

## Assignment 3

---

**Dongyao He 215917610**

**CSE:dhe16**

**Email:**[dhe16@my.yorku.ca](mailto:dhe16@my.yorku.ca)

**Instructor: Manos Papagelis**

**Date:11/29/2019**

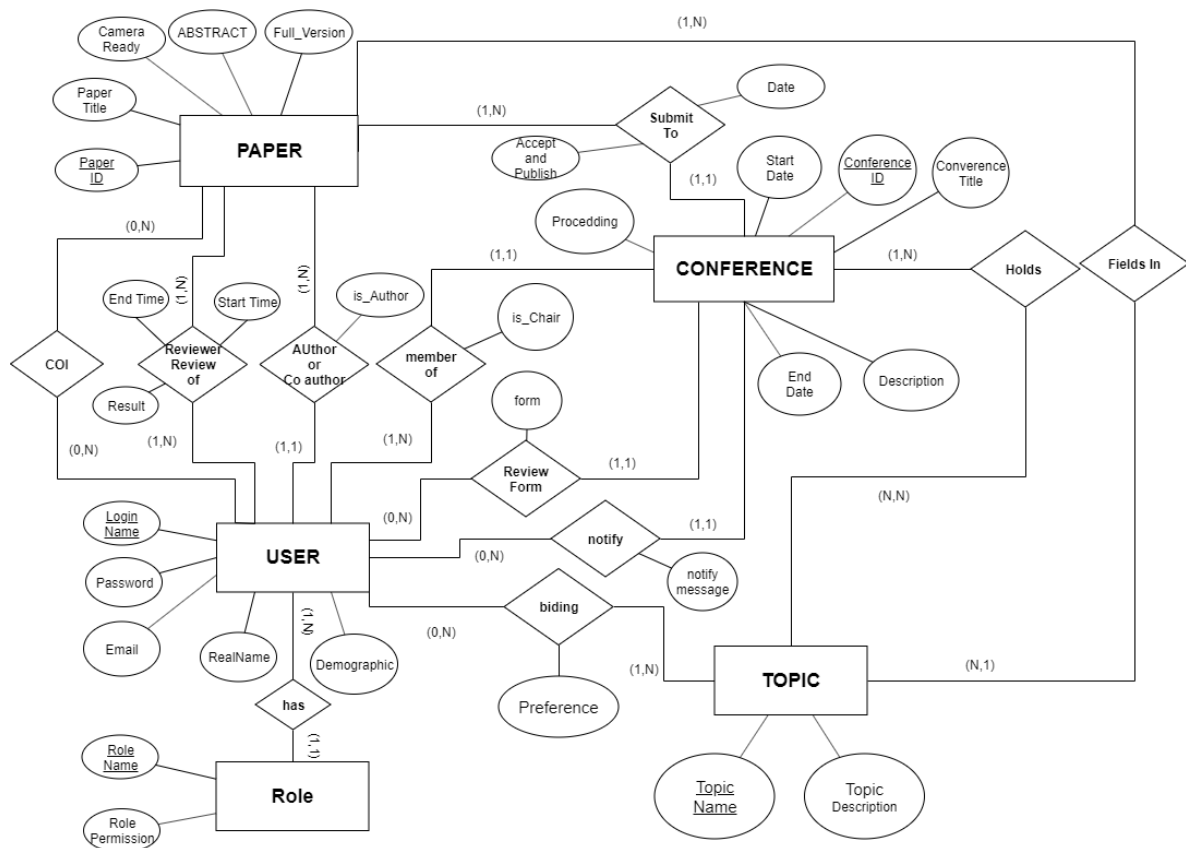
# Part1

---

## A detailed description of the assumptions you made about the system's functionality

1. The system store all the members as an entity called **USER** , user should have entity called **Role** which relate to **USER** which categorized 3 kinds of person which contain different duties and permissions each has relation refer to person, the system will exam the permission and regulate which action each user can perform
2. we assumed if an member can access to their user profile and make submission to the conference, they already have their account and logged in
3. there should be an entity called **conference** which only accept setup from user if user role is a PC chair, also **PAPER** can allowed submit to the conference and conference , the submit to relation will also has accept/decline relation to the **PAPER** entity and also store the date
4. for **conference** setup there should be object to store the property of conference setting save into directory of the system
5. for **USER** ,one user can represent in several PC\_CHAIR in different conferences and User role name must be a reviewer of the conference to get a review form
6. for **PAPER**, Abstract , Full\_version,Camera ready are all saved in system as a URL link to the a remote server.
7. For **USER** which role as reviewers ,they should all assigned by PC\_Chair or algorithm to specific **PAPER**.
8. The **member\_of** relation and is\_chair attribute refer to if the member of conference is a chair(True) or a member/reviewer(False)
9. **USER** can have preferences on topic as options of like/dislike/ignore
10. the system will collect and send notification to author when decision is made, as the notification work as a queue.
11. the system will store the result of submit into accept and publish in relation submit to as a BOOLEAN

## Part2



## Part3

---

Note: Bold for foreign key , underline for primary key

Conference (ConferenceID ,ConferenceTitle,Startdate,EndDate,Discription,Procedding)

Holds\_Topic(ConferenceID ,TopicName)

Role(RoleName,RolePermission)

Has\_Role(RoleName,LoginName)

User(LoginName,password,Email,RealName,Demographic)

Author\_or\_Co\_author(PaperID,LoginName,is\_Author)

Reviewer\_Review\_To(PaperID,LoginName,EndTime,StartTime,Result)

