**University of British Columbia, Vancouver**
Department of Computer Science

# CPSC 304 Project Cover Page

Milestone #: 2

Date: October 20, 2022

Group Number: 76

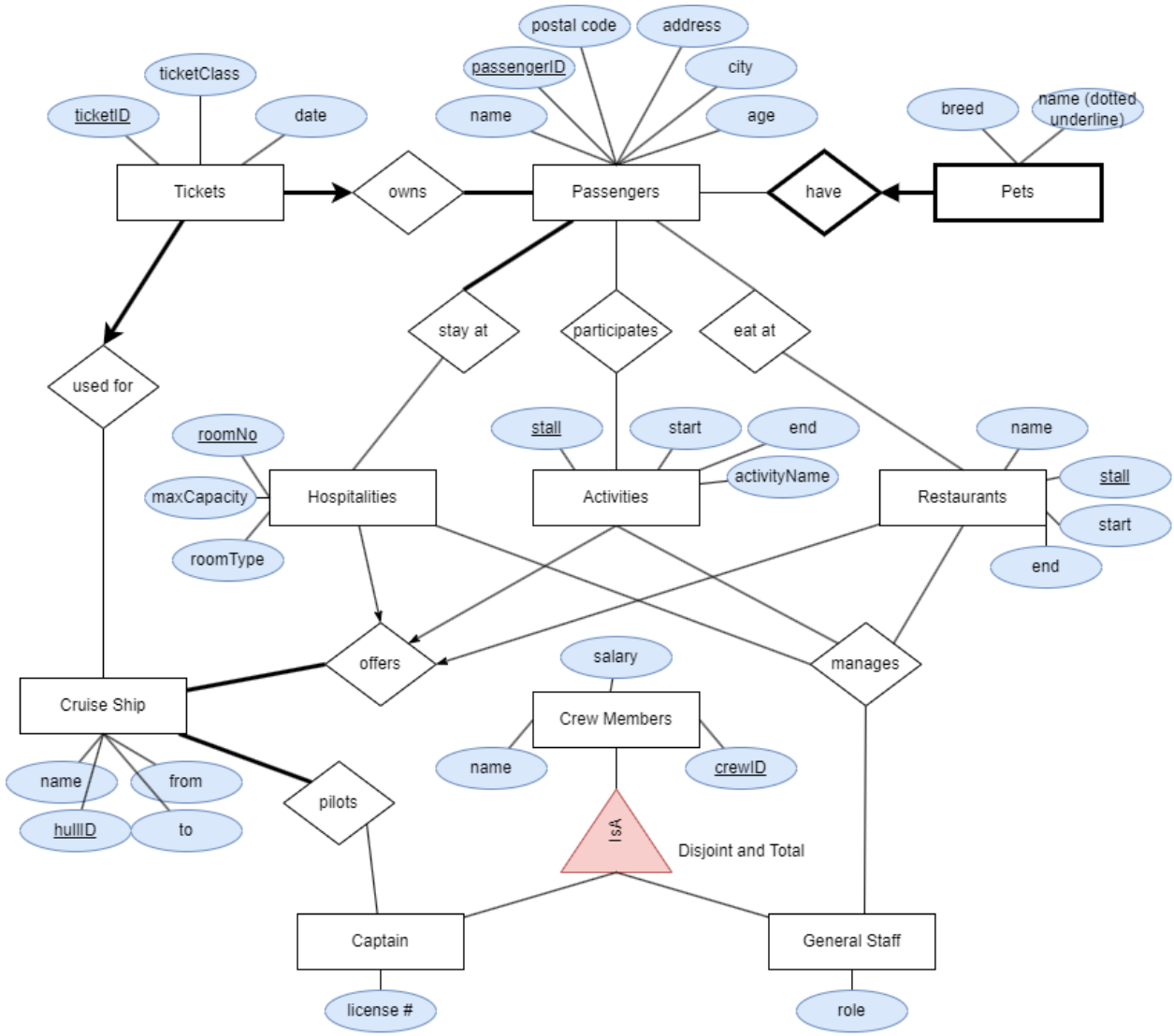
| Name | Student Number | CS Alias (Userid) | Preferred E-mail Address |
|---|---|---|---|
| Jason Nguyen | 11170081 | h1a3b | nguyen.j1305@gmail.com |
| Andy Lee | 13349634 | h2j4n | Andy2002221@gmail.com |
| West Liu | 40081531 | q9i4u | westliu20200101@gmail.com |

