<u>PROJECT PROPOSAL Revised</u>

BIZCONNECT

COMS W4111 - Part 1 (Front-End Option)

Animesh Bhasin (ab5051), Arko Kundu (apk2153)

Description

We are building an application that will enable 2 main things:

- Help a business connect with other businesses/individuals
- Help individuals connect with businesses

Our application will take an inspiration from dating apps such as tinder, but cater to businesses & individuals. Businesses can match with individuals(freelancers etc.)/other businesses to cater to their business needs.
Individuals can also find opportunities at businesses they are interested in .

Once both the parties connect on our app, they can do a chat/video meeting directly on our platform , or it can be a physical meeting which can also be booked via our platform.

Eg.1. Business <-> Business
Business A is looking for an organisation to build their website. Business B is an organisation that has expertise in building websites and is looking for a new client. Business A & Business B match on our platform (based on their preferences). So, here we can define a CONNECTION to take place between Business A and Business B.

Eg2. Individuals <-> Business
Individual A is looking for a freelance project to build a website. Business B is looking for an individual to build their website -> CONNECTION.

Interesting things in this application can be the extent of ML which can be leveraged for relevant matching. Implementing 'Stable Marriage Algorithm' for better quality of matches.

Challenges

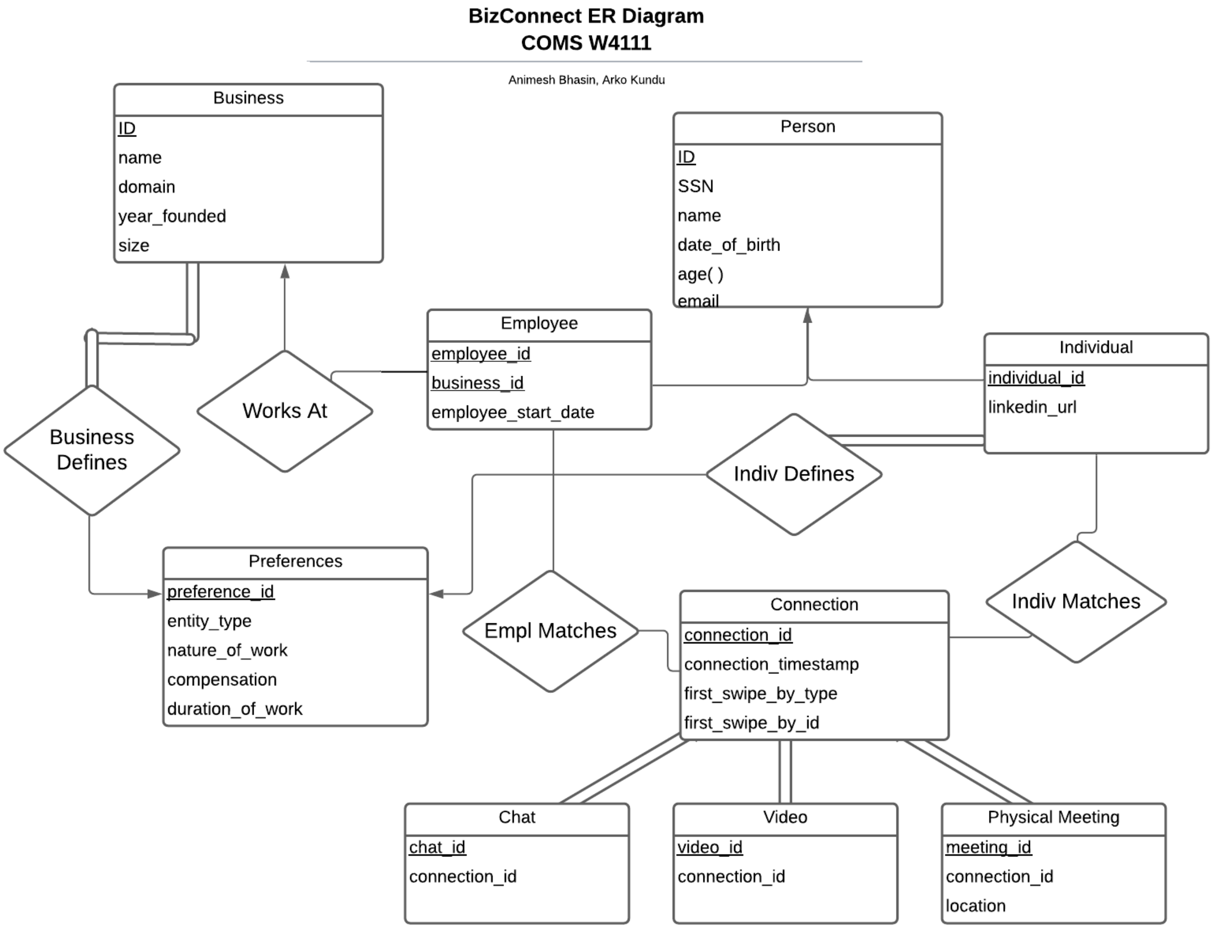
- Most entities will require dummy data as it is not available on the web
- Query to retrieve match information based on preferences might be complex.

