

CPSC 304 Project Cover Page

Milestone #: 4

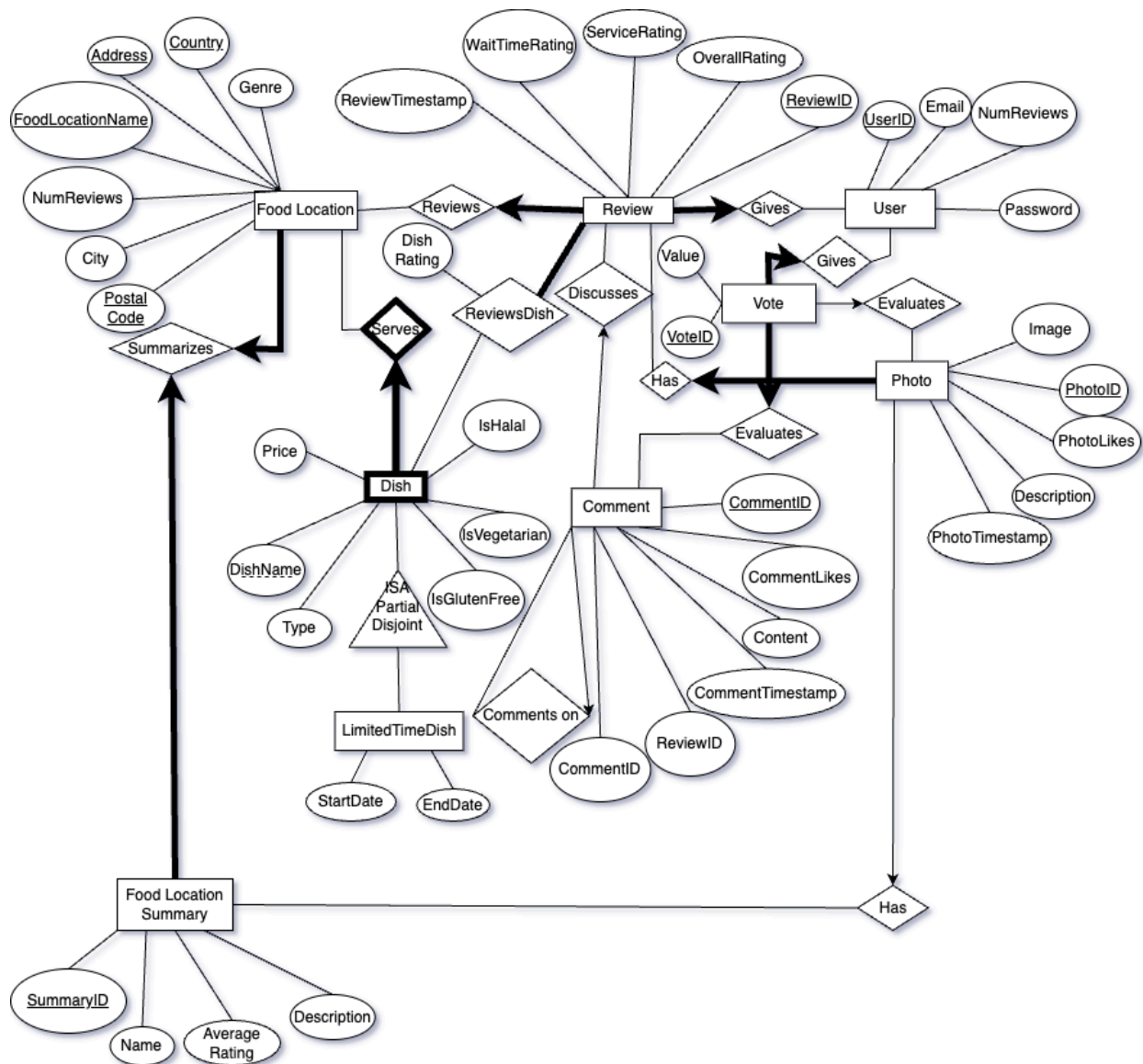
Date: November 27, 2024

Group Number: 11

Name	Student Number	CS Alias (Userid)	Preferred E-mail Address
Alexander Jacobson	66453994	h9u3i	xander.jacobson@gmail.com
Jerry Chiang	71503379	i1g9a	jeaerstar@gmail.com
Sabrina Woo	27504737	o6d9r	sabrinawoo3895@gmail.com

By typing our names and student numbers in the above table, we certify that the work in the attached assignment was performed solely by those whose names and student IDs are included above. (In the case of Project Milestone 0, the main purpose of this page is for you to let us know your e-mail address, and then let us assign you to a TA for your project supervisor.)