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

## *2. Changes made to ER Diagram:*

- Added attributes for Passenger (postal code, city, address) and CrewMembers (salary)
- Changed ‘hours’ in Activities and Restaurants to ‘start’ and ‘end’ time.
- Changed Hospitalities ‘numGuests’ to ‘maxCapacity’




Comments:

stall is where the amenity is located as well as the ship's id prefixed to it to make the amenity unique (e.g. alaska2-c3)

## *3. Schema a) & b)*

Underlines are Primary Keys
Bolds are Foreign Keys

Ticket(ticketID : number, ticketClass : string, date : string, **hullID** : number, **passengerID**: number)

- Constraints: hullID and passengerID are not null.

Passengers(passengerID : number, name: string, age: number, postalCode : string, address : string, city : string)

Pets(**passengerID**: number, petName: string, breed: string)

CruiseShip(hullID: number, name: string, from: string, to: string)

Hospitality(roomNo : string, maxCapacity: number, roomType: string, **hullID** : number)

PassengersStayAt(**passengerID** : number, **roomNo**: number)

Activities(stall : string, start : number, end : number, activityName: string, **hullID** : number)

PassengersParticipateIn(**stall** : string, **passengerID** : number)

Restaurants(stall : string, name : string, start: number, end : number, **hullID** : number)

PassengersEatAt(**passengerID** : number, **stall** : string)

Captain(crewID: number, name : string, salary: number, licenseNum: number)

Pilots(**crewID** : number, **hullID** : number)

GeneralStaff(crewID: number, name : string, role: string, salary : number)

ManageHospitalities(**crewID** : number, **roomNum** : number)

ManageActivities(**crewID** : number, **stall** : string)

ManageRestaurants(**crewID** : number, **stall** : string)

## *4. Functional Dependencies:*

Ticket(ticketID, ticketClass, date, hullID, passengerID)
1. ticketID -> ticketClass, date
2. ticketID -> hullID
3. ticketID -> passengerID

Passengers(passengerID, name, age, postalCode, address, city)
1. passengerID -> name, age, postalCode, address, city
2. postalCode -> city
3. address, city -> postalCode

Pets(passengerID, petName, breed)
1. passengerID, petName -> breed

CruiseShip(hullID, name, from, to)
1. hullID -> name, from, to

Hospitality(roomNo, maxCapacity, roomType, hullID)
1. roomNo -> maxCapacity, roomType
2. roomNo -> hullID

PassengersStayAt(passengerID, roomNo)
1. NONE

Activities(stall, start, end,  activityName, hullID)
1. stall -> start, end, activityName
2. stall -> hullID

PassengersParticipateIn(stall, passengerID)
1. NONE

Restaurants(stall, name, start, end, hullID)
1. stall -> name, start, end
2. stall -> hullID

PassengersEatAt(passengerID, stall)
1. NONE

Captain(crewID, name, salary, licenseNum)
1. crewID -> name, licenseNum

Pilots(crewID, hullID)
1. NONE

GeneralStaff(crewID, name, role, salary)
1. crewID -> name, role, salary
2. role -> salary

ManageHospitalities(crewID, roomNum)
1. NONE

ManageActivities(crewID, stall)

1. NONE

ManageRestaurants(crewID, stall)

1. NONE

## *5. Normalization:*

### Passenger Table Normalization:

Passenger(passengerID, name, age, postalCode, address, city)
Functional Dependencies:

1. passengerID -> name, age, postalCode, address, city
2. postalCode -> city
3. address, city -> postalCode

The key of the table Passenger is passengerID. The LHS of the second functional dependency is not a superkey, so we need to decompose on postalCode -> city.

