University of British Columbia, Vancouver
Department of Computer Science

CPSC 304 Project Cover Page

Milestone #: ___2____

Date: ___March 1, 2024___

Group Number: ____24_____

Name	Student Number	CS Alias (Userid)	Preferred E-mail Address
Katherine Poon	90607425	o1w1j	katherinepoon12345@gmail.com
Maya Arafa	75726604	k9i0h	arafa.maya@gmail.com
Inaki Blasco	93863819	g5m5d	inaki.blasco1@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

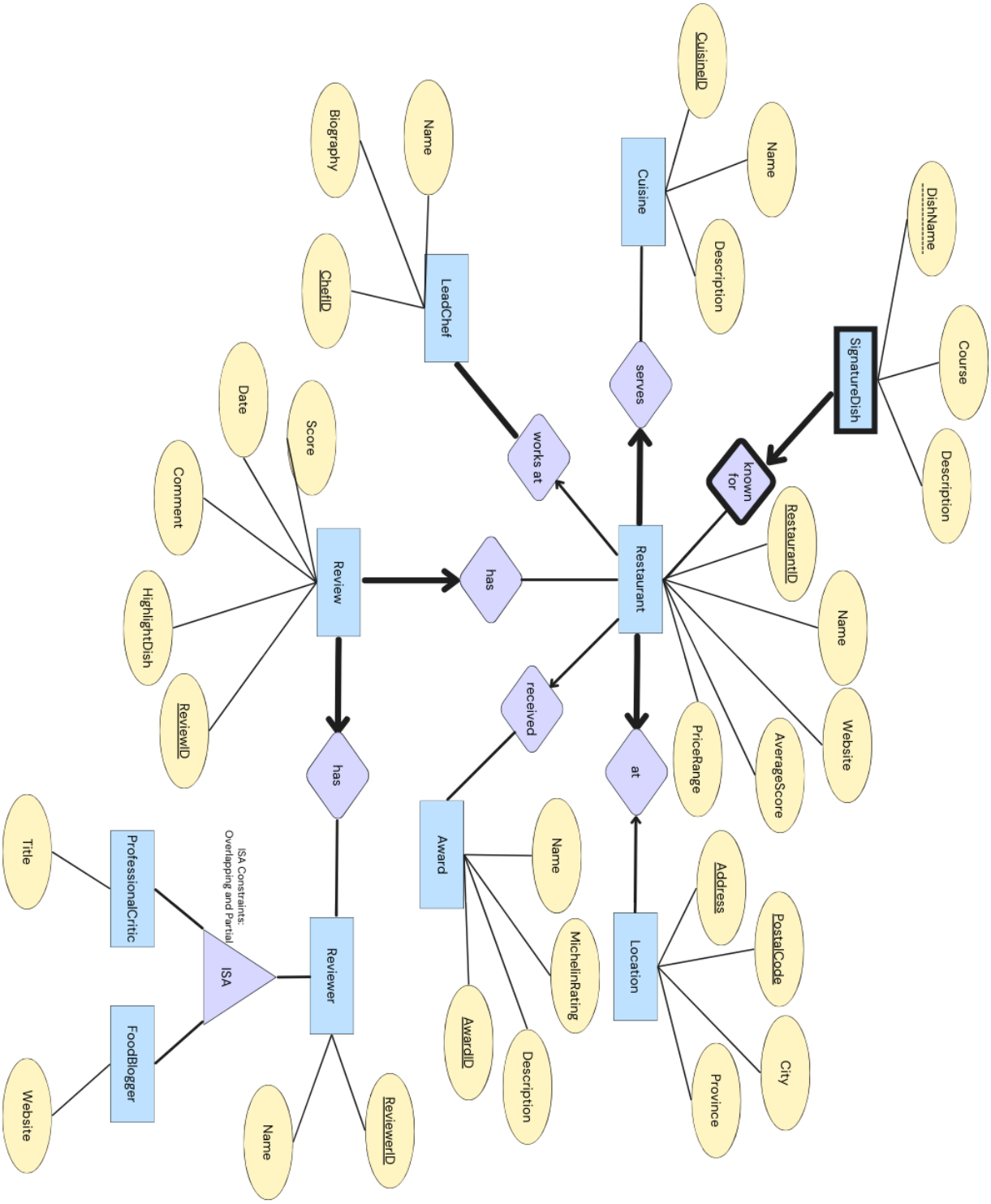
University of British Columbia, Vancouver
Department of Computer Science

Brief Project Summary (2-3 sentences)

Our project focuses on developing a platform for the culinary industry, particularly gourmet dining and restaurant management in Vancouver. It aims to include Michelin-rated restaurants with details on chefs, specialty dishes, and user reviews to facilitate informed dining decisions. The application allows users to write reviews, access others' reviews, explore restaurants based on various criteria, and discover local chefs, enhancing accessibility and engagement within the gourmet restaurant scene.

