University of British Columbia, Department of Computer Science

CPSC 304

Cover Page for Project Part 2

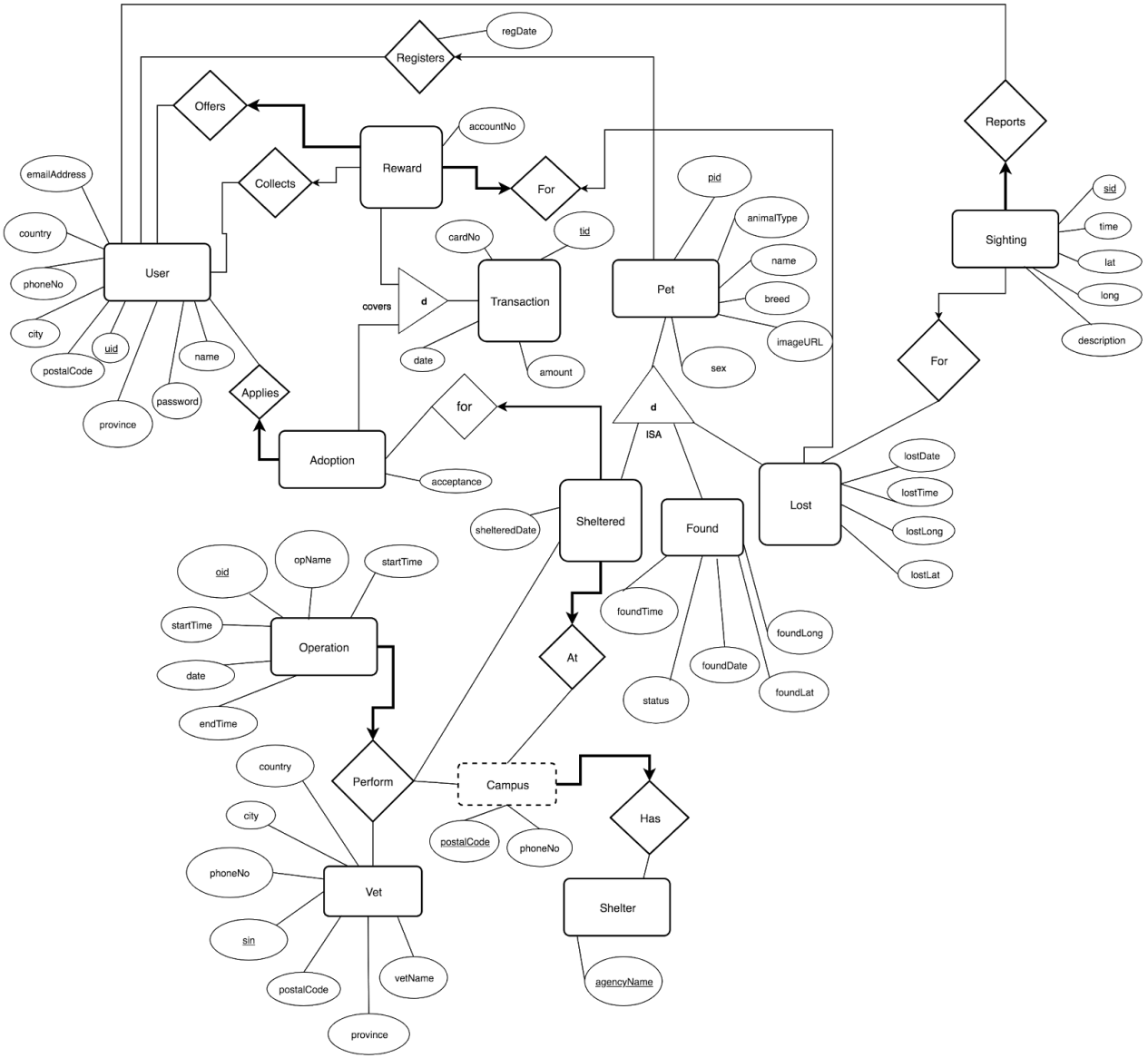
Date: October 14, 2018

Group Members:

Name	Student Number	CS Userid	Tutorial Section	Email Address
Samuel Or	45629300	u4h1b	T1B	or.samuel1@gmail.com
Harry Tao	31267157	k6o0b	T1G	htao1997@hotmail.com
Dante Spadinger-Fengler	14381157	t5z0b	T1C	dantefengler@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 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