PassengerLocation(postalCode, city), Passenger(passengerID, name, age, **postalCode,** address)

### GeneralStaff Table Normalization:

GeneralStaff(crewID, name, role, salary)

1. crewID -> name, role, salary
2. role -> salary

The key of the table GeneralStaff is crewID. The LHS of the second functional dependency is not a superkey, so we need to decompose on role -> salary.

GeneralStaffSalary(role, salary), GeneralStaff(crewID, name, **role**)

Every other table is already in BCNF so we don't need to normalize the rest.

## *Full list of tables post-normalization:*

Ticket(ticketID : number, ticketClass : string, date : string, **hullID** : number, **passengerID**: number)

PassengerLocation(postalCode : string, city : string)

Passengers(passengerID : number, name: string, age: number, postalCode : string, address : string)

Pets(**passengerID**: number, petName: string, breed: string)

CruiseShip(hullID: number, name: string, from: string, to: string)

Hospitality(roomNo : string, maxCapacity: number, roomType: string, **hullID** : number)

PassengersStayAt(**passengerID** : number, **roomNo**: number)

Activities(stall : string, start : number, end : number, activityName: string, **hullID** : number)

PassengersParticipateIn(**stall** : string, **passengerID** : number)

Restaurants(stall : string, name : string, start : number, end : number, **hullID** : number)

PassengersEatAt(**passengerID** : number, **stall** : string)

Captain(crewID: number, name : string, salary: number, licenseNum: number)

Pilots(**crewID** : number, **hullID** : number)

GeneralStaffSalary(role : string, salary : number)

GeneralStaff(crewID : number, name : string, **role** : string)

ManageHospitalities(**crewID** : number, **roomNum** : number)

ManageActivities(**crewID** : number, **stall** : string)

ManageRestaurants(**crewID** : number, **stall** : string)