Contingency plan

Scope to be reduced to just enabling connections via the app. Only entities that will be covered:

- Business
- Individual
- Employee
- Preferences
- Connections

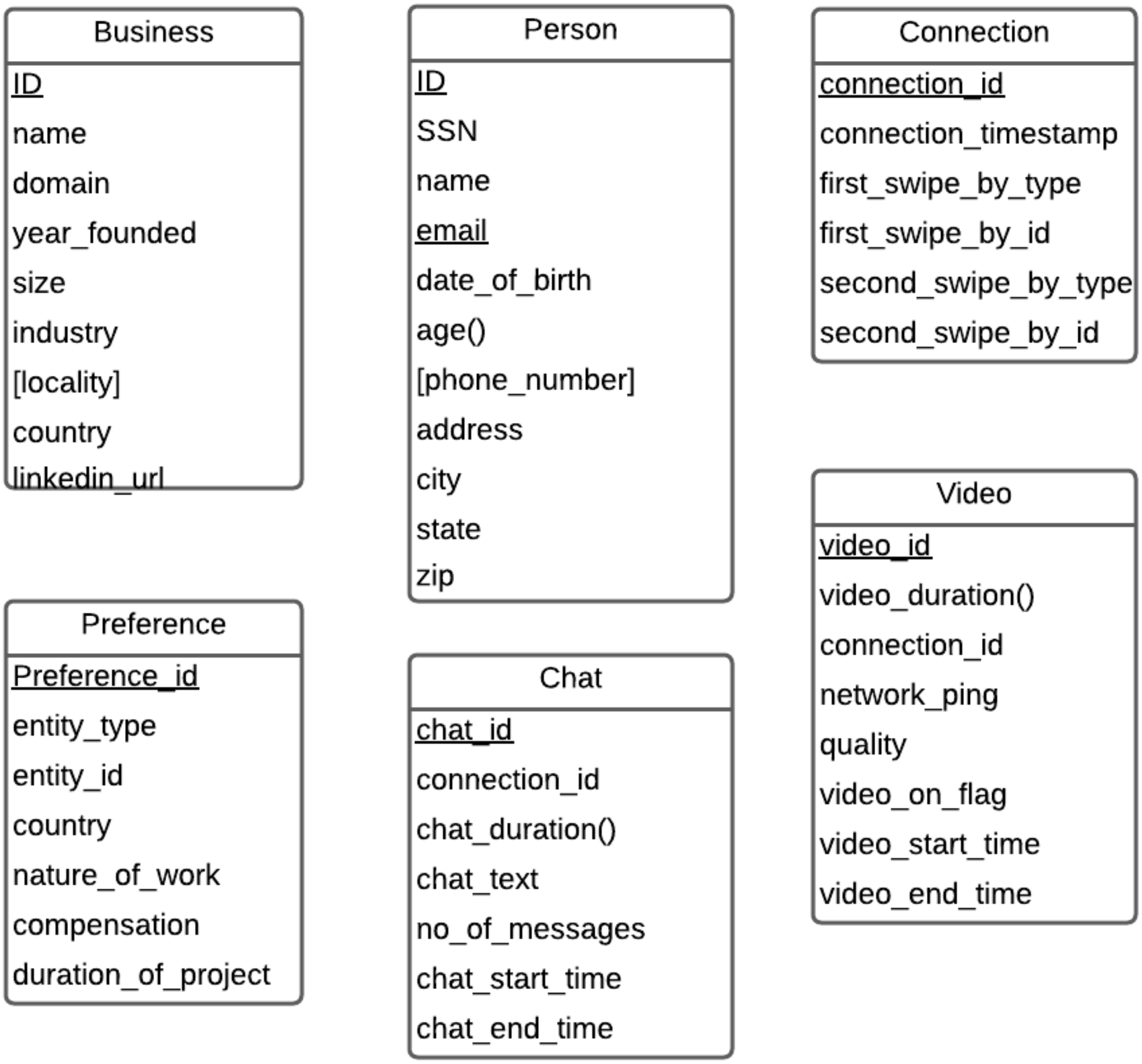
ER Diagram



Constraints

1. Age of a person should be greater than 18 years.
2. Name column in all entities should not allow NULL values
3. Foreign Key constraints enabled on tables.
4. Chat messages can not be longer than 500 characters.

