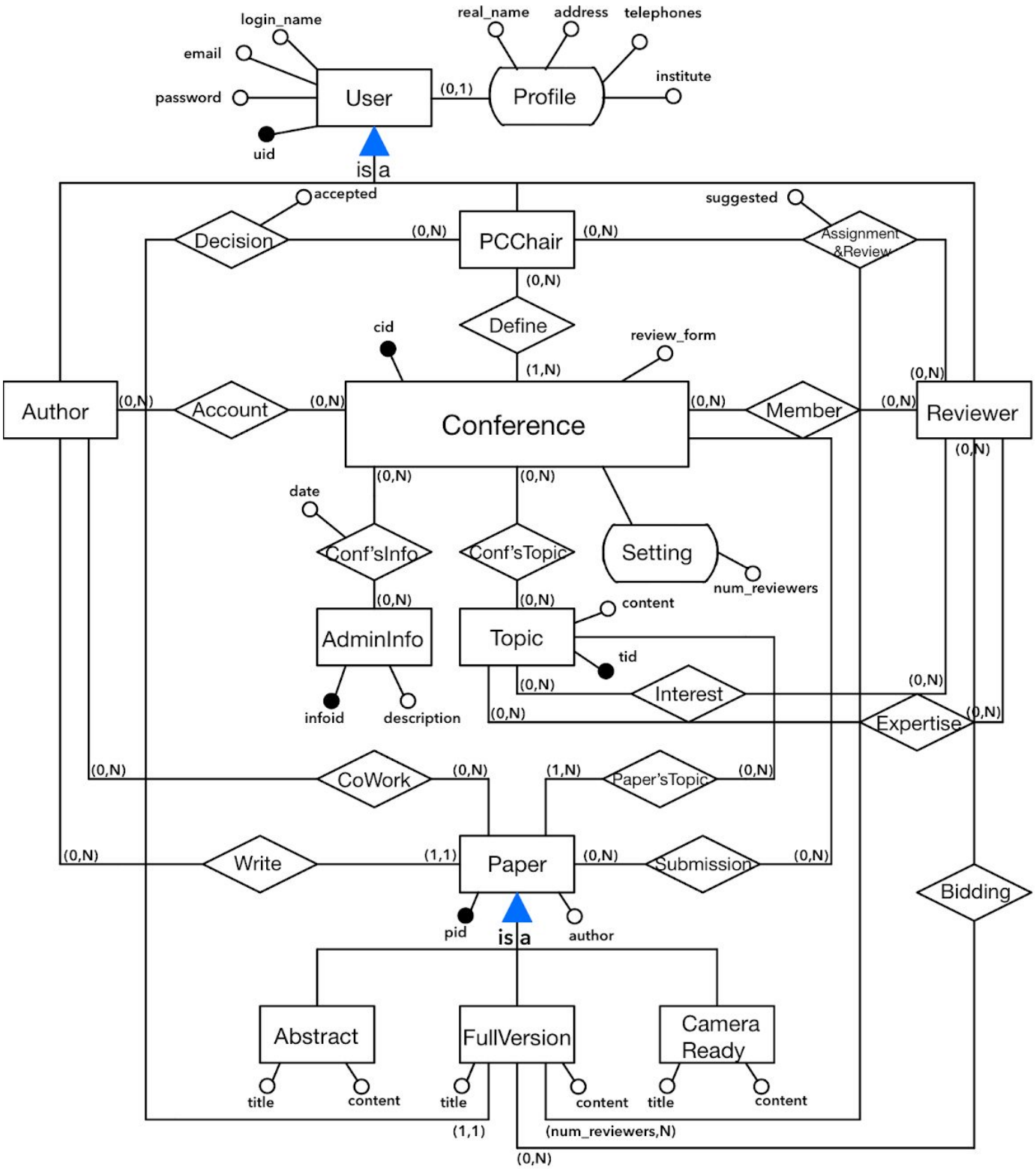
Assumptions:

- Profile is optional, and users have at most one profile.
- Chairs can define reviewers, additional chairs, settings…. For all those defining, we only store the results. For example, a chair defined a reviewer in a conference, we only record the result “the user in that conference is a reviewer” and ignore the chair who defined it.
- Authors, pcChairs and reviewers are all users(i.e they all use uid in any conference).
- A conference have at least one PC chair and any number of authors and reviewers
- A User can enroll in multiple conferences and act different roles
- A topic has a unique content, can appear in multiple conference.
- An administrative conference information has a unique description, can appear in multiple conference at different dates.
- Conferences must contain a review form, and the review form referred below is an integer representation of a real review form which is a file stored in a computer's file system. So we just store the integer representation, the app will recognize the representation and then open the corresponding form for user.
- Every conference has an administrative setting, and so far the number of reviewers required for a paper is the only setting which has default value 3.
- A paper with a unique pid can only have one author and have at least one topic, can be submitted in multiple conferences.
- There are three versions of paper stored in db so that users have the flexibility to switch between versions. And the title, content, submitted conferences, co-authors, topics might different amount versions although they share the same paper id.
- Abstract Papers may don’t have any content, but Full version and Camera Ready Papers must have contents.
- Only full-version papers that are accepted can have camera-ready version.
- Reviews and decisions are made based on full-version papers.
- A submitted full version paper in a conference can only have one decision (accept or reject) made by one chair.
- A submitted full version paper in a conference can be assigned to multiple reviewers by different chairs.
- We don’t need an entity or relationship for COI, because we can join FullVersionPaperCoAuthor, FullVersionPaperSubmission, Reviewer to get all COIs.
- A chair can assign a reviewer papers, a reviewer can give suggestions toward a paper, the paper is submitted in conference that the chair in charge. So we trade this as a ternary relationships.
- Reviewers can have multiple interested or Expertised topics, and can bid for multiple submitted full version papers.

ER:


login_name
email
password
uid
User
(0,1)
real_name
address
telephones
institute
Profile
is a
accepted
Decision
(0,N)
PCChair
(0,N)
suggested
Assignment &Review
(0,N)
Define
cid
review_form
(1,N)
Author
(0,N)
Account
(0,N)
Conference
(0,N)
Member
(0,N)
Reviewer
(0,N)
(0,N)
(0,N)
date
Conf'sInfo
Conf'sTopic
Setting
num_reviewers
(0,N)
(0,N)
content
AdminInfo
Topic
tid
infoid
description
(0,N)
Interest
(0,N)
(0,N)
Expertise
(0,N)
(0,N)
CoWork
(0,N)
(1,N)
Paper'sTopic
(0,N)
(0,N)
Write
(1,1)
Paper
(0,N)
Submission
(0,N)
Bidding
pid
is a
author
Abstract
FullVersion
Camera Ready
title
content
title
content
title
content
(1,1)
(num_reviewers,N)
(0,N)

User(uid, login_name, email, password)

```
CREATE TABLE User(
        uid             INTEGER             PRIMARY KEY,
        login_name      VARCHAR(25)         NOT NULL,
        email           VARCHAR(25)         NOT NULL,
        password        VARCHAR(25)         NOT NULL
        );
```

Profile(user, real_name, institute, telephone, address)

Profile[user] ⊆ User[uid]

```
CREATE TABLE Profile(
        user            INTEGER             REFERENCES User(uid) ON DELETE RESTRICT,
        real_name       VARCHAR(25),
        institute       VARCHAR(25),
        telephone       VARCHAR(25),
        address         VARCHAR(25)
        );
```

Conference(cid, review_form, num_reviewers)

```
CREATE TABLE Conference(
        cid                 INTEGER     PRIMARY KEY,
        review_form         INTEGER     NOT NULL,
        num_reviewers       INTEGER     NOT NULL DEFAULT 3
        );
```

