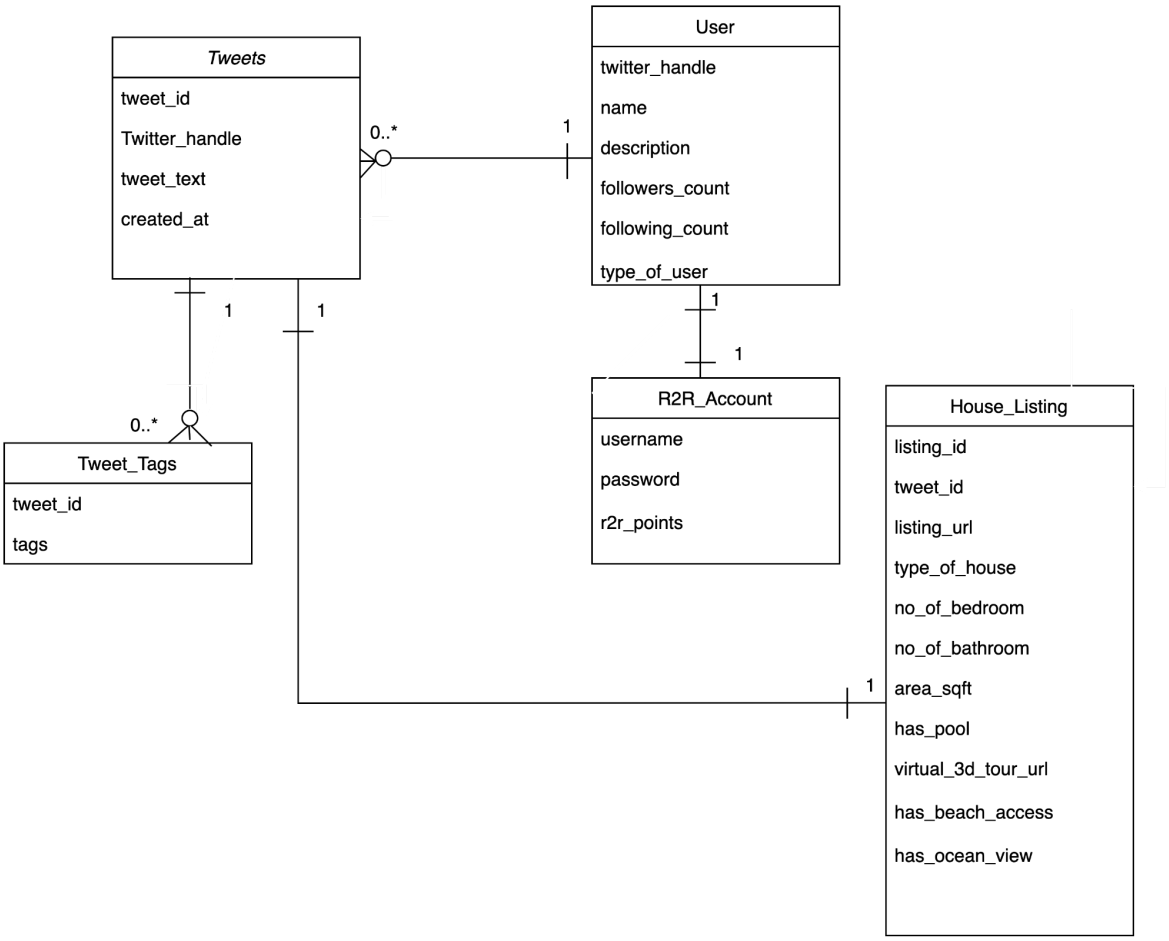
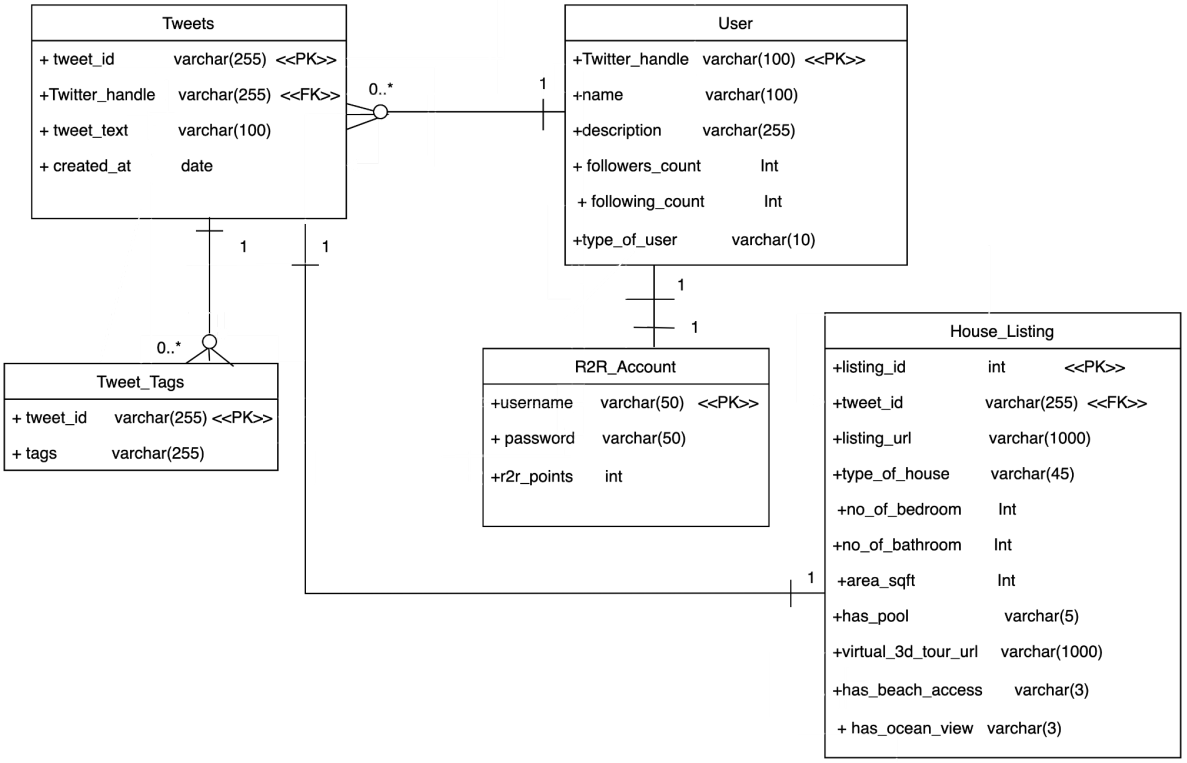
A Model on Getting Recommendations for Houses using Twitter