## *6. SQL DDL Statements:*

```
CREATE TABLE Ticket (
        ticketID INTEGER PRIMARY KEY,
        ticketClass CHAR(20),
        date CHAR(6),
        hullID INTEGER NOT NULL,
        passengerID INTEGER NOT NULL,
        FOREIGN KEY (hullID) REFERENCES Cruise Ship
                ON DELETE NO ACTION
                ON UPDATE CASCADE
        FOREIGN KEY (passengerID) REFERENCES Passengers
                ON DELETE NO ACTION
                ON UPDATE CASCADE
)

CREATE TABLE Passengers (
        passengerID INTEGER PRIMARY KEY,
        name CHAR(20),
        age INTEGER,
        postalCode CHAR(6),
        address CHAR(20),
        FOREIGN KEY (postalCode) REFERENCES PassengerLocations
                ON DELETE NO ACTION
                ON UPDATE CASCADE
)

CREATE TABLE PassengerLocations (
        postalCode CHAR(6) PRIMARY KEY,
        city CHAR(20)
)

CREATE TABLE Pets (
        passengerID INTEGER,
        petName CHAR(20),
        breed CHAR(20),
        PRIMARY KEY (passengerID, petName)
        FOREIGN KEY (passengerID) REFERENCES Passengers,
                ON DELETE CASCADE
                ON UPDATE CASCADE
)

CREATE TABLE CruiseShip(
        hullID INTEGER PRIMARY KEY,
        name CHAR(20),
```

```
        from CHAR(20),
        to CHAR(20)
)

CREATE TABLE Hospitality(
        roomNo CHAR(20) PRIMARY KEY,
        maxCapacity INTEGER,
        roomType CHAR(20),
        hullID INTEGER,
        FOREIGN KEY (hullID) REFERENCES CruiseShip
                ON DELETE CASCADE
                ON UPDATE CASCADE
)

CREATE TABLE PassengersStayAt(
        passengerID INTEGER,
        roomNo INTEGER,
        PRIMARY KEY (passengerID, roomNo),
        FOREIGN KEY (passengerID) REFERENCES Passenger
                ON DELETE CASCADE
                ON UPDATE CASCADE
        FOREIGN KEY (roomNo) REFERENCES Hospitalities
                ON DELETE CASCADE
                ON UPDATE CASCADE
)

CREATE TABLE Activities (
        stall CHAR(20) PRIMARY KEY,
        start INTEGER,
        end INTEGER,
        activityName CHAR(20),
        hullID INTEGER,
        PRIMARY KEY (stall),
        FOREIGN KEY (hullID) REFERENCES CruiseShip
                ON DELETE CASCADE
                ON UPDATE CASCADE
)

CREATE TABLE PassengersParticipateIn(
        stall CHAR(20),
        passengerID INTEGER,
        PRIMARY KEY (stall, passengerID)
        FOREIGN KEY (stall) REFERENCES Activities
                ON DELETE CASCADE
                ON UPDATE CASCADE
        FOREIGN KEY (passengerID) REFERENCES Passenger
                ON DELETE CASCADE
                ON UPDATE CASCADE
```

```
)

CREATE TABLE Restaurants (
        stall CHAR(20) PRIMARY KEY,
        name CHAR(20),
        start INTEGER,
        end INTEGER,
        hullID INTEGER,
        FOREIGN KEY (hullID) REFERENCES CruiseShip
                ON DELETE CASCADE
                ON UPDATE CASCADE
)

CREATE TABLE PassengersEatAt(
        passengerID INTEGER,
        stall CHAR(20),
        PRIMARY KEY (passengerID, stall)
        FOREIGN KEY (passengerID) REFERENCES Passengers
                ON DELETE CASCADE
                ON UPDATE CASCADE
        FOREIGN KEY (stall) REFERENCES Restaurants
                ON DELETE CASCADE
                ON UPDATE CASCADE
)

CREATE TABLE Captain (
        crewID INTEGER PRIMARY KEY,
        name CHAR(20),
        salary INTEGER,
        licenseNum INTEGER,
)

CREATE TABLE Pilots (
        crewID INTEGER
        hullID INTEGER
        PRIMARY KEY (crewID, hullID)
        FOREIGN KEY (crewID) REFERENCES Captain
                ON DELETE NO ACTION
                ON UPDATE CASCADE
        FOREIGN KEY (hullID) REFERENCES CruiseShip
                ON DELETE NO ACTION
                ON UPDATE CASCADE
)

CREATE TABLE GeneralStaffSalary(
        role CHAR(20) PRIMARY KEY,
        salary INTEGER
```

```
)

CREATE TABLE GeneralStaff(
        crewID INTEGER PRIMARY KEY,
        name CHAR(20),
        role CHAR(20)
        FOREIGN KEY (role) REFERENCES GeneralStaffSalary
                ON DELETE NO ACTION
                ON UPDATE CASCADE
)

CREATE TABLE ManageHospitalities(
        crewID INTEGER,
        roomNum INTEGER,
        PRIMARY KEY (crewID, roomNum),
        FOREIGN KEY (crewID) REFERENCES GeneralStaff,
                ON DELETE NO ACTION
                ON UPDATE CASCADE
        FOREIGN KEY (roomNum) REFERENCES Hospitality
                ON DELETE NO ACTION
                ON UPDATE CASCADE
)

CREATE TABLE ManageActivities(
        crewID INTEGER,
        stall CHAR(20),
        PRIMARY KEY (crewID, stall),
        FOREIGN KEY (crewID) REFERENCES GeneralStaff,
                ON DELETE NO ACTION
                ON UPDATE CASCADE
        FOREIGN KEY (stall) REFERENCES Activities
                ON DELETE NO ACTION
                ON UPDATE CASCADE
)

CREATE TABLE ManageRestaurants(
        crewID INTEGER,
        stall CHAR(20),
        PRIMARY KEY (crewID, stall),
        FOREIGN KEY (crewID) REFERENCES GeneralStaff,
                ON DELETE NO ACTION
                ON UPDATE CASCADE
        FOREIGN KEY (stall) REFERENCES Restaurants
                ON DELETE NO ACTION
```

```
                ON UPDATE CASCADE
)
```

