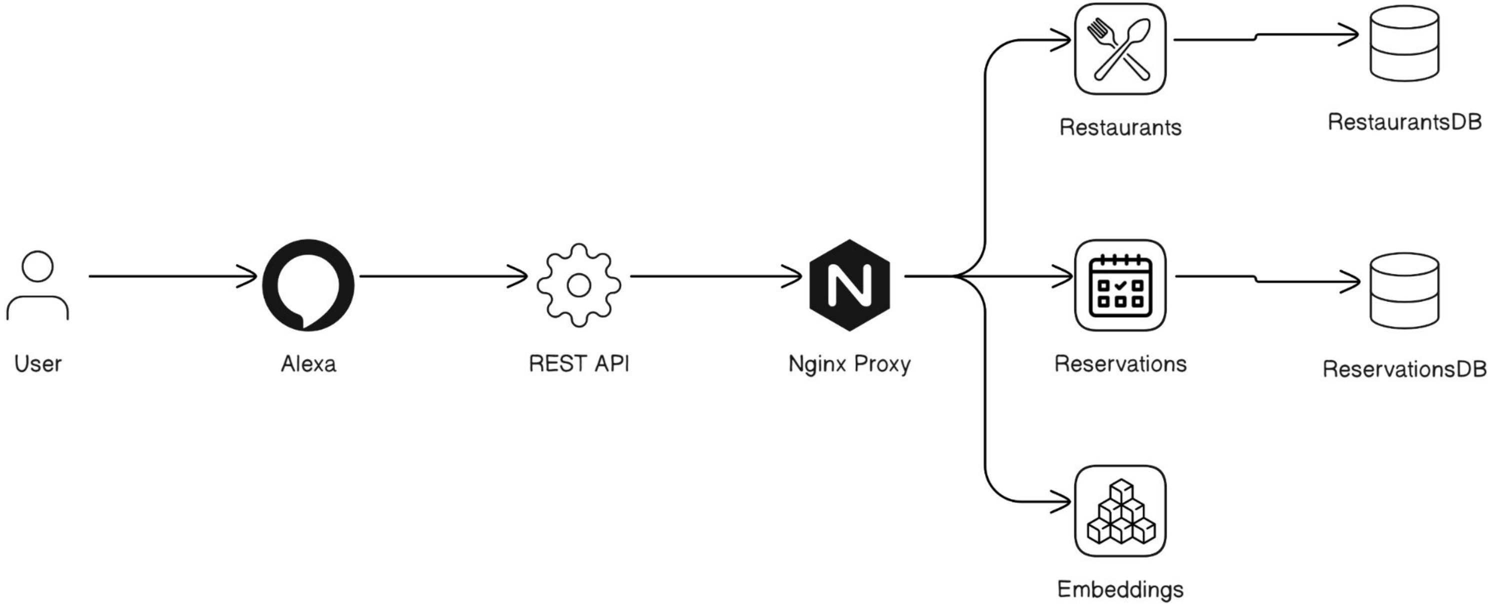
Yes

Found desired restaurant

No

Discard the restaurant and keep on disambiguating

Backend


User
Alexa
REST API
Nginx Proxy
Restaurants
RestaurantsDB
Reservations
ReservationsDB
Embeddings

Tests

- Different users
- **Think aloud** approach
- Tests both in English and Italian
- Several iterations in order to solve issues in the previous versions

You can find the results of the tests and a demo showing the system in the attached material

Thanks for the attention