The online recommendation model has been updated to be more specific for a particular real estate company. The recommendation model also incorporates Twitter database schema. In this model, the user can find details about houses by tweeting his requirements. The R2R account is the main admin who can handle the recommendation model and user database.

Conceptual Model of the R2R Model



Physical Model of the R2R model



Explanation on some of the design decisions:

- The R2R account has a login and password. This login is the same as a user's Twitter handle. The Twitter handle is unique – hence it can also be treated as the primary key of the table.
- 'User' table stores data such as the user's twitter handle, twitter description, his followers and following, etc. Each user can tweet any number of tweets. The R2R-user (admin user) is also one of the users and this information can be stored in the user table itself.
- A user can ask for a house recommendation through Twitter by tweeting about his/her requirements. This data is saved in the 'Tweets' table.
- Any tags in a tweet will be saved in the 'Tweet_Tags' table.
- The 'House_Listing' table contains details extracted from a tweet posted by a seller. This table has data which is essential while buying a house. The attributes of this table are - listing_id, listing_url, the number of bedrooms in a house listing, the number of bathrooms, etc.

SQL Statements for the conceptual model

User Table:

```
CREATE TABLE `user` (
  `Twitter_handle` varchar(100) NOT NULL,
  `name` varchar(100) DEFAULT NULL,
  `description` varchar(255) DEFAULT NULL,
  `followers_count` int DEFAULT NULL,
  `following_count` int DEFAULT NULL,
  `type_of_user` varchar(10) NOT NULL,
  PRIMARY KEY (`Twitter_handle`),
  CONSTRAINT `r2r_account_fk1` FOREIGN KEY (`Twitter_handle`) REFERENCES
`r2r_account` (`username`)
);
```

Tweets Table:

```
CREATE TABLE `tweets` (
  `tweet_id` varchar(255) NOT NULL,
  `Twitter_handle` varchar(100) DEFAULT NULL,
  `tweet_text` varchar(1000) DEFAULT NULL,
  `created_at` date NOT NULL,
  PRIMARY KEY (`tweet_id`),
  KEY `user_tweet_handle_fk1` (`Twitter_handle`),
  CONSTRAINT `user_tweet_handle_fk1` FOREIGN KEY (`Twitter_handle`) REFERENCES
`User` (`Twitter_handle`)
);
```

R2R_Account Table:

```
CREATE TABLE `r2r_account` (
  `username` varchar(50) NOT NULL,
  `password` varchar(50) DEFAULT NULL,
  `r2r_points` int DEFAULT NULL,
  PRIMARY KEY (`username`)
);
```

Tweet_Tags Table:

```
CREATE TABLE `tweet_tags` (
```

```
  `tweet_id` varchar(255) NOT NULL,
  `tags` varchar(255) DEFAULT NULL,
  PRIMARY KEY (`tweet_id`),
  CONSTRAINT `Tweets_tweet_id_fk1` FOREIGN KEY (`tweet_id`) REFERENCES `Tweets`
(`tweet_id`)
);
```

House_Listing Table:

```
CREATE TABLE `house_listing` (
  `listing_id` int NOT NULL,
  `tweet_id` varchar(50) NOT NULL,
  `listing_url` varchar(1000) DEFAULT NULL,
  `type_of_house` varchar(45) DEFAULT NULL,
  `no_of_bedroom` int DEFAULT NULL,
  `no_of_bathroom` int DEFAULT NULL,
  `area_sqft` int DEFAULT NULL,
  `has_pool` varchar(5) DEFAULT 'NO',
  `virtual_3d_tour_url` varchar(1000) DEFAULT NULL,
  `has_beach_access` varchar(3) NOT NULL DEFAULT 'NO',
  `has_ocean_view` varchar(3) NOT NULL DEFAULT 'NO',
  PRIMARY KEY (`listing_id`),
  KEY `tweets_tweet_id` (`tweet_id`),
  CONSTRAINT `tweets_tweet_id` FOREIGN KEY (`tweet_id`) REFERENCES `tweets`
(`tweet_id`)
);
```

Adding Foreign Key Constraint

Constraint for Tweets Table:

```
ALTER TABLE `tweets`
ADD CONSTRAINT `user_tweet_handle_fk1` FOREIGN KEY (`twitter_handle`)
```

REFERENCES User(`Twitter_handle`);

Constraint for Tweet Tags Table:

ALTER TABLE `tweet_tags`
ADD CONSTRAINT `Tweets_tweet_id_fk1` FOREIGN KEY (`tweet_id`)
REFERENCES Tweets(`tweet_id`);

Constraint for User Table:

ALTER TABLE `user`
ADD CONSTRAINT `r2r_account_fk1` FOREIGN KEY (`twitter_handle`)
REFERENCES r2r_account(`twitter_handle`);

Constraint for House_Listing Table:

ALTER TABLE `house_listing`
ADD CONSTRAINT `tweets_tweet_id` FOREIGN KEY (`tweet_id`)
REFERENCES Tweets(`tweet_id');

Relational Algebraic Expressions of the Use Cases:

Use Case 1: Register for an account in R2R account

Use Case 2: View the properties which have more than 3 bedrooms

Π {hl.has_ocean_view, hl.listing_url} (σ{hl.no_of)bedroom > 3}(house_listing)