PCChair(user, conference)

PCChair[user] ⊆ User[uid]

PCChair[conference] ⊆ Conference[cid]

```
CREATE TABLE PCChair(
        user            INTEGER     REFERENCES User(uid) ON DELETE RESTRICT,
        conference      INTEGER     REFERENCES Conference(cid) ON DELETE RESTRICT,
        PRIMARY KEY(user, conference));
```

Reviewer(user, conference)

Reviewer[user] ⊆ User[uid]

Reviewer[conference] ⊆ Conference[cid]

```
CREATE TABLE Reviewer(
        user            INTEGER     REFERENCES User(uid) ON DELETE RESTRICT,
        conference      INTEGER     REFERENCES Conference(cid) ON DELETE RESTRICT,
        PRIMARY KEY(user, conference));
```

Author(user, conference)

Author[user] ⊆ User[uid]

Author[conference] ⊆ Conference[cid]

```
CREATE TABLE Author(
```

```
        user            INTEGER     REFERENCES User(uid) ON DELETE RESTRICT,
        conference      INTEGER     REFERENCES Conference(cid) ON DELETE RESTRICT,
        PRIMARY KEY(user, conference));
```

Topic(tid, content)

```
CREATE TABLE Topic(
        tid             INTEGER     PRIMARY KEY,
        content         TEXT        NOT NULL
        );
```

ConferenceTopic(topic,conference)

ConferenceTopic[topic] ⊆ Topic[tid]

ConferenceTopic[conference] ⊆ Conference[cid]

```
CREATE TABLE ConferenceTopic(
        topic           INTEGER     REFERENCES Topic(tid) ON DELETE RESTRICT,
        conference      INTEGER     REFERENCES Conference(cid) ON DELETE RESTRICT,
        PRIMARY KEY(user, conference));
```

AdminInfo(infoid, description)

```
CREATE TABLE AdminInfo(
        infoid          INTEGER     PRIMARY KEY,
        description     TEXT        NOT NULL
        );
```

ConferenceInfo(admin_info, conference, date)

ConferenceInfo[admin_info] ⊆ AdminInfo[infoid]

ConferenceInfo[conference] ⊆ Conference[cid]

```
CREATE TABLE ConferenceInfo(
        admin_info      INTEGER     REFERENCES AdminInfo(infoid) ON DELETE RESTRICT,
        conference      INTEGER     REFERENCES Conference(cid) ON DELETE RESTRICT,
        date            DATE        NOT NULL,
        PRIMARY KEY(user, conference)
        );
```

Paper(pid, author)

Paper[author] ⊆ Author[user]

```
CREATE TABLE Paper(
        pid             INTEGER     PRIMARY KEY,
        author          INTEGER     REFERENCES Author(user) ON DELETE RESTRICT
        );
```

AbstractPaper(paper, title, content)

AbstractPaper[paper] ⊆ Paper[pid]

```
CREATE TABLE AbstractPaper(
        paper           INTEGER     REFERENCES Paper(pid) ON DELETE RESTRICT,
```

```
        title           VARCHAR(50)         NOT NULL,
        content         TEXT,
        PRIMARY KEY(paper));
```

AbstractPaperSubmission(paper, conference)

AbstractPaperSubmission[paper] ⊆ Paper[pid]

AbstractPaperSubmission[conference] ⊆ Conference[cid]

```
CREATE TABLE AbstractPaperSubmission(
        paper           INTEGER     REFERENCES Paper(pid) ON DELETE RESTRICT,
        conference      INTEGER     REFERENCES Conference(cid) ON DELETE RESTRICT,
        PRIMARY KEY(paper, conference));
```

AbstractPaperCoAuthor(paper, co_author)

AbstractPaperCoAuthor[paper] ⊆ Paper[pid]