Paper(PaperId,PaperTitle,Abstract,CameraReady,Full\_Version)

COI(PaperID,LoginName)

Paper\_Fields(PaperId,TopicName)

Topic(TopicName,TopicDescription)

Submit\_To(SubmitID,**PaperID**,**ConferenceID**,Accept\_and\_Publish,Date)

User\_Member\_OF(LoginName,ConferenceID,is\_Chair)

User\_Biding(LoginName,TopicName,Prefrence)

Review\_Form(LoginName,ConferenceID,form)

Notify(LoginName,ConferenceID,notify\_message)

## Part4

---

```
1
2 CREATE TABLE Paper(
3     PaperId      INTEGER PRIMARY KEY,
4     PaperTitle   varchar not null,
5     Abstract     varchar not null,
6     CameraReady  varchar not null,
7     Full_Version varchar not null
8 );
9
10 CREATE TABLE Topic(
11     TopicName varchar PRIMARY KEY ,
12     TopicDescription varchar not null
13 );
```

```

14
15 CREATE TABLE CONFERENCE(
16     ConferenceID INTEGER PRIMARY KEY,
17     ConferenceTitle VARCHAR NOT NULL,
18     Startdate date not null ,
19     EndDate date not null ,
20     Discription varchar NOT NULL ,
21     Proceeding varchar NOT NULL
22 );
23
24 CREATE TABLE Holds_Topic(
25     ConferenceID INTEGER REFERENCES CONFERENCE(ConferenceID) ON DELETE
    RESTRICT ,
26     TopicName varchar REFERENCES Topic(TopicName) ON DELETE RESTRICT ,
27     PRIMARY KEY(ConferenceID,TopicName)
28 );
29
30 CREATE TABLE Role(
31     RoleName varchar PRIMARY KEY ,
32     RolePermission INTEGER NOT NULL
33 );
34
35 CREATE TABLE Conference_User(
36     LoginName varchar PRIMARY KEY ,
37     password varchar not null ,
38     Email varchar not null ,
39     RealName varchar not null ,
40     Demographic varchar not null
41 );
42
43 CREATE TABLE Has_Role(
44     RoleName varchar REFERENCES role(RoleName) ON DELETE RESTRICT ,
45     LoginName varchar REFERENCES Conference_User(LoginName) ON DELETE
    RESTRICT ,
46     PRIMARY KEY (RoleName,LoginName)
47 );
48
49 CREATE TABLE Author_or_Co_author(
50     PaperId INTEGER REFERENCES Paper(PaperId) ON DELETE RESTRICT,
51     LoginName varchar REFERENCES Conference_User(LoginName) ON DELETE
    RESTRICT ,
52     is_Author BOOLEAN not null,
53     PRIMARY KEY (PaperId,LoginName)
54
55 );
56
57 CREATE TABLE Reviewer_Review_To(
58     PaperID INTEGER REFERENCES Paper(PaperId),
59     LoginName varchar REFERENCES Conference_User(LoginName),
60     EndTime date not null ,
61     StartTime date not null ,
62     Result boolean not null
63 );
64
65 CREATE TABLE COI(
66     PaperId INTEGER REFERENCES Paper(PaperId) ON DELETE RESTRICT,
67     LoginName varchar REFERENCES Conference_User(LoginName) ON DELETE
    RESTRICT ,

```

```

68     PRIMARY KEY (PaperId,LoginName)
69 );
70
71 CREATE TABLE Paper_Fields(
72     PaperId INTEGER REFERENCES Paper(PaperId) ON DELETE RESTRICT,
73     TopicName varchar REFERENCES Topic(TopicName) ON DELETE RESTRICT ,
74     PRIMARY KEY (PaperId,TopicName)
75 );
76
77 CREATE TABLE Submit_To(
78     SubmitID INTEGER PRIMARY KEY ,
79     PaperID INTEGER REFERENCES Paper(PaperId) ON DELETE RESTRICT ,
80     ConferenceID INTEGER REFERENCES Conference(ConferenceID),
81     Accept_and_Publish BOOLEAN not null ,
82     Date date not null
83 );
84
85 CREATE TABLE User_Member_OF(
86     LoginName varchar REFERENCES Conference_User(LoginName),
87     ConferenceID INTEGER REFERENCES CONFERENCE(ConferenceID),
88     is_Chair BOOLEAN not null,
89     PRIMARY KEY (LoginName,ConferenceID)
90 );
91
92 CREATE TABLE User_Biding(
93     LoginName varchar REFERENCES Conference_User(LoginName) ON DELETE
94     RESTRICT ,
95     TopicName varchar REFERENCES Topic(TopicName) ON DELETE RESTRICT ,
96     Prefrence varchar not null,
97     PRIMARY KEY (LoginName,TopicName)
98 );
99
100 CREATE TABLE Review_Form(
101     LoginName varchar REFERENCES Conference_User(LoginName) ON DELETE
102     RESTRICT ,
103     ConferenceID INTEGER REFERENCES CONFERENCE(ConferenceID),
104     form varchar not null ,
105     PRIMARY KEY (LoginName,ConferenceID)
106 );
107
108 CREATE TABLE Notify(
109     LoginName varchar REFERENCES Conference_User(LoginName) ON DELETE
110     RESTRICT ,
111     ConferenceID INTEGER REFERENCES CONFERENCE(ConferenceID),
112     notify_message varchar not null ,
113     PRIMARY KEY (LoginName,ConferenceID)
114 );

```