In addition, we indicate that we are fully aware of the rules and consequences of plagiarism, as set forth by the Department of Computer Science and the University of British Columbia



Link to repository:

https://github.students.cs.ubc.ca/CPSC304-2024W-T1/project_h9u3i_i1g9a_o6d9r

Description:

'Foodie' is a specialized restaurant and cafe review platform that allows users to leave detailed reviews on individual dishes, service, wait times, and amenities, addressing the lack of dish-specific feedback on platforms like Tripadvisor or Yelp. Users can also browse nearby food locations, filtering by fields such as ratings, restaurant names and country. The application is built using React.js for the frontend and Express.js for the backend.

Differing Schema

We did not make any changes to the relationships, but we removed some extra fields that we were not using such as the day of the week visited by the user. We also changed the data type

of a photo to an image URL instead of a BLOB. We slightly modified the fields of comment so that there are two fields, review ID and parent comment ID, rather than parent type and parent ID. Finally, for our one to one relationship between Food Location and Food Location Summary with total participation for both sides, we initially insert a Food Location with the foreign key reference to Food Location Summary set as NULL. Immediately following, we insert the corresponding Food Location Summary, then set the foreign key in Food Location appropriately.

2.1.1 INSERT

Inserts a review. Located in the file ReviewService.js, line 4.

2.1.2 UPDATE

Updates user's email. Located in the file UserService.js, line 61.

2.1.3 DELETE

Deletes a user's account. Located in the file UserService.js, line 21.

2.1.4 SELECTION

Advanced filtering results for searching for food locations, allowing users to add additional search requirements such as location and average review. Located in Home.jsx, line 23

2.1.5 PROJECTION

Allows user to view only certain fields of a dish. Located in DishService.js, line 70.

2.1.6 JOIN

Can search through users reviews on their profile page where the review contains a dish with a given name. Located in the file PhotoService.js, line 47.

2.1.7 Aggregation with GROUP BY

Displays a user's average reviews.

```
SELECT AVG(OVERALLRATING), AVG(SERVICERATING), AVG(WAITTIMERATING) FROM REVIEW
WHERE USERID=:userID GROUP BY USERID
```

2.1.8 Aggregation with HAVING

```
SELECT r.FOODLOCATIONNAME, r.ADDRESS, r.POSTALCODE, r.COUNTRY,
AVG(s.AVERAGERATING) AS AVG_RATING, COUNT(*) AS REVIEW_COUNT FROM
FOODLOCATIONSUMMARY s, REVIEW r WHERE s.FOODLOCATIONNAME = r.FOODLOCATIONNAME
AND s.ADDRESS = r.ADDRESS AND s.POSTALCODE = r.POSTALCODE AND s.COUNTRY =
r.COUNTRY GROUP BY r.FOODLOCATIONNAME, r.ADDRESS, r.POSTALCODE, r.COUNTRY
HAVING COUNT(*) > 1 AND AVG(s.AVERAGERATING) > 4
```

2.1.9 Nested aggregation with GROUP BY

```
SELECT r.FOODLOCATIONNAME, r.ADDRESS, r.POSTALCODE, r.COUNTRY FROM
FOODLOCATIONSUMMARY s, REVIEW r WHERE s.FOODLOCATIONNAME = r.FOODLOCATIONNAME
AND s.ADDRESS = r.ADDRESS AND s.POSTALCODE = r.POSTALCODE AND s.COUNTRY =
r.COUNTRY GROUP BY r.FOODLOCATIONNAME, r.ADDRESS, r.POSTALCODE, r.COUNTRY
HAVING COUNT(*) > (SELECT AVG(review_count) FROM (SELECT COUNT(*) AS
```

```
review_count FROM FOODLOCATIONSUMMARY s, REVIEW r WHERE s.FOODLOCATIONNAME =  
r.FOODLOCATIONNAME AND s.ADDRESS = r.ADDRESS AND s.POSTALCODE = r.POSTALCODE  
AND s.COUNTRY = r.COUNTRY GROUP BY r.FOODLOCATIONNAME, r.ADDRESS, r.POSTALCODE,  
r.COUNTRY))
```

2.1.10 Division

A feature that allows users to find all users that have visited every Mcdonald's in a given country. Located in the file FoodLocationService.js, line 102.

Link to Init Script

https://github.students.cs.ubc.ca/CPSC304-2024W-T1/project_h9u3i_i1g9a_o6d9r/blob/e89a900ff5c01b092b95170c27fb42831282f920/scripts/sql/Init.sql#L1