AbstractPaperCoAuthor[co_author] ⊆ Author[user]

```
CREATE TABLE AbstractPaperCoAuthor(
        paper           INTEGER     REFERENCES Paper(pid) ON DELETE RESTRICT,
        co_author       INTEGER     REFERENCES Author(user) ON DELETE RESTRICT,
        PRIMARY KEY(paper, co_author));
```

AbstractPaperTopic(paper, topic)

AbstractPaperTopic[paper] ⊆ Paper[pid]

AbstractPaperTopic[topic] ⊆ Topic[topic]

```
CREATE TABLE AbstractPaperTopic(
        paper           INTEGER     REFERENCES Paper(pid) ON DELETE RESTRICT,
        topic           INTEGER     REFERENCES Topic(tid) ON DELETE RESTRICT,
        PRIMARY KEY(paper, topic));
```

FullVersionPaper(paper, title, content)

FullVersionPaper[paper] ⊆ Paper[pid]

```
CREATE TABLE FullVersionPaper(
        paper           INTEGER     REFERENCES Paper(pid) ON DELETE RESTRICT,
        title           VARCHAR(50)         NOT NULL,
        content         TEXT                NOT NULL,
        PRIMARY KEY(paper));
```

FullVersionPaperSubmission(paper, conference, chair, is_accepted)

FullVersionPaperSubmission[paper] ⊆ Paper[pid]

FullVersionPaperSubmission[conference] ⊆ Conference[cid]

FullVersionPaperSubmission[chair] ⊆ PCChair[user]

```
CREATE TABLE FullVersionPaperSubmission(
        paper           INTEGER     REFERENCES Paper(pid) ON DELETE RESTRICT,
        conference      INTEGER     REFERENCES Conference(cid) ON DELETE RESTRICT,
        chair           INTEGER     REFERENCES PCChair(user) ON DELETE RESTRICT,
        is_accepted     BOOLEAN,
        PRIMARY KEY(paper, conference));
```

FullVersionPaperCoAuthor(paper, co_author)

FullVersionPaperCoAuthor[paper] ⊆ Paper[pid]

FullVersionPaperCoAuthor[co_author] ⊆ Author[user]

```
CREATE TABLE FullVersionPaperCoAuthor(
    paper       INTEGER     REFERENCES Paper(pid) ON DELETE RESTRICT,
    co_author   INTEGER     REFERENCES Author(user) ON DELETE RESTRICT,
    PRIMARY KEY(paper, co_author));
```

FullVersionPaperTopic(paper, topic)

FullVersionPaperTopic[paper] ⊆ Paper[pid]

FullVersionPaperTopic[topic] ⊆ Topic[topic]

```
CREATE TABLE FullVersionPaperTopic(
    paper       INTEGER     REFERENCES Paper(pid) ON DELETE RESTRICT,
    topic       INTEGER     REFERENCES Topic(tid) ON DELETE RESTRICT,
    PRIMARY KEY(paper, topic));
```

Camera-readyPaper(paper, title, content)

Camera-readyPaper[paper] ⊆ FullVersionPaper[paper]

```
CREATE TABLE Camera-readyPaper(
    paper       INTEGER     REFERENCES Paper(pid),
    title       VARCHAR(50)         NOT NULL,
    content     TEXT                NOT NULL,
    CHECK (paper in (select paper from FullVersionPaperSubmission where is_accepted)),
    PRIMARY KEY(paper));
```

Camera-readyPaperSubmission(paper, conference)

Camera-readyPaperSubmission[paper] ⊆ FullVersionPaper[paper]

Camera-readyPaperSubmission[conference] ⊆ FullVersionPaperSubmission[conference]

```
CREATE TABLE Camera-readyPaperSubmission(
    paper       INTEGER     REFERENCES Paper(pid) ON DELETE RESTRICT,
    conference  INTEGER     REFERENCES Conference(cid) ON DELETE RESTRICT,
    CHECK (paper in (select paper from FullVersionPaperSubmission where is_accepted)),
    PRIMARY KEY(paper, conference));
```