## *7. INSERT Statements:*

Ticket(ticketID : number, ticketClass : string, date : string, **hullID** : number, **passengerID**: number)

| ticketID | ticketClass | date | hullID | passengerID |
|---|---|---|---|---|
| 1 | “luxury” | 6/23/22 | 1 | 1 |
| 2 | “economy” | 6/23/22 | 1 | 2 |
| 3 | “suite” | 9/20/22 | 2 | 3 |
| 4 | “suite” | 9/20/22 | 2 | 4 |
| 5 | “economy” | 9/25/22 | 3 | 5 |
| 6 | “luxury” | 10/15/22 | 4 | 6 |
| 7 | “suite” | 10/21/22 | 5 | 7 |
| 8 | “suite” | 10/21/22 | 5 | 8 |

PassengerLocation(postalCode : string, city : string)

| postalCode | city |
|---|---|
| V3T | Surrey |
| V1M | Surrey |
| V3J | Burnaby |
| V3N | Burnaby |
| V5Z | Vancouver |

Passengers(passengerID : number, name: string, age: number, postalCode : string, address : string)

| passengerID | name | age | postalCode | address |
|---|---|---|---|---|
| 1 | Jason Smith | 21 | V3T | 31232 140 Street |
| 2 | John Legend | 35 | V1M | 87732 128 Street |
| 3 | Kim Jane | 28 | V3J | 68232 64 Avenue |
| 4 | Daniel Jane | 29 | V3J | 68232 64 Avenue |
| 5 | Alex Green | 33 | V5Z | 24521 120A Street |

| | | | | |
|---|---|---|---|---|
| 6 | Adrian Chun | 25 | V3N | 43262 80 Avenue |
| 7 | Ashley Campbell | 31 | V5Z | 14461 96 Street |
| 8 | David Campbell | 32 | V5Z | 14461 96 Street |

Pets(**passengerID**: number, petName: string, breed: string)

| passengerID | petName | breed |
|---|---|---|
| 1 | Bogey | Golden Retriever |
| 3 | Armpit | Pitbull |
| 5 | Steve | Siamese Cat |
| 6 | Bummy | Burmese Cat |
| 7 | Hammy | Syrian Hamster |

CruiseShip(hullID: number, name: string, from: string, to: string)

| hullID | name | from | to |
|---|---|---|---|
| 1 | Princess Cruises | Vancouver, BC | Vancouver Island, BC |
| 2 | Disney Cruise Line | Tsawwassen, BC | Tsawwassen, BC |
| 3 | Viking Cruises | Vancouver, BC | Vancouver, BC |
| 4 | Crystal Cruises | Vancouver, BC | Bowen Island, BC |
| 5 | Klondike Cruises | Nanaimo, BC | Vancouver, BC |

Hospitality(roomNo : string, maxCapacity: number, roomType: string, **hullID** : number)

| roomNo | maxCapacity | roomType | hullID |
|---|---|---|---|
| PC1-01 | 4 | “double kings” | 1 |
| PC1-02 | 2 | “double twin” | 1 |
| DCL2-01 | 2 | “single queen” | 2 |
| VC3-01 | 1 | “single twin” | 3 |
| CC4-01 | 4 | “double kings” | 4 |

| | | | |
|---|---|---|---|
| KC5-01 | 2 | “double queen” | 5 |

PassengersStayAt(**passengerID** : number, **roomNo**: number)

| passengerID | roomNo |
|---|---|
| 1 | PC1-01 |
| 2 | PC1-02 |
| 3 | DCL2-01 |
| 4 | DCL2-01 |
| 5 | VC3-01 |
| 6 | CC4-01 |
| 7 | KC5-01 |
| 8 | KC5-01 |