ER Diagram


regDate
Registers
Offers
accountNo
Reward
Collects
For
emailAddress
country
phoneNo
User
city
uid
postalCode
name
password
province
Applies
covers
d
cardNo
tid
Transaction
date
amount
pid
animalType
name
breed
imageURL
Pet
sex
Reports
Sighting
sid
time
lat
long
description
For
d
ISA
for
Adoption
acceptance
Sheltered
shelteredDate
Found
Lost
lostDate
lostTime
lostLong
lostLat
foundTime
foundLong
foundDate
foundLat
status
At
oid
opName
startTime
startTime
Operation
date
endTime
Perform
country
city
phoneNo
Vet
sin
postalCode
vetName
province
Campus
postalCode
phoneNo
Has
Shelter
agencyName

Schema Definitions and Functional Dependencies

Bolded attributes denote foreign keys; underlined attributes denote primary keys.

#	Schema	Functional Dependencies	BCNF?	3NF?	Notes
1	Pet(pid: string, **uid**: string**,** name: string, sex: string, birthDate: YYYY-MM-DD, regDate: YYYY-MM-DD, animalType: string, breed: string, imageURL: string) **Foreign keys:** ● uid references User	1. pid -> **uid,** birthDate, sex, regDate, animalType, breed, imageURL 2. breed -> animalType	No	No	'breed' is not a primary key and does not belong to any candidate keys.
2	Lost**(pid**: string, lostDate: YYYY-MM-DD, lostTime: hh:mm:ss, lostLat: float, lostLong: float) **Foreign keys:** ● pid references Pet	1. **pid** -> lostDate, lostTime, lostLat, lostLong	Yes[1]	Yes[1]	
3	Found**(pid** : string, foundDate: YYYY-MM-DD, foundLat: float, foundLong: float, foundTime: hh:mm:ss, status: string[2]) **Foreign keys:** ● pid references Pet	1. **pid** -> foundDate, foundLat, foundLong, foundTime, status	Yes[1]	Yes[1]	
4	Sheltered**(pid**: string, **agencyName**: string**, postalCode**: string **tid**: string**,** shelteredDate: YYYY-MM-DD) **Foreign keys:**	1. **pid** -> **agencyName, postalCode, tid,** shelteredDate	Yes[1]	Yes[1]	

[1]For these schema, all FDs have a superkey on the left-hand side, so they are in BCNF and 3NF.
[2] Status is either 'deceased' or 'returned'.

	• pid references Pet • tid references Adoption • (agencyName, postalCode) references Campus (not null)				
5	Sighting(<u>sid</u>: string, date: YYYY-MM-DD, time: hh:mm:ss, lat: float, long: float, **pid**: string**, uid:** string**,** desc: string) **Foreign keys:** 1. pid references Pet 2. uid references User (not null)	1. <u>sid</u> -> date, time, lat, long, pid, uid, desc 2. (**uid**, date, time) -> sid, lat, long, **pid**, desc	Yes[1]	Yes[1]	**Superkeys:** • (**uid,** date, time)
6	Reward(<u>tid</u>, cardNo, amount, date: YYYY-MM-DD, accountNo: integer, **uid-collect:** string**, pid:** string) **Foreign keys:** 1. uid-collect references User (not null) 2. pid references Pet (not null)	1. <u>tid</u> -> cardNo, amount, accountNo, **uid-collect, pid** 2. (**uid-collect, pid,** date**)** -> <u>tid</u>, cardNo, amount, accountNo	Yes[1]	Yes[1]	**Superkeys:** • (**uid-collect, pid**, date) The 'uid' corresponding to 'offers' can be uniquely identified by the entry in Pet pointed to by 'pid'.
7	User(<u>uid: s</u>tring, name: string, password: string, emailAddress: string, phoneNo: integer, city: string, province: string, country: string, postalCode: string)	1. <u>uid</u> -> name, phoneNo, emailAddress, password, province, city, country, postalCode 2. postalCode -> city, province, country 3. phoneNo -> city, province, country	No	No	'postalCode' and 'phoneNumber' are not primary keys and do not belong to any candidate keys.
8	Vet(<u>sin: integer</u>, vetName: string, postalCode: string, phoneNo: integer, emailAddress: string, city: string, province: string, country: string)	1. <u>sin</u> -> vetName, postalCode, phoneNo, emailAddress, city, province, country 2. postalCode -> city, province, country 3. phoneNo -> city, province, country	No	No	'postalCode' and 'phoneNumber' are not primary keys and do not belong to any candidate keys.
9	Operation(<u>oid</u>: string, **vid** string**, pid:** string**,** opName: string, date: YYYY-MM-DD, startTime:	1. <u>oid</u> -> **vid, pid,** opName, date, startTime, endTime	Yes[1]	Yes[1]	**Superkeys:**

