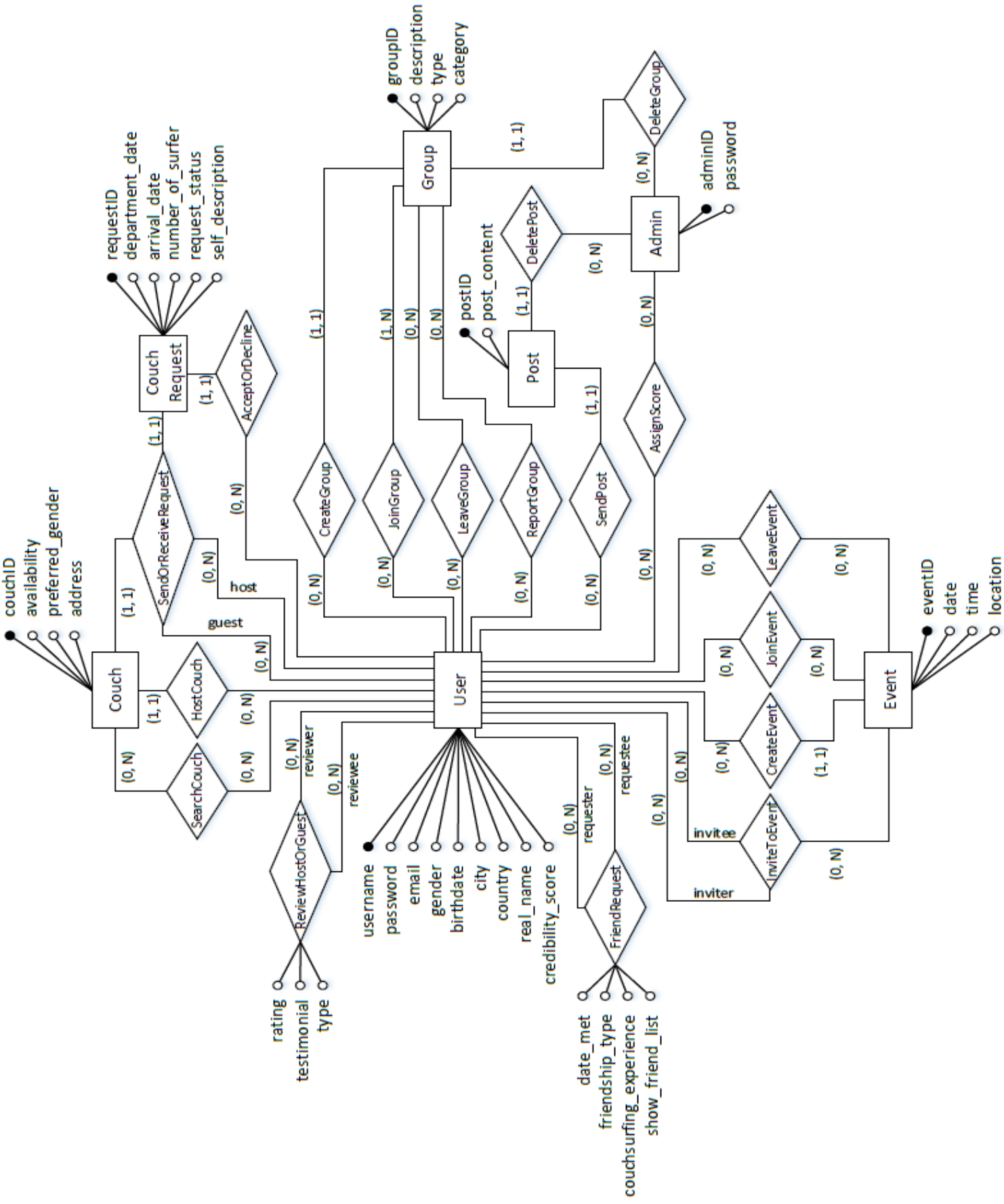
CSC343H1 S Assignment 3

Team of two
c4wangha, Han Wang, 998001638
g2pooniv, Iven Poon, 999169607
Zero grace days used, two grace days remaining

Part A: Assumptions

1. All guests and hosts are users
2. A person can register multiple user accounts
3. All couch addresses are distinct
4. Any user (guest or host) can make a review during/after the stay
5. There are two types of reviews: 1) a review written by a guest on a host 2) a review written by a host on a guest
6. Every user has a distinct username
7. A user can have multiple hosting locations
8. Replying to a post is equivalent to sending a post because in order to reply, you will be sending a post
9. Each couch request must be accepted or declined 30 days before the arrival date. Else, it would be automatically declined.
10. While a couch request status is not accepted, declined or maybe, it will be pending.
11. A group or event can only be created by one user
12. A creator of a group or event automatically joins the group or event