University of British Columbia, Vancouver
Department of Computer Science

ER Diagram



University of British Columbia, Vancouver
Department of Computer Science

Changes Made to ER Diagram

- Combine restaurant and works into a relation (many-to-one) WorksAtRestaurant
- Combine restaurant and serves into a relation (many-to-one) Serves
- Combine review and Has into a relation (many-to-one) HasReview
- Location to restaurant relationship is now one to one
- Remove ReviewerStatus
- Move Score from Reviewer to Review
- Add AverageScore attribute to Restaurant
- Change Received relationship to many to one, PK is RestaurantID
- Add relation for Serves
- Remove year from Award
    - The Michelin Guide offers current information rather than awarding restaurants awards every year
- SQL DDL -> website for restaurant could be a candidate key
- Add Name to Restaurant entity
- VARCHAR instead of CHAR
- Make website a candidate key for Restaurant
- Remove affiliation in reviewer
    - Redundant
- Add CuisineID to Cuisine
    - Name cannot be a primary key as it is not specific enough to determine the description
- Website is CK for WorksAtRestaurant

Schemas

Note: Relation names may differ from the ER Diagram where we combined entities or relationships to represent in a single relation.

Note: Primary keys are underlined, foreign keys are bolded, and *candidate keys* are italicized.

KnownForSignatureDish(DishName: VARCHAR, **RestaurantID**: INTEGER), Course: VARCHAR , Description: VARCHAR)

Cuisine(CuisineID: INTEGER, Name: VARCHAR, Description: VARCHAR)

LeadChef(ChefID: INTEGER, Name: VARCHAR, Biography: VARCHAR)

WorksAtRestaurant(RestaurantID: INTEGER, Name: VARCHAR, **ChefID**: INTEGER, *Website*: VARCHAR, PriceRange: VARCHAR, AverageScore: DECIMAL)

Serves(**RestaurantID**: INTEGER, **CuisineID**: INTEGER)

University of British Columbia, Vancouver
Department of Computer Science

HasReview(ReviewID: INTEGER, Date: DATE, Comment: VARCHAR, HighlightDish: VARCHAR, Score: INTEGER, **RestaurantID**: INTEGER)

ReviewerHas(ReviewerID: INTEGER, Name: VARCHAR, **ReviewID**: INTEGER)

ProfessionalCritic(**ReviewerID**: INTEGER, Title: VARCHAR)

FoodBlogger(**ReviewerID**: INTEGER, *Website*: VARCHAR)

Received(**RestaurantID**: INTEGER, AwardID: INTEGER)

Award(AwardID: INTEGER, Name: VARCHAR, MichelinRating: INTEGER, Description: VARCHAR)

Functional Dependencies

KnownForSignatureDish Table
- DishName → Course, Description, RestaurantID

Cuisine Table
- CuisineID → Name, Description
- Description → Name

LeadChef Table
- ChefID → RestaurantID, Name, Biography
- RestaurantID → ChefID

WorksAtRestaurant Table
- RestaurantID → Name, ChefID, Website, PriceRange, AverageScore
- Website → RestaurantID

Serves Table
- RestaurantID → CuisineID

HasReview Table
- ReviewID → Date, Comment, HighlightDish, Schore, RestaurantID

ReviewerHas Table
- ReviewerID → Name, ReviewID

ProfessionalCritic Table
- ReviewerID → Title

FoodBlogger Table

- ReviewerID → Website
- Website → ReviewerID

Received Table
- RestaurantID → AwardID

Award Table
- AwardID → Name, MichelinRating, Description

Reviewer Table
- ReviewerID → Score, Affiliation, Name

ProfessionalCritic Table
- ReviewerID → Title

FoodBlogger Table
- ReviewerID → Website

Received Table
- (RestaurantID, Year, Name) → Name

Normalization

Note: Primary keys are underlined, **foreign keys are bolded**, and *Candidate keys* are italicized.