Activities(stall : string, start: number, end : number, activityName: string, **hullID** : number)

| stall | start | end | activityName | hullID |
|---|---|---|---|---|
| PC1-DANCE1 | 1200 | 1500 | “dancing” | 1 |
| PC1-DANCE2 | 1600 | 1800 | “dancing” | 1 |
| DCL2-DRAW1 | 1400 | 1900 | “painting” | 2 |
| DCL2-SWIM1 | 1200 | 2000 | “swimming” | 2 |
| VC3-SWIM1 | 1100 | 2100 | “swimming” | 3 |
| CC4-MNGLF1 | 1400 | 2300 | “mini golf” | 4 |
| KC5-MOVIE1 | 1800 | 2000 | “movies” | 5 |

PassengersParticipateIn(**stall** : string, **passengerID** : number)

| stall | passengerID |
|---|---|
| DCL2-SWIM1 | 3 |
| DCL2-SWIM1 | 4 |
| KC5-MOVIE1 | 7 |

| | |
|---|---|
| KC5-MOVIE1 | 8 |
| PC1-DANCE1 | 1 |

Restaurants(stall : string, name : string, start : number, end : number, **hullID** : number)

| **stall** | **name** | **start** | **end** | **hullID** |
|---|---|---|---|---|
| DCL2-MCD | McDonalds | 0000 | 2400 | 2 |
| KC5-KEG | The Keg | 1600 | 2400 | 5 |
| PC1-TACO | Tacofina | 0800 | 2200 | 1 |
| CC4-POKE | Pokaye | 1000 | 2100 | 4 |
| VC3-EARLS | Earls | 1200 | 2400 | 3 |

PassengersEatAt(**passengerID** : number, **stall** : string)

| **passengerID** | **stall** |
|---|---|
| 3 | DCL2-MCD |
| 4 | DCL2-MCD |
| 7 | KC5-KEG |
| 8 | KC5-KEG |
| 1 | PC1-TACO |

Captain(crewID: number, name : string, salary: number, licenseNum: number)

| **crewID** | **name** | **salary** | **licenseNum** |
|---|---|---|---|
| 3 | Masa | 120000 | 1278 |
| 4 | Brian | 160000 | 2834 |
| 5 | Alex | 170000 | 7263 |
| 6 | Mike | 200000 | 4028 |
| 7 | Chris | 110000 | 3948 |

Pilots(**crewID** : number, **hullID** : number)

| crewID | hullID |
|---|---|
| 3 | 1 |
| 4 | 2 |
| 5 | 3 |
| 6 | 4 |
| 7 | 5 |

GeneralStaffSalary(role : string, salary : number)

| role | salary |
|---|---|
| Housekeeping | $50,000 |
| Waiter | $55,000 |
| Cook | $70,000 |
| Activity Manager | $52,000 |
| Janitor | $60,000 |

GeneralStaff(crewID : number, name : string, **role** : string)

| crewID | name | role |
|---|---|---|
| 10 | Jason | Housekeeping |
| 11 | Andy | Waiter |
| 12 | West | Cook |
| 13 | Janel | Activity Manager |
| 14 | Ariana | Housekeeping |
| 15 | Kawhi | Janitor |
| 16 | Jinmin | Housekeeping |
| 17 | Elon | Activity Manager |

ManageHospitalities(**crewID** : number, **roomNum** : number)

| crewID | roomNum |
|---|---|
| 10 | PC1-01 |
| 10 | PC1-02 |
| 11 | PC1-03 |
| 11 | PC1-04 |
| 12 | PC1-05 |

ManageActivities(**crewID** : number, **stall** : string)

| crewID | stall |
|---|---|
| 13 | PC1-DANCE1 |
| 17 | PC1-DANCE2 |
| 15 | DCL2-DRAW1 |
| 15 | VC3-SWIM1 |
| 17 | KC5-MOVIE1 |

ManageRestaurants(**crewID** : number, **stall** : string)

| crewID | stall |
|---|---|
| 11 | DCL2-McD |
| 11 | KC5-KEG |
| 12 | PC1-TACO |
| 12 | CC4-POKE |
| 12 | VC3-EARLS |