Use Case 3: View the properties that are studio apartments

Use Case 4: View the properties which have ocean view

Π {hl.has_ocean_view, hl.listing_url} (σ{hl.no_of)bedroom >= 3}(house_listing)

Use Case 5: View the properties that are above a particular area (say 800 sqft)

Use Case 6: View the properties containing a swimming pool

$\sigma_{\text{"has\_pool" = "YES"}}$ *house*_listing

Use Case 7: View the properties that are having a 3d virtual tour included in their listings

Null values are not represented in relational Algebra expressions

Use Case 8: View the name of the lister who has posted a listing of a 4-bedroom apartment

π t . tweet_text, t . twitter_handle, hl . no_of_bedroom

σ hl . no_of_bedroom = 4

(ρ t tweets ⋈ t . tweet_id = hl . tweet_id

ρ hl house_listing)

Use Case 9: The user has lost or forgotten the password and requests the admin for password update

N/A

Use Case 10: What are the tags used by the user in all his tweets

π t . twitter_handle, tt . tags

σ t . twitter_handle = "bobcowanrealtor"

(ρ t tweets ⋈ t . tweet_id = tt . tweet_id

ρ tt tweet_tags)

Use Case 11: Sellers who posted properties having 2 or more bathrooms

π u . twitter_handle, t . tweet_text, hl . no_of_bathroom

(ρ t tweets ⋈ t . twitter_handle = u . twitter_handle

ρ u user ⋈ t . tweet_id = hl . tweet_id

ρ hl house_listing)

Use Case 12: View all listings of a seller

$\pi_{twitter\_handle,\ tweet\_text}$

$\sigma_{twitter\_handle = \text{"bobcowanrealtor"}}\ tweets$

Use Case 13: List popular sellers

$\tau_{followers\_count\ \downarrow}\ user$

Use Case 14: Top 5 recent listings

τ created_at $\downarrow$

π t . twitter_handle, t . created_at, t . tweet_text

σ u . type_of_user = "SELLER"

(ρ t tweets $\bowtie$ t . twitter_handle = u . twitter_handle

ρ u user)

Use Case 15: View list of all buyers/sellers

$\sigma_{type\_of\_user = \text{"BUYER"}}\ user$

$\sigma_{type\_of\_user = \text{"SELLER"}}\ user$

Use Cases:

Use Case 1: Register for an account in R2R account

Description: User registers for an account in R2R account

Actor: User

Precondition: When a prospective user wants to access property records, he must have to register.

Steps:

Actor action: User request for registration

System Responses: If prospective users' information is correct then prospective users are registered and the use case ends.

Post Condition: User successfully registered

Alternate Path: The prospective users' request is not correct, and system throws an error

Error: User information is incorrect

Use Case 2: View the properties which have more than 3 bedrooms

Description: User views the properties which have more than 3 bedrooms

Actors: User

Precondition: User should be an R2R registered user

Steps:

Actor action: User views the products above a particular number of bedrooms

System Responses: The query is implemented, and the list of properties above a particular number of bedrooms is displayed

Post Condition: User can choose to view properties among the listings

Use Case 3: View the properties that are studio apartments

Description: User wants to view the properties which are studio apartments

Actors: User

Precondition: User should be an R2R registered user

Steps:

Actor action: User views the properties which have studio or apartment in their search results

System Responses: The list of properties which are studio apartments is displayed

Post Condition: User can choose to view properties among the listing

Use Case 4: View the properties which have ocean view

Description: User wants to view the properties which have ocean view

Actors: User

Precondition: User should be an R2R registered user

Steps:

Actor action: User views the properties which have ocean view in their search results

System Responses: The list of properties which have ocean view is displayed

Post Condition: User can choose to view properties among the listing

Use Case 5: View the properties that are above a particular area (say 800 sqft)

Description: User views the properties that are above a particular area in their floor size

Actors: User

Precondition: User should be an R2R registered user

Steps:

Actor action: User views the properties which that are above a particular area in their floor size.

System Responses: The list of properties which are above a particular area in their floor size is displayed

Post Condition: System displays a list of properties that are more than the mentioned floor area.