KnownForSignatureDish Table
- R_1(DishName, Course, Description, **RestaurantID**)

Restaurant Table
- R_1(RestaurantId, Website, PriceRange, AverageScore, **CuisineName**, ***ChefID,*** **Location_Address**)

Cuisine Table
- R_1(**CuisineID**, Name, Description)

LeadChef Table
- R_1(ChefID, Name, Biography)

Review Table
- R_1(ReviewID, Date, Comment, HighlightDish, ReviewerStatus, Score, **RestaurantID**, **ReviewerID**)

University of British Columbia, Vancouver
Department of Computer Science

Reviewer Table
- R_1(<u>ReviewerID</u>, Name)

ProfessionalCritic Table
- R_1(<u>ReviewerID</u>, Title)

FoodBlogger Table
- R_1(<u>ReviewerID</u>, Website)

Received Table
- R_1(RestaurantID, Year, Name)

Award Table
- R_1(<u>AwardID,</u>Name, MichelinRating, Description, **RestaurantID,**)

ReviewerHasReview
- R_1(<u>ReviewID</u>, **ReviewerID**)

RestaurantHasReview
- R_1(**<u>ReviewID,</u> RestaurantID**)

University of British Columbia, Vancouver
Department of Computer Science

SQL DDL Statements

```
CREATE TABLE Location (
   Address VARCHAR NOT NULL,
   PostalCode VARCHAR NOT NULL,
   City VARCHAR NOT NULL,
   Province VARCHAR NOT NULL,
   PRIMARY KEY (Address)
);
CREATE TABLE Restaurant (
   RestaurantID INT AUTO_INCREMENT PRIMARY KEY,
   Name VARCHAR NOT NULL,
   Website VARCHAR NOT NULL,
   PriceRange VARCHAR,
   Location_Address VARCHAR NOT NULL,
   AverageScore DECIMAL,
   FOREIGN KEY (Location_Address) REFERENCES Location(Address) ON UPDATE CASCADE ON DELETE SET NULL
);
CREATE TABLE Cuisine (
   CuisineID INT AUTO_INCREMENT PRIMARY KEY,
   Name VARCHAR NOT NULL,
   Description VARCHAR
);
CREATE TABLE RestaurantServes (
   RestaurantID INT,
   CuisineID INT,
   FOREIGN KEY (RestaurantID) REFERENCES Restaurant(RestaurantID),
   FOREIGN KEY (CuisineID) REFERENCES Cuisine(CuisineID),
   PRIMARY KEY (RestaurantID, CuisineID)
);
CREATE TABLE LeadChef (
   ChefID INT AUTO_INCREMENT PRIMARY KEY,
   Name VARCHAR NOT NULL,
   Biography TEXT
);
CREATE TABLE WorksAtRestaurant (
   ChefID INT,
   RestaurantID INT,
   FOREIGN KEY (ChefID) REFERENCES LeadChef(ChefID),
   FOREIGN KEY (RestaurantID) REFERENCES Restaurant(RestaurantID),
   PRIMARY KEY (RestaurantID, ChefID)
);
CREATE TABLE SignatureDish (
   DishName VARCHAR NOT NULL,
   Description VARCHAR,
```

```
    Course VARCHAR,
    PRIMARY KEY (DishName),
);
CREATE TABLE KnownFor (
    DishName VARCHAR NOT NULL,
    RestaurantID INT,
    FOREIGN KEY (DishName) REFERENCES SignatureDish(DishName),
    FOREIGN KEY (RestaurantID) REFERENCES Restaurant(RestaurantID),
    PRIMARY KEY (DishName)
);
CREATE TABLE Award (
    AwardID INT AUTO_INCREMENT PRIMARY KEY,
    Name VARCHAR NOT NULL,
    MichelinRating INT,
    Description TEXT,
    RestaurantID INT,
    FOREIGN KEY (RestaurantID) REFERENCES Restaurant(RestaurantID) ON UPDATE CASCADE ON DELETE SET NULL
);
CREATE TABLE Reviewer (
    ReviewerID INT AUTO_INCREMENT PRIMARY KEY,
    Name VARCHAR NOT NULL
);
CREATE TABLE Review (
    ReviewID INT AUTO_INCREMENT PRIMARY KEY,
    Date DATE NOT NULL,
    Comment TEXT,
    Score DECIMAL,
    FOREIGN KEY (HighlightDish, RestaurantID) REFERENCES SignatureDish(DishName, RestaurantID) ON UPDATE
CASCADE ON DELETE SET NULL
);
CREATE TABLE RestaurantHasReview (
    ReviewID INT,
    RestaurantID INT,
    PRIMARY KEY (ReviewID),
    FOREIGN KEY (ReviewID) REFERENCES Review(ReviewID),
    FOREIGN KEY (RestaurantID) REFERENCES Restaurant(RestaurantID)
);
CREATE TABLE ReviewerHasReview (
    ReviewID INT,
    ReviewerID INT,
    PRIMARY KEY (ReviewID),
    FOREIGN KEY (ReviewID) REFERENCES Review(ReviewID),
    FOREIGN KEY (ReviewerID) REFERENCES Reviewer(ReviewerID)
);
```