Expanded Entities -



Business
- ID
- name
- domain
- year_founded
- size
- industry
- [locality]
- country
- linkedin_url

Person
- ID
- SSN
- name
- email
- date_of_birth
- age()
- [phone_number]
- address
- city
- state
- zip

Connection
- connection_id
- connection_timestamp
- first_swipe_by_type
- first_swipe_by_id
- second_swipe_by_type
- second_swipe_by_id

Preference
- Preference_id
- entity_type
- entity_id
- country
- nature_of_work
- compensation
- duration_of_project

Chat
- chat_id
- connection_id
- chat_duration()
- chat_text
- no_of_messages
- chat_start_time
- chat_end_time

Video
- video_id
- video_duration()
- connection_id
- network_ping
- quality
- video_on_flag
- video_start_time
- video_end_time

Data Plan

1. For our business entity, we will obtain data from the internet (kaggle/Bing data sets)
2. For the rest of the entities, we will generate dummy data from https://www.mockaroo.com/

SQL Syntax

```
create table business
(id varchar(64),
name varchar(200) not null,
domain varchar(30),
year_founded numeric(4,0),
industry varchar(20),
locality varchar(30),
country varchar(30),
linkedin_url varchar(30),
primary key(id));

create table person
(id varchar(64),
ssn char(9),
name varchar(200) not null,
email varchar(64),
date_of_birth date,
--age interval generated always as (current_date - date_of_birth) stored check (age>18),
phone_number numeric(10,0),
address varchar(30),
city varchar(30),
state varchar(30),
zip numeric(5,0),
primary key(id, email));

create table individual(
individual_id varchar(64),
person_id varchar(64),
linkedin_url varchar(30),
primary key (individual_id,personal_id),
foreign key (person_id) references person(id)
);

create table employee(
employee_id varchar(64),
person_id varchar(64),
ssn char(9),
business_id varchar(64),
employee_start_date date,
primary key (employee_id),
foreign key (person_id) references person(id)
foreign key (business_id) references employee_works_at_business);

create table empl_matches_connection(
employee_id varchar(64),
connection_id varchar(64),
primary key (employee_id,connection_id),
foreign key (employee_id) references employee,
foreign key (connection_id) references connection);

create table indiv_matches_connection(
individual_id varchar(64),
connection_id varchar(64),
primary key (individual_id,connection_id),
foreign key (individual_id) references individual,
foreign key (connection_id) references connection);

create table employee_works_at_business(
employee_id varchar(64),
business_id varchar(64)
primary key(employee_id,business_id),
foreign key (employee_id) references employee,
foreign key (business_id) references business);

create table preferences(
preference_id varchar(64),
entity_type varchar(64) check (entity_type in (‘business’,’individual’)),
entity_id varchar(64),
country varchar(30),
nature_of_work varchar(64),
compensation numeric(12,0)
duration_of_project numeric(4,0),
primary key(preference_id));

create table business_defined_preferences(
business_id varchar(64),
preference_id varchar(64),
primary key(business_id,preference_id),
foreign key (preference_id) references preference
foreign key(business_id) references business(id));

create table indiv_defines_preferences(
individual_id varchar(64),
preference_id varchar(64),
primary key(individual_id,preference_id),
foreign key (preference_id) references preference
foreign key (individual_id) references individual);



create table connection(
connection_id varchar(64),
connection_timestamp timestamp,
first_swipe_by_type varchar(20) check (first_swipe_by_type in (‘business’,’individual’)),,
first_swipe_by_type_id varchar(64),
second_swipe_by_type varchar(20) check (second_swipe_by_type in (‘business’,’individual’)),,
second_swipe_by_type_id varchar(64),
primary key (connection_id));

create table chat(
chat_id varchar(64),
connection_id varchar(64),
chat_duration interval generated always as (chat_end_time - chat_start_time) stored,
chat_text varchar(500),
no_of_messages numeric(5,0),
chat_start_time timestamp ,
chat_end_time timestamp ,
primary key(chat_id),
foreign key (connection_id ) references (connection));

create table video(
video_id varchar(64),
connection_id varchar(64),
video_duration interval generated always as (video_end_time - video_start_time) stored,
network _ping varchar(5),
quality varchar(10),
video_on_flag numeric(1,0),
video_start_time timestamp ,
video_end_time timestamp ,
primary key (video_id),
foreign key (connection_id ) references (connection));

create table physical_meeting(
meeting_id varchar(64),
connection_id varchar(64),
location varchar(100),
primary key (meeting_id),
foreign key (connection_id ) references (connection));
```

Questions from Design Meeting (TA : Jacob Ian Fisher)

1. Would a business define preferences or an employee of business?
2. Can we show the attributes of the entities separately rather than cluttered inside the ER diagram?
3. How can we incorporate aggregates in our ER model? Eg. Can we add Employee and business as one aggregate entity? If we do, can we still use employees in a ‘is-a’ relationship with the Person entity.
4. What other constraints can we have?

TA Feedback -

Need to change one of Defines and matches relationship (have to be unique)
No need for a lot of attributes
No need for aggregates