Part B: ERD


couchID
availability
preferred_gender
address
Couch
(1, 1)
(0, N)
HostCouch
SearchCouch
(1, 1)
(0, N)
(0, N)
(0, N)
SendOrReceiveRequest
(1, 1)
(0, N)
host
guest
(0, N)
requestID
department_date
arrival_date
number_of_surfer
request_status
self_description
Couch Request
(1, 1)
AcceptOrDecline
(0, N)
groupID
description
type
category
Group
(1, 1)
CreateGroup
(1, 1)
(0, N)
JoinGroup
(1, N)
(0, N)
LeaveGroup
(0, N)
(0, N)
ReportGroup
(0, N)
(0, N)
SendPost
(0, N)
(1, 1)
postID
post_content
Post
(1, 1)
DeletePost
(0, N)
DeleteGroup
(0, N)
adminID
password
Admin
(0, N)
AssignScore
(0, N)
(0, N)
LeaveEvent
(0, N)
(0, N)
JoinEvent
(0, N)
(0, N)
CreateEvent
(1, 1)
(0, N)
InviteToEvent
(0, N)
eventID
date
time
location
Event
User
username
password
email
gender
birthdate
city
country
real_name
credibility_score
(0, N)
reviewer
(0, N)
reviewee
ReviewHostOrGuest
rating
testimonial
type
(0, N)
requester
(0, N)
requestee
FriendRequest
date_met
friendship_type
couchsurfing_experience
show_friend_list
(0, N)
invitee
inviter

Part C: Relational Schema

Entities:

User (username, password, email, gender, birthdate, city, country, real_name, credibility_score);

Event (eventID, date, time, location);

Couch (couchID, availability, preferred_gender, address);

Group (groupID, description, type, category);

Post (postID, post_content);

Admin (adminID, password);

User/Event Relations:

CreateEvent (eventID, username);

JoinEvent (username, eventID);

LeaveEvent (username, eventID);

InviteToEvent (inviter, invitee, eventID);

User/Couch/CouchRequest Relations:

SearchCouch (username, couchID);

HostCouch (couchID, username);

SendOrRecieveRequest (couchID, requestID, guest, host);

AcceptOrDecline (requestID, username);

User/Post/Group Relations:

CreateGroup (groupID, username);

JoinGroup (username, groupID);

LeaveGroup (username, groupID);

ReportGroup (username, groupID);

SendPost (postID, username);

Admin/User/Post/Group Relations:

AssignScore (adminID, username);

DeletePost (postID, adminID);

DeleteGroup (groupID, adminID);

User/User Relations:

ReviewHostOrGuest (reviewer, reviewee, rating, testimonial, type);

FriendRequest (requester, requestee, date_met, friendship_type, couchsurfing_experience, show_friend_list);

Part D: PostgreSQL Definition

Entities:

```
CREATE TABLE User (
   username              VARCHAR(20)   PRIMARY KEY,
   password              VARCHAR(20)   NOT NULL,
   email                 VARCHAR(40)   NOT NULL,
   gender                CHAR          NOT NULL,
   birthdate             VARCHAR(20)   NOT NULL,
   city                  VARCHAR(20)   NOT NULL,
   country               VARCHAR(20)   NOT NULL,
   real_name             VARCHAR(20)   NOT NULL,
   credibility_score     INTEGER       NOT NULL);

CREATE TABLE Event (
   eventID               INTEGER       PRIMARY KEY,
   date                  VARCHAR(20)   NOT NULL,
   time                  VARCHAR(20)   NOT NULL,
   location              VARCHAR(40)   NOT NULL);

CREATE TABLE Couch (
   couchID               INTEGER       PRIMARY KEY,
   availability          CHAR          NOT NULL,
   preferred_gender      CHAR          NOT NULL,
   address               VARCHAR(40)   NOT NULL);
```

```
CREATE TABLE Group (
   groupID              INTEGER         PRIMARY KEY,
   description          VARCHAR(40)     NOT NULL,
   type                 CHAR            NOT NULL,
   category             VARCHAR(20)     NOT NULL);

CREATE TABLE Post (
   postID               INTEGER         PRIMARY KEY,
   post_content         VARCHAR(100)    NOT NULL);

CREATE TABLE Admin (
   adminID              VARCHAR(20)     PRIMARY KEY,
   password             VARCHAR(20)     NOT NULL);
```

User/Event Relations:

```
CREATE TABLE CreateEvent (
  // eventID is PRIMARY KEY
   eventID              INTEGER         REFERENCES Event(eventID) ON DELETE
RESTRICT,
   username             VARCHAR(20)     REFERENCES User(username) ON DELETE
RESTRICT);

CREATE TABLE JoinEvent (
  // username and eventID is PRIMARY KEY
   username             VARCHAR(20)     REFERENCES User(username) ON DELETE
RESTRICT,
   eventID              INTEGER         REFERENCES Event(eventID) ON DELETE
RESTRICT);

CREATE TABLE LeaveEvent (
// username and eventID is PRIMARY KEY
   username             VARCHAR(20)     REFERENCES User(username) ON DELETE
RESTRICT,
   eventID              INTEGER         REFERENCES Event(eventID) ON DELETE
RESTRICT);

CREATE TABLE InviteToEvent (
// inviter, invitee and eventID is PRIMARY KEY
   inviter              VARCHAR(20)     REFERENCES User(username) ON DELETE
RESTRICT,
   invitee              VARCHAR(20)     REFERENCES User(username) ON DELETE
RESTRICT,
   eventID              INTEGER         REFERENCES Event(eventID) ON DELETE
RESTRICT);
```