	hh:mm:ss, endTime: hh:mm:ss, **agencyName:** string**, postalCode**: string) **Foreign keys:** 1. pid references Pet (not null) 2. vid references Vet (not null) 3. (agencyName, postalCode) references Campus (not null)	2. (**vid,** date, startTime) -> oid, **pid,** opName, endTime, **agencyName, postalCode** 4. (**pid,** date, startTime) -> oid, **vid,** opName, endTime, **agencyName, postalCode** 5. (**vid,** date, endTime) -> oid, **pid,** opName, startTime, **agencyName, postalCode** 6. (**pid,** date, endTime) -> oid, **vid,** opName, startTime, **agencyName, postalCode**			• (**vid,** date, startTime) • (**pid,** date, startTime) • (**vid,** date, endTime) • (**pid,** date, endTime)
10	Adoption(tid**:** string, cardNo: integer, amount: float, date: YYYY-MM-DD, **uid**: string**, pid:** string acceptance: boolean) **Foreign keys:** 1. uid references User (not null) 2. pid references Sheltered	1. tid -> cardNo, amount, date, uid, pid acceptance	Yes[1]	Yes[1]	
11	Shelter(agencyName: string)		Yes[1]	Yes[1]	
12	Campus(postalCode: string, **agencyName:** string phoneNo: integer) **Partial identifier:** postalCode **Primary key: (**agencyName, postalCode) **Foreign keys:** 1. agencyName references Shelter	1. (**agencyName**, postalCode) -> phoneNo	Yes[1]	Yes[1]	

Normalization

#		Normalized Tables	Functional Dependencies	BCNF?	3NF?

1	1.1	Pet(pid: string, **uid**: string**,** name: string, sex: M or F, birthDate: YYYY-MM-DD, regDate: YYYY-MM-DD, imageURL: string)	1. pid -> **uid**, name, sex, birthDate, regDate, **breed**, imageURL	Yes	Yes
	1.2	Pet-Type(animalType: string, breed: string)	1. breed -> animalType	Yes	Yes
7	7.1	User(uid: string, name: string, password: string, emailAddress: string, phoneNo: integer, postalCode: string	1. uid -> name, phoneNo, password, postalCode, emailAddress	Yes	Yes
	7.2	Address-User-PC(postalCode: string, city: string, province: string, country: string)	1. postalCode -> city, province, country	Yes	Yes
	7.3	Address-User-PN(phoneNo: integer, city: string, province: string, country: string)	1. phoneNo -> city, province, country	Yes	Yes
8	8.1	Vet(sin: integer, vetName: string, postalCode: string, phoneNo: integer, emailAddress: string)	1. sin -> vetName, postalCode, phoneNo, emailAddress	Yes	Yes
	8.2	Address-Vet-PC(postalCode: string, city: string, province: string, country: string)	1. postalCode -> city, province, country	Yes	Yes
	8.3	Address-Vet-PN(phoneNo: integer, city: string, province: string, country: string)	1. phoneNo -> city, province, country	Yes	Yes

DDL

#		#	
1.1	`CREATE TABLE Pet`	1.2	`CREATE TABLE Pet-Type`

	(pid CHAR(16), uid CHAR(16), name CHAR(10), sex CHAR(1), birthDate DATE, regDate DATE, breed CHAR(20), imageURL CHAR(100), PRIMARY KEY (pid) FOREIGN KEY(uid) REFERENCES User ON DELETE NO ACTION);		(breed CHAR(20), animalType CHAR(16), PRIMARY KEY (breed));
2	CREATE TABLE Lost (pid CHAR(16), lostDate CHAR(6), lostTime CHAR(6), lostLat FLOAT, lostLong FLOAT, PRIMARY KEY (pid), FOREIGN KEY (pid) REFERENCES Pet);	3	CREATE TABLE Found (pid CHAR(16), uid CHAR(16), foundDate DATE, foundLat FLOAT, foundLong FLOAT, foundTime TIME, status CHAR(10), PRIMARY KEY(pid), FOREIGN KEY(uid) REFERENCES User ON DELETE NO ACTION ON UPDATE CASCADE);
4	CREATE TABLE Sheltered (pid CHAR(16), agencyName CHAR(20) NOT NULL, postalCode CHAR(6) NOT NULL,	5	CREATE TABLE Sighting (sid CHAR(16), time TIME, lat FLOAT,