University of British Columbia, Vancouver
Department of Computer Science

Statements to Populate Table

```
INSERT INTO KnownForSignatureDish
VALUES
  ('Shrimp-Crusted Striped Bass with XO sauce', 1, 'Main', 'Shrimp-crusted striped bass with XO sauce'),
  ('Panang Curry', 2, 'Main', 'Panang Curry'),
  ('Chinese Wine-Marinated Crispy Duck Tongues and Kidneys', 3, 'Main', 'Chinese Wine-Marinated Crispy Duck Tongues and Kidneys'),
  ('Jellyfish Strips Tossed in Garlic Ponzu', 4, 'Main', 'Jellyfish Strips Tossed in Garlic Ponzu'),
  ('Beef Short Rib with Tangy Gheymeh', 5, 'Main', 'Beef Short Rib with Tangy Gheymeh');

INSERT INTO Cuisine
VALUES
  (1, 'Contemporary', 'Modern Cuisine'),
  (2, 'Thai', 'Regional Cuisine'),
  (3, 'Chinese', 'Cantonese Cuisine'),
  (4, 'Japanese Contemporary', 'Sushi'),
  (5, 'Persian', 'Regional Cuisine');

INSERT INTO LeadChef
VALUES
  (1, 'Jack', 'Jack has been a Chef for 10 years and specializes in Contemporary cuisine.'),
  (2, 'Susie', 'Susie has been a Chef for 4 years and specializes in Thai cuisine.'),
  (3, 'Timothy', 'Timothy has been a Chef for 3 years and specializes in Chinese cuisine.'),
  (4, 'Jinny', 'Jinny has been a Chef for 4 years and specializes in Japanese cuisine.'),
  (5, 'Ken', 'Ken has been a Chef for 12 years and specializes in Persain cuisine.');

INSERT INTO WorksAtRestaurant
VALUES
  (1, 1, 'www.farmersapprentice.com', '$$'),
  (2, 2, 'www.maenam.com', '$$$'),
  (3, 3, 'www.chefschoicechinesecuisine.com', '$$$'),
  (4, 4, 'www.octopusgarden.com', '$$$$'),
  (5, 5, 'www.delara.com', '$$');

INSERT INTO Serves
VALUES
  (1, 1),
  (2, 2),
  (3, 3),
  (4, 4),
  (5, 5);

INSERT INTO HasReview
VALUES
```

```
  (1, 2000-01-01, 'Great Food!', 'Shrimp-Crusted Striped Bass with XO sauce', 4, 1),
  (2, 2000-01-01, 'Awesome', 'Panang Curry', 4, 2),
  (3, 2000-01-01, 'Yummy Dishes', 'Chinese Wine-Marinated Crispy Duck Tongues and Kidneys', 4, 3),
  (4, 2000-01-01, 'Delicious', 'Jellyfish Strips Tossed in Garlic Ponzu', 5, 4),
  (5, 2000-01-01, 'I love the food', 'Beef Short Rib with Tangy Gheymeh', 5, 5);

INSERT INTO ReviewerHas
VALUES
  (1, 'Yasmin', 1),
  (2, 'George', 2),
  (3, 'Edward', 3),
  (4, 'Steven', 4),
  (5, 'Tim',  5);

INSERT INTO ProfessionalCritic
VALUES
  (1, 'Instagam Blogger'),
  (2, 'Instagam Blogger'),
  (3, 'Youtube Blogger'),
  (4, 'Food Critic'),
  (5, 'Instagam Blogger');
INSERT INTO FoodBlogger
VALUES
  (1, 'www.tinasblog.com'),
  (2, 'www.foodreviews.com'),
  (3, 'www.jamreviewers.com'),
  (4, 'www.pensreview.com'),
  (5,  'www.fooddiary.com');

INSERT INTO Received
VALUES
  (1, 'Restaurant1'),
  (2, 'Restaurant2'),
  (3, 'Restaurant3'),
  (4, 'Restaurant4'),
  (5,  'Restaurant5');

INSERT INTO Award
VALUES
  (1, 'Michelin Star', 2, 'MichelinStar Rating 2'),
  (2,  'Michelin Star', 2, 'MichelinStar Rating 2'),
  (3,  'Michelin Star', 2, 'MichelinStar Rating 2'),
  (4,  'Bib Gourmand', null, 'Bib Gourmand'),
  (5,   'Michelin Star', 3, 'MichelinStar Rating 3');
```