Camera-readyPaperCoAuthor(paper, co_author)

Camera-readyPaperCoAuthor[paper] ⊆ FullVersionPaper[paper]

Camera-readyPaperCoAuthor[co_author] ⊆ Author[user]

```
CREATE TABLE Camera-readyPaperCoAuthor(
    paper       INTEGER     REFERENCES Paper(pid) ON DELETE RESTRICT,
    co_author   INTEGER     REFERENCES Author(user) ON DELETE RESTRICT,
    CHECK (paper in (select paper from FullVersionPaperSubmission where is_accepted)),
    PRIMARY KEY(paper, co_author));
```

Camera-readyPaperTopic(paper, topic)

Camera-readyPaperTopic[paper] ⊆ FullVersionPaper[paper]

Camera-readyPaperTopic[topic] ⊆ Topic[topic]

```
CREATE TABLE Camera-readyPaperTopic(
        paper           INTEGER     REFERENCES Paper(pid) ON DELETE RESTRICT,
        topic           INTEGER     REFERENCES Topic(tid) ON DELETE RESTRICT,
        CHECK (paper in (select paper from FullVersionPaperSubmission where is_accepted)),
        PRIMARY KEY(paper, topic));
```

TopicInterest(topic, reviewer)

Preference[paper] ⊆ FullVersionPaper[paper]

Preference[reviewer] ⊆ Reviewer[user]

```
CREATE TABLE TopicInterest(
        topic           INTEGER     REFERENCES Topic(tid) ON DELETE RESTRICT,
        reviewer        INTEGER     REFERENCES Reviewer(user) ON DELETE RESTRICT,
        PRIMARY KEY(topic, reviewer));
```

TopicExpertise(topic, reviewer)

Preference[paper] ⊆ FullVersionPaper[paper]

Preference[reviewer] ⊆ Reviewer[user]

```
CREATE TABLE TopicExpertise(
        topic           INTEGER     REFERENCES Topic(tid) ON DELETE RESTRICT,
        reviewer        INTEGER     REFERENCES Reviewer(user)
        PRIMARY KEY(topic, reviewer));
```

Bidding(paper, conference, reviewer)

Bidding[paper] ⊆ FullVersionPaper[paper]

Bidding[conference] ⊆ FullVersionPaperSubmission[conference]

Bidding[reviewer] ⊆ Reviewer[user]

```
CREATE TABLE Bidding(
        paper           INTEGER,
        conference      INTEGER,
        reviewer        INTEGER,
        FOREGIN KEY (paper, conference) REFERENCES FullVersionPaperSubmission
        (paper, conference) ON DELETE RESTRICT,
        FOREGIN KEY (reviewer, conference) REFERENCES  Reviewer (user,
        conference) ON DELETE RESTRICT,
        CHECK (paper in (select paper from FullVersionPaperSubmission where is_accepted)),
        PRIMARY KEY(paper, conference, reviewer));
```

AssignAndReview(paper,conference, reviewer, suggested)

AssignAndReview[paper] ⊆ FullVersionPaper[paper]

AssignAndReview[conference] ⊆ FullVersionPaperSubmission[conference]

AssignAndReview[reviewer] ⊆ Reviewer[user]

```
CREATE TABLE AssignAndReview(
        paper           INTEGER,
        conference      INTEGER,
        reviewer        INTEGER,
```

```
suggested       BOOLEAN,
FOREGIN KEY (paper, conference) REFERENCES FullVersionPaperSubmission
(paper, conference) ON DELETE RESTRICT,
FOREGIN KEY (reviewer, conference) REFERENCES  Reviewer (user,
conference) ON DELETE RESTRICT,
CHECK (paper in (select paper from FullVersionPaperSubmission where is_accepted)),
PRIMARY KEY(paper, conference, reviewer));
```