Alternate Path:

Error: Listing not available

Use Case 6: View the properties containing a swimming pool

Description: User views the properties that have a pool

Actors: User

Precondition: User should be an R2R registered user

Steps:

Actor action: User views the properties which that that have a pool or pool access

System Responses: The list of properties which have a pool or pool access is displayed

Post Condition: System displays a list of properties that have a pool

Use Case 7: View the properties that are having a 3d virtual tour included in their listings

Description: User views the properties that have listed the URL of a 3d virtual tour of the said property

Actors: User

Precondition: User should be an R2R registered user

Steps:

Actor action: User views the properties which that have listed the URL of a 3d virtual tour of the property

System Responses: The list of properties which the tour URL is displayed

Post Condition: System displays a list of properties that have the virtual tour URL is displayed

Alternate Path:

Error: Listing not available

Use Case 8: View the name of the lister who has posted a listing of a 4-bedroom apartment

Description: This would return a list of listers who have posted specific listings of 4-bedroom apartment

Actors: User

Precondition: User should be an R2R registered user

Steps:

Actor action: User views the properties which that have a listing containing a 4-bedroom apartment

System Responses: The list of properties which have 4 bedrooms is displayed

Post Condition: System displays a list of properties that have the list of all 4-bedroom apartment available

Alternate Path:

Error: Listing not available

Use Case 9: The user has lost or forgotten the password and requests the admin for password update

Description: This would allow the admin the rights to update the user password as wished by the user.

Actors: User

Precondition: User should be an R2R registered user

Steps:

Actor action: The user requests the admin for a password update

System Responses: The admin then implements the user query and updates the user password in the database

Post Condition: User password has been updated

Use Case 10: What are the tags used by the user in all his tweets

Description: This would allow us to find all the tags used by the user in all their tweets

Actors: User

Precondition: User should be an R2R registered user

Steps:

Actor action: The user could view all the tags used in their tweets

System Responses: The admin then implements query which allows us to view all the tags used by the user in all their tweets

Post Condition: View tags by user

Use Case 11: Sellers who posted properties having 2 or more bathrooms

Description: This would be the list of all sellers who have posted a listing of properties which have two or more number of bathrooms

Actors: User

Precondition: User should be an R2R registered user

Steps:

Actor action: The user views the list of all sellers who have posted a listing of properties which have two or more number of bathrooms

System Responses: The admin then implements the user query and returns a list of properties with two or more bathrooms

Post Condition: User views the queried list

Use Case 12: View all listings of a seller

Description: This would be the list of all the listings posted by a seller

Actors: User

Precondition: Seller should be an R2R registered user

Steps:

Actor action: The user views the list of all listings posted by a seller

System Responses: The query is implemented, and it returns a list of properties listed or posted by the seller

Post Condition: User' views the queried list

Use Case 13: List popular sellers

Description: This would be the list of all popular sellers

Actors: User

Precondition: Seller should be an R2R registered user

Steps:

Actor action: The user views the list of most popular seller

System Responses: The query is implemented, and it returns a list all popular sellers in accordance with their followers count in descending order

Post Condition: User views the queried list

Use Case 14: Top 5 recent listings

Description: This would be the list of top 5 latest listings

Actors: User

Precondition: Seller should be an R2R registered user

Steps:

Actor action: The user views the list of top 5 latest listings

System Responses: The query is implemented, and it returns a list of top 5 latest listings by sellers based on their time of posting in descending order

Post Condition: User views the queried list

Use Case 15: View list of all buyers/sellers

Description: This would be the list of all buyers or sellers depending on the option the user has chosen while creating the R2R account

Actors: User

Precondition: User should be an R2R registered user

Steps:

Actor action: The user views the list of all buyers or sellers

System Responses: The query is implemented, and it returns a list of all buyers and sellers

Post Condition: User views the queried list

SQL Statements:

Use Case 1: Register for an account in R2R account

Buyer registration:

```
INSERT INTO `r2r_account` (Twitter_handle, password, r2r_points)
VALUES ('ShashankDongr16', 'shashank', 100);
INSERT INTO `r2r_account` (Twitter_handle, password, r2r_points)
VALUES ('riyavirani', 'riya', 200);
INSERT INTO `r2r_account` (Twitter_handle, password, r2r_points)
VALUES ('ojasvipatel', 'ojasvi', 300);
```

Seller registration:

```
INSERT INTO `r2r_account` (Twitter_handle, password, r2r_points)
 VALUES ('bobcowanrealtor', 'bob', 300);
INSERT INTO `r2r_account` (Twitter_handle, password, r2r_points)
VALUES ('Trudeaurealtor', 'trudeau', 300);
```

Use Case 2: View the properties which have more than 3 bedrooms

```
SELECT * FROM `house_listing`
WHERE `no_of_bedroom` > 3;
```

Use Case 3: View the properties that are studio apartments

```
SELECT * FROM `house_listing`
WHERE `type_of_house` = 'Studio';
```

Use Case 4: View the properties which have ocean view

```
SELECT * FROM `house_listing`
WHERE `has_ocean_view` = 'YES';
```

Use Case 5: View the properties that are above a particular area (say 800 sqft)

```
SELECT * FROM `house_listing`
WHERE `area_sqft` > 800;
```

Use Case 6: View the properties containing a swimming pool

```
SELECT * FROM `house_listing`
```

```
WHERE `has_pool` = 'YES';
```

Use Case 7: View the properties that are having a 3d virtual tour included in their listings

```
SELECT * FROM `house_listing`
WHERE `virtual_3d_tiur_url` IS NOT NULL;
```

Use Case 8: View the name of the lister who has posted a listing of a 4-bedroom apartment

```
SELECT t.tweet_text, t.twitter_handle, hl.no_of_bedroom FROM tweets t
INNER JOIN house_listing hl ON t.tweet_id = hl.tweet_id
WHERE hl.no_of_bedroom = 4;
```

Use Case 9: The user has lost or forgotten the password and requests the admin for password update

```
UPDATE `r2r_account`
SET `password` = 'riya13'
WHERE `username` = 'riyavirani';
```

Use Case 10: What are the tags used by the user in all his tweets

```
SELECT t.twitter_handle, tt.tags FROM tweets t
INNER JOIN tweet_tags tt ON t.tweet_id = tt.tweet_id
WHERE t.twitter_handle='bobcowanrealtor';
```

Use Case 11: Sellers who posted properties having 2 or more bathrooms

```
SELECT u.twitter_handle, t.tweet_text, hl.no_of_bathroom FROM tweets t
JOIN user u ON n t.twitter_handle = u.Twitter_handle
JOIN house_listing hl ON t.tweet_id = hl.tweet_id;
```

Use Case 12: View all listings of a seller

```
SELECT `twitter_handle`, `tweet_text` FROM `tweets`
WHERE `twitter_handle` = 'bobcowanrealtor';
```

Use Case 13: List popular sellers

```
SELECT * FROM `user` ORDER BY `followers_count` desc;
```

Use Case 14: Top 5 recent listings

```
SELECT t.twitter_handle, t.created_at, t.tweet_text FROM tweets t
INNER JOIN user u ON t.twitter_handle = u.twitter_handle
WHERE u.type_of_user = 'SELLER'
ORDER BY `created_at desc`
LIMIT 5;
```

Use Case 15: View list of all buyers/sellers

```
SELECT * FROM `user`
WHERE `type_of_user` = 'BUYER';

SELECT * FROM `user`
WHERE `type_of_user` = 'SELLER';
```

GROUP MEMBER DETAILS:

Team Members:
- Name - Shashank Dongre
NUID - 002747740
Email - dongre.s@northeastern.edu
- Name - Riya Virani
NUID - 002747048
Email - virani.r@northeastern.edu
- Name - Ojasvi Patel
NUID - 002793770
Email - patel.oj@northeastern.edu

GITHUB LINK:

https://github.com/ShashankDongre/Realty-to-Reality---Find-your-dream-home-here-