	tid CHAR(16), shelteredDate DATE PRIMARY KEY(pid), FOREIGN KEY (tid) REFERENCES Adoption, FOREIGN KEY (agencyName, postalCode) REFERENCES Campus);		long FLOAT, pid CHAR(16), uid CHAR(16) NOT NULL, desc CHAR(256), PRIMARY KEY (sid) FOREIGN KEY (pid) REFERENCES Pet, FOREIGN KEY (uid) REFERENCES User ON DELETE NO ACTION ON UPDATE CASCADE);
6	CREATE TABLE Reward (tid CHAR(16), cardNo BIGINT(16), amount DECIMAL(19, 2), accountNo BIGINT(12), uid-collect CHAR(16), pid CHAR(16) NOT NULL, PRIMARY KEY (tid), FOREIGN KEY (uid-collect) REFERENCES User FOREIGN KEY(pid) REFERENCES Found ON DELETE NO ACTION ON UPDATE CASCADE);	7.1	CREATE TABLE User (uid CHAR(16), name CHAR(20), phoneNo BIGINT(11), password CHAR(20), postalCode CHAR(6), emailAddress CHAR(64), PRIMARY KEY (uid));
7.2	CREATE TABLE Address-User-PC (postalCode CHAR(6), city CHAR(20), province CHAR(20),	7.3	CREATE TABLE Address-User-PN (phoneNo CHAR(6), city CHAR(20), province CHAR(20),

	country CHAR(20), PRIMARY KEY (postalCode));		country CHAR(20), PRIMARY KEY (phoneNo));
8.1	CREATE TABLE Vet (sin INT(9), vetName CHAR(16), phoneNo BIGINT(11), postalCode CHAR(6), emailAddress city CHAR(20), province CHAR(20), country CHAR(20), PRIMARY KEY (vid));	8.2	CREATE TABLE Address-Vet (postalCode CHAR(6), city CHAR(20), province CHAR(20), country CHAR(20), PRIMARY KEY (postalCode));
8.3	CREATE TABLE Address-Vet-PN (phoneNo CHAR(6), city CHAR(20), province CHAR(20), country CHAR(20), PRIMARY KEY (phoneNo));		

9	CREATE TABLE Operation (oid CHAR(16), vid CHAR(16) NOT NULL, pid CHAR(16) NOT NULL, agencyName CHAR(20) NOT NULL, postalCode CHAR(6) NOT NULL, opName CHAR(16), date DATE, startTime TIME, endTime TIME, PRIMARY KEY (oid), FOREIGN KEY vid REFERENCES Vet(vid) ON DELETE NO ACTION ON UPDATE CASCADE, FOREIGN KEY pid REFERENCES Pet(pid) ON DELETE NO ACTION ON UPDATE CASCADE, FOREIGN KEY (agency,postalCode)REFERENCES Campus(pid) ON DELETE NO ACTION ON UPDATE CASCADE);	10	CREATE TABLE Adoption (tid CHAR(16), cardNo BIGINT(16), amount DECIMAL(19, 2), date DATE, uid CHAR(16), pid CHAR(16), acceptance BOOLEAN, PRIMARY KEY (tid), FOREIGN KEY (uid) REFERENCES User ON DELETE NO ACTION FOREIGN KEY (pid) REFERENCES Sheltered ON DELETE NO ACTION);
11	CREATE TABLE Shelter (agencyName CHAR(16), PRIMARY KEY (agencyName));	12	CREATE TABLE Campus (postalCode CHAR(6), agencyName CHAR(16), phoneNo INT(20) PRIMARY KEY (agencyName, postalCode), FOREIGN KEY (agencyName) REFERENCES Shelter ON DELETE CASCADE);