User/Couch/CouchRequest Relations:

```
CREATE TABLE SearchCouch (
// username and couchID is PRIMARY KEY
   username                VARCHAR(20)   REFERENCES User(username) ON DELETE
RESTRICT,
   couchID                 INTEGER       REFERENCES Couch(couchID) ON DELETE
RESTRICT);

CREATE TABLE HostCouch (
// couchID is PRIMARY KEY
  couchID                  INTEGER       REFERENCES User(username) ON DELETE
RESTRICT,
   username                VARCHAR(20)   REFERENCES User(username) ON DELETE
RESTRICT);

CREATE TABLE SendOrRecieveRequest (
// couchID and requestID is PRIMARY KEY
  couchID                  INTEGER       REFERENCES Couch(couchID) ON DELETE
RESTRICT,
   requestID               INTEGER       REFERENCES CouchRequest(requestID) ON
DELETE RESTRICT,
   guest                   VARCHAR(20)   REFERENCES User(username) ON DELETE
RESTRICT,
   host                    VARCHAR(20)   REFERENCES User(username) ON DELETE
RESTRICT);

CREATE TABLE AcceptOrDecline (
// requestID is PRIMARY KEY
  requestID                INTEGER       REFERENCES CouchRequest(requestID) ON
DELETE RESTRICT,
   username                VARCHAR(20)   REFERENCES User(username) ON DELETE
RESTRICT);
```

User/Post/Group Relations:

```
CREATE TABLE CreateGroup (
// groupID is PRIMARY KEY
  groupID                  INTEGER       REFERENCES Group(groupID) ON DELETE
RESTRICT,
   username                VARCHAR(20)   REFERENCES User(username) ON DELETE
RESTRICT);

CREATE TABLE JoinGroup (
// username and groupID is PRIMARY KEY
   username                VARCHAR(20)   REFERENCES User(username) ON DELETE
RESTRICT,
   groupID                 INTEGER       REFERENCES Group(groupID) ON DELETE
RESTRICT);
```

```
CREATE TABLE LeaveGroup (
// username and groupID is PRIMARY KEY
   username                VARCHAR(20)   REFERENCES User(username) ON DELETE
RESTRICT,
   groupID                 INTEGER       REFERENCES Group(groupID) ON DELETE
RESTRICT);

CREATE TABLE ReportGroup (
// username and groupID is PRIMARY KEY
   username                VARCHAR(20)   REFERENCES User(username) ON DELETE
RESTRICT,
   groupID                 INTEGER       REFERENCES Group(groupID) ON DELETE
RESTRICT);

CREATE TABLE SendPost (
// postID is PRIMARY KEY
   postID                  INTEGER       REFERENCES Post(postID) ON DELETE
RESTRICT,
   username                VARCHAR(20)   REFERENCES User(username) ON DELETE
RESTRICT);
```

Admin/User/Post/Group Relations:

```
CREATE TABLE AssignScore (
// adminID and username is PRIMARY KEY
   adminID                 INTEGER       REFERENCES Admin(adminID) ON DELETE
RESTRICT,
   username                VARCHAR(20)   REFERENCES User(username) ON DELETE
RESTRICT,
   score                   INTEGER       NOT NULL);

CREATE TABLE DeletePost (
// postID is PRIMARY KEY
  postID                   INTEGER       REFERENCES Post(postID) ON DELETE
RESTRICT,
   adminID                 INTEGER       REFERENCES Admin(adminID) ON DELETE
RESTRICT);

CREATE TABLE DeleteGroup (
// groupID is PRIMARY KEY
  groupID                  INTEGER       REFERENCES Group(groupID) ON DELETE
RESTRICT,
   adminID                 INTEGER       REFERENCES Admin(adminID) ON DELETE
RESTRICT);
```

User/User Relations:

```
CREATE TABLE ReviewHostOrGuest (
// reviewer and reviewee is PRIMARY KEY
   reviewer              VARCHAR(20)   REFERENCES User(username) ON DELETE
RESTRICT,
   reviewee              VARCHAR(20)   REFERENCES User(username) ON DELETE
RESTRICT,
   rating                INTEGER       NOT NULL,
   testimonial           VARCHAR(100)  NOT NULL,
   type                  CHAR          NOT NULL);

CREATE TABLE FriendRequest (
// requester and requestee is PRIMARY KEY
   requester             VARCHAR(20)   REFERENCES User(username) ON DELETE
RESTRICT,
   requestee             VARCHAR(20)   REFERENCES User(username) ON DELETE
RESTRICT,
   date_met              VARCHAR(20)   NOT NULL,
   friendship_type       CHAR          NOT NULL,
   couchsurfing_experience VARCHAR(100) NOT NULL,
   show_friend_list      BOOLEAN       NOT NULL);
```